Université de Toulouse Paul-Sabatier
IMT (Institut de Mathématiques de Toulouse)

Habilitation à diriger des recherches

présentée en première version en vu d'obtenir le grade de Docteur habilité, spécialité « Mathématiques appliquées »

par

Raphaël Loubère

Contribution au domaine des méthodes numériques Lagrangiennes et Arbitrary-Lagrangian-Eulerian

Habilitation soutenue le 28 Juin 2013 devant le jury composé de :

M.	Andy Barlow	Atomic Weapon Establishment, UK	(Rapporteur)
M.	Daniel Bouche	CEA-DIF, France	(Examinateur)
M.	Stéphane Clain	Universitade Do Minho, Portugal	(Examinateur)
M.	Frédéric Coquel	Université Paris VI, France	(Rapporteur)
M.	Pierre Degond	Université Paul-Sabatier Toulouse 3, France	(Directeur)
M.	Patrick Hild	Université Paul-Sabatier Toulouse 3, France	(Examinateur)
M.	Richard Liska	Czech Technical University in Prague, République Tchèque	(Examinateur)
M.	Pierre-Henri Maire	CEA-CESTA, France	(Examinateur)
M.	Philip Roe	University Michigan, U.S.A	(Rapporteur)
M.	Richard Saurel	IUF Université Aix Marseille 1, France	(Examinateur)

A Margaux, Calixte et Clotilde.

Socrates once said, "I swear it upon Zeus an outstanding runner cannot be the equal of an average wrestler."

From the last day of my student time up to now, at the mathematical laboratory of Bordeaux university, the CEA, the Los Alamos National Laboratory in the New-Mexican desert, the Mathematical institute in Toulouse, I came across so many outstanding persons. Some of them ultimately became friends of mine and I am particularly proud of it. Let me use this acknowledgment section to praise them the best way I can.

First of all I would like to warmly thank Philip Roe, Frédéric. Coquel and Andrew J. Barlow for carefully reviewing this habilitation. This is a thankless job and the fact that they spent some of their free time on this thesis is genuinely appreciated — especially because they are important scientists with busy life.
Also I would like to thank Daniel Bouche, Stéphane Clain, Patrick Hild, Richard Liska, Pierre-Henri Maire, and Richard Saurel for being member of this jury : Their presence is greatly appreciated. I am also indebted to Pierre Degond who accepted to supervise this work and be part of the jury. During my stay in the U.S.A I met many important persons who helped me and inspired me. Amongst many of them stand Mikhail Shashkov who mentored me for three years back then. He introduced me to this new world of ALE and I am very grateful for his support and the possibility he offered me to parasite his aura.
I am also very grateful to Burton Wendroff with whom I had the chance to share an office at Los Alamos. Let me put that straight : Burt is a legend. Sharing time, stories, discussions and working with him were some of great moments of my scientific life mostly because behind the legend there is a adorable man, even greater than the legend.
The last very important scientific person I had to meet in the U.S.A was Ed J. Caramana. During a difficult time for Los Alamos National Laboratory I enjoyed his vehement diatribe against "the system" in general but also his passionate faith for the compatible staggered Lagrangian scheme. He was kind enough to spend hours trying to teach me each and every subtle details of this scheme, which sometimes, I can confess now, were not entirely clear to me. Nevertheless he was truly patient and associated me with his investigations so that, once again, I was shining through him.
Of course I can not forget the sympathetic atmosphere of the ex-T-7 team hold by very friendly colleagues plus some others whom I met during intense working session or simply to share an espresso with or without sugar.
My life could not be as pleasant as it is now without the unfailing support from the Czech Republic. Well, maybe not the entire country, but at least three good Czech fellows, Richard Liska, Pavel Váchal and Milan Kuchařík. Scientifically and personally it is always a pleasure to meet them.
Back to France I wish to acknowledge the scientific support of Remi Abgrall since the time of my PhD up to now.
Presumably the person I am the most indebted to is Pierre-Henri Maire. Mainly because he has accepted to pair up with me for many different research topics. In some sense we are a kind of improbable scientific couple ; he is methodical and rigorous while I do everything with too much artistic freedom, he has bright ideas while I have stupid bugs, he is all dressed up to the nines while I still dress up like a teenager, etc. But despite that, I am proud of the work we have done.
While Pierre-Henri was a team leader at CELIA (University of Bordeaux) he made me meet Jérôme Breil and Stéphane Galera with whom we actively collaborated. I had the chance to meet Vladimir Tikhonchuk (CELIA) during my PhD, this was during previous century. He was literally building a team mixing physics and numerics and did a great job. Very supportive from the beginning he is surely at the origin of my work keeping contact with real applications. Jean Ovadia did supervise my PhD from the engineering point of view and, I always look back at this time with good and refreshing souvenirs, I would like to use this opportunity to acknowledge his still-on-going-spiritual

contribution. It is often scary to realize by how much one mimics some of our fathers !
In Toulouse, since 2006 I had the chance to collaborate and interact on personal basis with many colleagues at IMT ; some of them became co-workers, some others partners for coffee, which is equally important for me. More specifically I would like to praise Fabrice Deluzet, Marie-Hélène Vignal, Alexei Lozinski and Giacomo Dimarco for their contributions as collaborators but also as friends. I am also indebted to Stéphane Clain who used to be the 'colleague next door' and is now my best address in Portugal. We used to collaborate in Toulouse while he was at IMT, and we still do since he arrived in Braga, Portugal. Not so long ago we were both proud when Steven Diot (our shared PhD student) has defended not because of our contribution, most of the job has been done by Steven, but for the probable good (enough) supervision we were able to provide. I wish we will pursue on this path. By the way I wish to thank Steven for the good MOOD he breathed into our common scientific lives.
Then there is a whole crowd of collaborators and friends from Paris which I would like to thank for their warm welcome each time I have to spend time there. Especially Philippe Hoch whom I often bother evenings and nights in Paris, also Jean-Philippe Braeunig for good time out and also letting me participating to the development of his code.
I would like to take the opportunity to warmly thank Renaud Motte from CEA and his whole team (more specifically the ones with whom I interact : Alexandra, Christophe, Corinne, Jean-Philippe, Laurent, Mathieu, Pascal and Raphaël), I wish we will pursue this collaboration for many more years. Also I would like to thank the CEA/DAM-DIF in general for a true support during the last six years which made my scientific life fairly easy : this is truly appreciated.
Should it be in Paris or Cargèse (Corsica) it is always a pleasure to work, interact, or simply exchange inappropriate jokes with Jean-Michel Ghidaglia. Thanks to him I have been introduced to the CMLA crowd where I met some of his co-workers. Also I was welcomed into some of his projects. Especially the connection with Florian de Vuyst is particularly fruitful and I genuinely hope that this is only the beginning of some good science with holograms and other 'geeky gadgets'.
Of course none of the above great moments would be possible without the stable basis brought by my in-laws and my family. I would like to deeply thank them for begin always supportive and helpful. Without any doubt they are more important than the number of written sentences of this paragraph dedicated to them.
Finally I am mostly grateful to Clotilde for her sure and constant support for these years. Not only she has accepted to marry (and have home 24/7) a pale reproduction of a character from "The big Bang Theory" series but she has also agreed to have two wonderful kids with me. Sometimes I wonder if she is not as much disturbed as I am.

Enfin à Margaux ma Princesse, et Calixte mon Champion merci de remplir mes jours de bonheur et d'insouciance.

Toulouse, le 17 juillet 2013.

Contents

Prologue and general introduction

This prologue is intended to provide some guide lines in order for the reader to follow my peregrinations, my collaborations and recast the publication list within this historical context.

Institut de physique fondamentale de Bordeaux, France
Centre d'Etude Lasers Intenses et Applications (CELIA)
November 2002 - January 2003

After the defense of a PhD from the University in Bordeaux in October 2002 [1] with R. Abgrall (INRIA Bordeaux) and J. Ovadia (retired fellow from CEA-CESTA) as supervisors, I have spent three months at the Fundamental Physics department under the supervision of V. Tikhonchuk (University of Bordeaux, CELIA laboratory). During this time I mostly interacted with S. Weber. The development of the cell-centered Lagrangian numerical scheme from [1, 2] has been pursued in a laser-plasma interaction context. This scheme has been coupled during this time to physics models in order to build a parallel transport simulation code for Inertial Confinement Fusion (ICF) [3, 4].

Los Alamos National Laboratory (LANL), New Mexico, U.S.A
Theoretical division - Mathematical, Modelisation and Analysis Group
February 2003 - December 2005

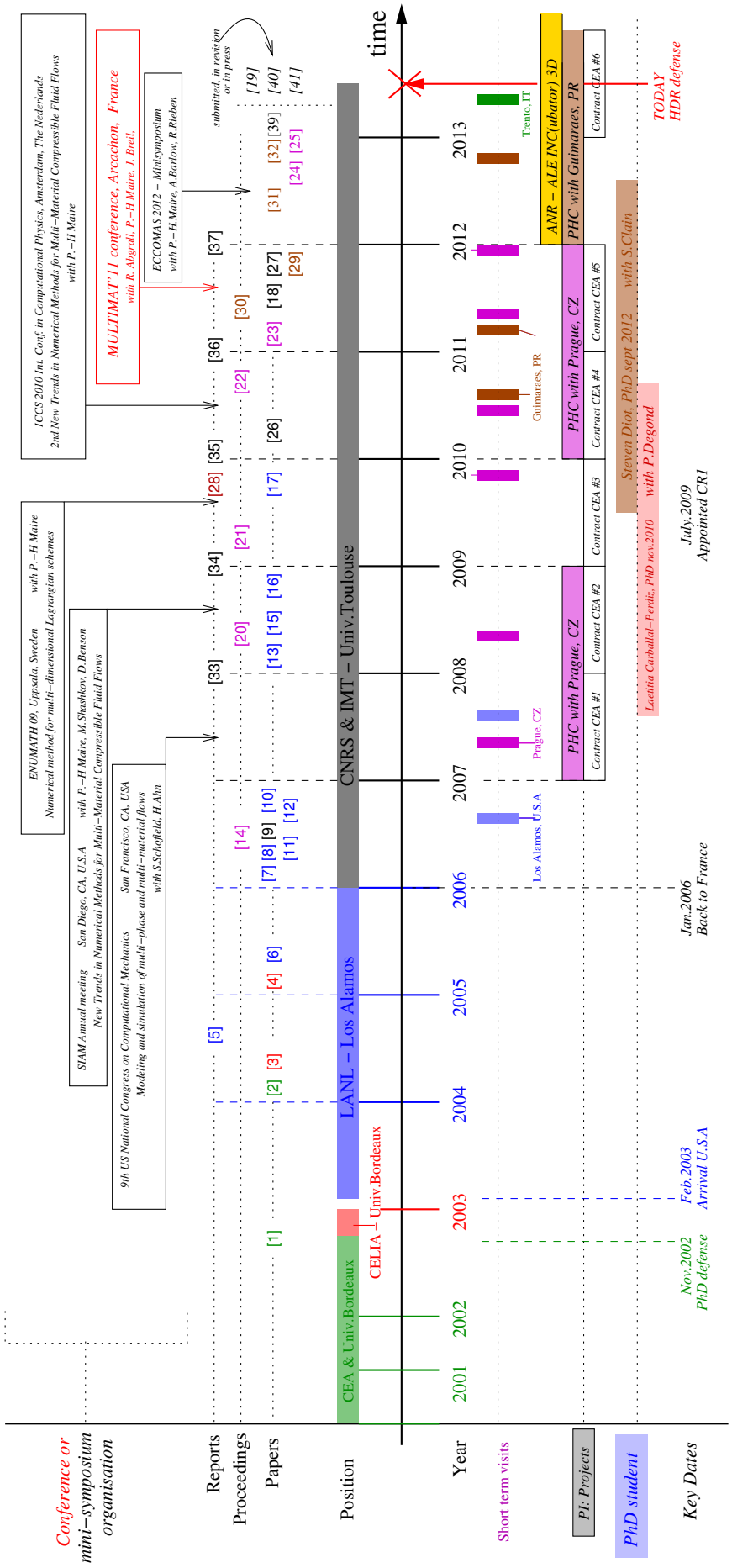
In February 2002 I have started a postdoctoral position at the Los Alamos National Laboratory (LANL) in New Mexico, U.S.A, under the supervision of M. Shashkov (a.k.a Misha). Most of my coauthors from this period are/were staff members at LANL or long/short term visitors (summer students, PhD students, postdocs) or invited professors.
Prior to my arrival at LANL a former postdoc of Misha, J. Campbell had developed a compatible staggered Lagrangian scheme in 2D on unstructured meshes. I have used this code to build an Arbitrary-Lagrangian-Eulerian simulation code for hydrodynamics equations called ALE INC(ubator) [5, 6]. More precisely I have implemented the untangling and rezone capabilities and developed with Misha a conservative remap module [7] and associated repair methods [8] with M. Staley and B. Wendroff (a.k.a. Burt). With B. Despres, summer visitor at LANL we have also written a 1D analysis of repair methods in [9].
In parallel E. J. Caramana associated me to his research on the compatible staggered Lagrangian scheme implemented in ALE INC(ubator). We have investigated the treatment of exceptional points (known also as T junction, dendritic zones) [10] and a vorticity damping artificial viscosity [11]. Later a joint effort with A. L. Bauer, D. E. Burton, M. J. Shashkov and P. P. Whalen gave birth to an

article dealing mostly with the analysis of this Lagrangian scheme in term of consistency, stability and accuracy in [12].
While developing multi-material capability into ALE INC(ubator) I have faced the situation of inadequate interface reconstruction method when three or more materials are present within the same mixed cell. Consequently with some of my colleagues of the ex-T-7 group at LANL, namely S.P. Schofield, R.V. Garimella and M.M. Francois, we have developed a material order independent interface reconstruction method using power diagrams in [13] which has however been published only in 2008.
Using the Czech connection of M. Shashkov I was lucky enough to meet M. Kuchařík, R. Liska and P. Váchal during their summer visits and we have started a never-ending (up to now) collaboration on ALE and Lagrangian numerical methods [14].
In December 2005, at the end of this postdoctoral position at LANL, I was involved in three main subjects of research : unstructured ALE simulation code development [5,6] (meshing, rezoning, remapping, repair), staggered Lagrangian scheme analysis and advanced interface reconstruction methods.

CNRS and Institut de Mathématiques de Toulouse (IMT), France
Mathématiques pour l'Industrie et la Physique (MIP) group
January 2006 - now

In January 2006 I was appointed as researcher at the Centre National de Recherche Scientifique (CNRS) at the Mathematics Institut in Toulouse (IMT).
For the first years I have mainly pursued research with colleagues from LANL. I also reconnected with former colleagues and advisors from Bordeaux and CEA researchers or postdocs appointed to CELIA laboratory, namely P.-H. Maire, J. Breil and S. Galera. The connection with the Czech team has been also strengthened with two EGIDE grants called Partenariats-Hubert-Curien (P.H.C) "Barrande" for 2007-2008 and 2010-2011. These grants consist of enhancing already existing collaborations by funding two short term visits of French researchers in Czech Republic and Czech researchers in France.
Concerning the analysis of Lagrangian scheme, B. Wendroff, A.L. Bauer and I have written the article [15] dealing with the proof of a conjectured stability result that was only numerically observed in [12]. Further with M. Shashkov and B. Wendroff we have analysed the problem of volume consistency of the staggered grid Lagrangian hydrodynamics scheme in [16]. The interface reconstruction method *via* power diagrams has been extended by the same team from Los Alamos (as in [13]) to a second-order accurate material-order-independent method in [17]. Recently in a collaboration with J.M. Ghidaglia (CMLA, ENS-Cachan) and J.P. Braeunig (CEA-DIF), we have adapted some techniques used for Lagrangian schemes and classical interface reconstruction to improve their Eulerian scheme in [18]. Moreover dealing with filament and structures smaller than the cell size is difficult with classical interface reconstruction techniques. As a consequence, following an idea from J. Ovadia (retired fellow from CEA-CESTA), C. Fochesato, R. Motte from the CEA-DIF and I have built an interface reconstruction method devoted to filament in [19].
The exchanges between IMT, CELIA and CTU gave also rise to several publications. We have investigated the comparison between staggered and cell-centered Lagrangian and ALE hydrodynamical methods in [20]. Following an idea of P.-H. Maire of recasting some technics used for cell-centered Lagrangian schemes into staggered Lagrangian schemes, P.-H. Maire, P. Váchal and I have developed a set of articles dealing with this general formalism to derive artificial viscosity and its second-order accurate version in 2D in [21, 22, 23]. Recently we also have extended this approach

to 3D with the same co-authors in [24]. Still surfing on the Czech connection I paired up with L.Bednarik, M. Kucharik and R. Liska to study to the concept of slide-line for the 2D compatible staggered Lagrangian scheme in [25].
Back in 2008 the ALE formalism in ALE INC(ubator) and in the CELIA ALE code CHIC did not allow any change of mesh connectivity while rezoning. Consequently with M. Shashkov and the CELIA team (P.-H. Maire, J. Breil and S. Galera) we have extended the ALE formalism to allow topology modifications of the mesh during the computation [26, 27]. This approach is called "ReALE" standing for Reconnection-based ALE.
In Toulouse at IMT with P. Degond we shared a PhD student (L. Carballal-Perdiz) from September 2007 up to November 2010 the subject of the PhD was the development of a multi-scale finite element method dedicated to the prediction of air contaminant transport on multiple scales [28]. The team involved in this research was also constituted of F. Deluzet, A. Lozinski and J.-M. Rovarch through a collaboration with DGA ("Direction Générale de L'Armement").
More recently a collaboration with S. Clain (a former colleague at IMT now appointed associate Professor at the Universidade do Minho, Guimaraẽs in Portugal) and our shared PhD student S. Diot brought by the opportunity to explore the world of very-high order Eulerian finite volume schemes and develop the MOOD method (Multi-dimensional Optimal Order Detection) for unstructured meshes in 2D in [29, 30, 31, 32] and in 3D in article [33]. This method is based on an unlimited high-degree polynomial reconstruction leading to a high-order accurate scheme complemented with an *a posteriori* polynomial order reduction on problematic detected cells. This method has shown very promissing behaviors both on advection and Euler equations on unstructured, non-regular 2D and 3D meshes. The PhD has been defended in August 2012. Meanwhile we won a P.H.C grant (program "Pessoa") for 2012-2013 to exchange researchers and students between the Portuguese institution and IMT which has already led to a common proceedings [31] and fruitful and promissing discussions.
Since 2006 I also have a very fruitful collaboration with researchers from CEA-DIF that led to many studies the topic of which covers staggered Lagrangian schemes and ALE methodology in [34, 35, 36, 37, 38].
A brand new collaboration with G. Dimarco (IMT) at the end of year 2011 led to the development of a fast discrete velocity method for kinetic equations in [39]. This method has been implemented on a mono-processor machine and we have shown that this method is efficient even in full dimensions : 3D in space and 3D in velocity, leading to the effective discretization of six dimensions. A second-order accurate extension of this method is under review in [40].



The time table on this page displays most of my research activities *versus* the time in abscissa : The positions I was appointed to, the short term visits in foreign institutions, the PhD students I mentored and the projects I run as Principal Investigator (PI).
Articles and reports are located according to the year of publication using a color to emphasize the collaboration it refers to, as instance navy blue color is associated to colleagues from Los Alamos, red color refers to articles written while being at CELIA, purple color corresponds to projects conducted with Czech colleagues, brown color to Portuguese colleagues and black to French collaborations. The very top lines provides the list of mini-symposia and conferences I organized or co-organized.

$\sim$

This thesis mainly focuses on works related to the domain of Lagrangian numerical schemes and Arbitrary-Lagrangian-Eulerian methods. Most of them are already published in international journals. Para-phrasing these publications would be of little interest for the readers. Instead I have tried to state the main contribution brought by some publications and to articulate them together in order to clarify the unity behind the scene.
Nevertheless I have borrowed some sentences and rephrased paragraphs from some of my papers which have been written with co-authors. Unavoidably some of the phrases the reader will find in this habilitation have been produced by some of my co-authors to whom I am very grateful. Finaly some descriptions in this thesis are freely inspired by seminal papers and books which are cited at the begining of each associated paragraph.

Undoubtedly my work is led by the constant desire to improve the code ALE INC(ubator) developed at LANL and still maintained at IMT. Also, due to my natural tendency to interact with people, I have had a lot of golden opportunities for collaboration... which I took without any hesitation.

$\sim$

At the time of the publication of this habilitation the number of articles published in international peer review journals is 25, the number of proceedings in international conference with review is 4, the number of unpublished work made for National laboratories (Los Alamos National Laboratory or CEA-DAM) is 12. All are cited at the begining of the bibliography.
Moreover the following articles are discussed and reproduced in this thesis :

1. **Lagrangian chapter**
   - [12] entitled *The internal consistency, accuracy and stability of the discrete compatible Formulation of Lagrangian Hydrodynamics* (pages 33-39)
   - [15] entitled *On stabiliy analysis of staggered schemes* (pages 41-45)
   - [16] entitled *Volume consistency in a staggered grid Lagrangian hydrodynamics scheme* (pages 47-49)
   - [11] entitled *"Curl-q" : A vorticity damping artificial viscosity for essentially irrotational Lagrangian hydrodynamics calculations*. (pages 52-54)
   - [10] entitled *The Force/Work Differencing of Exceptional Points in the Discrete, Compatible Formulation of Lagrangian Hydrodynamics* (pages 57-62)
   - [25] entitled *Enhancement of Lagrangian slide lines as a combined force and velocity boundary condition*. (pages 65-70)
   - [23] entitled *Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydro-dynamics scheme* (pages 77-87)
   - [24] *3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver based artificial viscosity* (pages 87-93)
2. **ALE chapter**
   - [7] entitled *A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods* (pages 106-115)

  - [8] entitled *The Repair Paradigm : New Algorithms and Applications to Compressible Flow* (pages 118-123) and paper [9] entitled *Convergence and Sensitivity Analysis of Repair Algorithms in 1D* (pages 124-129)
  - [26] entitled *ReALE : a reconnection-based arbitrary-Lagrangian-Eulerian method* (pages 146-159) and paper [27] entitled *ReALE : a Reconnection Arbitrary-Lagrangian-Eulerian method in cylindrical geometry* (pages 161-164)
  - [13, 17] entitled *Material order independent interface reconstruction using power diagrams* and *A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations* (pages 174-179 and pages 179-184 respectively)

3. **Miscellaneous chapter**
  - [29, 32, 33] entitled *A high-order finite volume method for hyperbolic systems : Multi-dimensional Optimal Order Detection (MOOD)* (pages 191-197) *Improved Detection Criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials* (pages 197-203) *The MOOD method in the three-dimensional case : Very-High-Order Finite Volume Method for Hyperbolic Systems* (pages 203-212)
  - [39] entitled *Towards an ultra efficient kinetic scheme. Part I : basics on the BGK equation* (pages 219-227)
  - [18] *A totally Eulerian Finite Volume solver for multi-material fluid flows : Enhanced Natural Interface Positioning (ENIP)* (pages 232-235) and [41] entitled *Dealing with more than two materials in FVCF-ENIP method* (pages 235-241)

Finally some presentations of the conferences and minisymposia organized by the "MULTIMAT community" can be downloaded under the conferences' links at `http://www.math.univ-toulouse.fr/HYDRO`. This web site maintained by myself in Toulouse is intended to create links for this specific community. This community revolves around developers of ALE type of numerical methods and meets every other year during an international week of conference appropriately called "MULTIMAT conference". Successful conferences held in Paris in 2002, in Oxford in 2005, in Prague in 2007, in Pavia in 2009 and Arcachon 2011 aim at bringing together researchers from universities and research labs to discuss the state of the art for multi-material hydrodynamics simulations. As far as I can tell the custom of these meetings started thanks to a minisymposium organized by Mikhail Shashkov during a SIAM Annual Meeting in San Diego in 2001. Nowadays the community is also trying to meet during 'less crowded' workshops organized by some of the main characters, the last one was held during ECCOMAS conference in Vienna in September 2012.

These meetings are of great importance to maintain some alive, dynamical and friendly competitive atmosphere between the members of the community.

Compatible staggered Lagrangian schemes

1

Contents

While new cell-centered Lagrangian schemes devoted to ALE simulations have been developed in the first 10 years of this centrury, see [42] for an exhaustive historical background presentation and most of all for the detailed description of the schemes developed in [43, 44, 45, 46, 47, 48, 49, 50, 51, 42], the literature on staggered Lagrangian numerical methods dates back to the origin of computers. Nonetheless it is still very much alive these days in a finite difference context [52, 53, 54, 12, 11, 10, 55, 56, 57, 46] or in a finite element context [58, 59, 60, 61, 62].

This chapter more specifically deals with the so-called compatible staggered Lagrangian numerical scheme dedicated to solve hydrodynamics equations on general polygonal/polyhedral grid. This method has been popularized by E. J. Caramana *et al.* in a series of articles in the late 90's [55, 63, 64, 12].

We first present the historical background of this venerable numerical method and then describe the version one considers for solving the compressible hydrodynamics equations. Apart from different notation and alternative ways of presenting the scheme, the results presented in these first two sections have not been obtained by myself. Contrarily in sections 1.3 and 1.4 are presented some of

the contributions obtained by my colleagues and myself concerning the developement, understanding and analysis of this numerical method. However an exhaustive presentation of the scheme is mandatory to genuinely enlight the difficulties and features of this numerical scheme.

1.1 History and presentation

The origins of this compatible staggered Lagrangian numerical scheme are probably to be found in classified document from the Los Alamos National Laboratory, NM, U.S.A, during World War II and the Manhattan project where the "calculation of certain time-dependent fluid flows played an important part in the wartime work of the laboratory" (preface to the first edition of [65]).

Indeed the Lagrangian formulation of the equations of hydrodynamics has a very old and venerable history. The very first numerical calculations that resemble modern computer simulations in the numerical issues considered utilized fluid equations in the Lagrangian frame of reference in 1D [66].

Newton's second law of motion, which is central to any Lagrangian frame of reference relates the force $\boldsymbol{F}_p$ acting on a point of mass M_p and its acceleration $\boldsymbol{A}_p$ computed as the second derivative in time of its position $\boldsymbol{X}_p$: $\boldsymbol{F}_p = m_p \boldsymbol{A}_p$. The discretization by respect to time quite naturaly involves three time levels denoted $n-1, n$ and $n+1$ and a three-level leap-frog scheme with the force centered at time level n. All early Lagrangian schemes in 1D [66] or 2D [65] utilized such a staggered discretization in time. Although this forms a simple and intuitive numerical integration scheme, it leaves the velocity of a mass point defined as the difference between its displacement vector at two different time levels, and therefore the velocity is trully defined only at the $n+1/2$ time levels. When one then considers the total energy of a fluid as a sum of kinetic energy and internal energy that can be exchanged between each other by the action of forces, this sum is difficult to conserve exactly in discrete form owing to the fact that the two components that comprise it are defined at different time levels. Quoting Caramana in [12] *When velocity dependent forces are explicitly added to this model, as with the artificial viscosity [67], this type of time integration becomes somewhat clumsy and looks even contrived [68], since the artificial viscosity terms must be lagged in time to preserve numerical stability.* The spatial discretization of the force in all early versions of Lagrangian hydrodynamics [65, 69] is some form of what is presently known as finite-volume differencing. That is, these various forms calculate the force as a stress (scalar pressure plus deviators) times a normal surface-area vector. The most modern of these older force calculations is the diamond differencing scheme due to Wilkins [69], which uses closed surface area contours to calculate the force acting on a point, and thus properly conserves linear momentum. Other authors arrange the force contributions together in various ways to form the total force acting on each fluid element such that strict conservation of linear momentum may, or may not, be obtained. Most Lagrangian hydrodynamics codes employ a spatially staggered placement of dependent variables with stress, density, and specific internal energy given in cells surrounded by points that have associated position and velocity vectors. This enables the calculation of forces by means of various kinds of finite-volume differencing, with masses and volumes ascribed to both cells and points in an interleaved manner. A difficulty with the older work is that there was no agreement amongst the various authors of these different algorithms as to how these schemes, aside from the noted common features, should be constructed. The choices made were largely arbitrary and not derived from solid mathematical concepts.

An early attempt to remedy this lack of a sound theoretical basis is the work of Goad [70], who used the method of virtual work to derive a form of finite-volume force differencing of the stress in 2D cylindrical geometry. This work was little noted, partly because this type of scheme does not yield the limit of 1D spherical geometry from 2D cylindrical geometry.

Up to our knowledge the work that first places this type of finite-volume algorithm on a firm theoretical basis is due to Favorskii [71], and independently, Margolin and Adams [72]. The first paper shows that the discrete equations in Lagrangian form can be generally derived from a variational principle. It also justifies the use of the surface area vectors of closed volumes as appropriate discretization objects, a practice which was previously employed, but not always correctly, because the surface areas about a point did not in all cases sum to zero. The second paper parallels this work. Its central thrust is to use the continuity equation in discrete form to derive finite-volume differencing given a discrete expression for the volume of a cell. This results in, and also justifies the use of surface areas to calculate the force. It emphasizes that the difference formulas that are derived are "operator" expressions that can be used to calculate discrete derivatives of any function, and not just of the velocity field. It is the discrete form of the continuity equation as emphasized in [72] that is central to the internal consistency of the scheme. A further extension of the work of Favorskii is nowadays known as the "method of support operators" [73]. Although this work is more general than just its application to the equations of Lagrangian hydrodynamics, it is this system of equations that is used in its original exposition. This method also utilizes the continuity equation in discrete form to derive the divergence operator and then uses the vector identities in summation form to derive discrete versions of all other operators. It emphasizes the relation in discrete form of the divergence and gradient operators as negative adjoints of each other as in the continuum case.
These publications all revolve around the central idea that the discrete equations must obey the global properties of the continuum ones in order to be considered as valid discretizations that will then mirror continuum conservation properties in their discrete analogs. As such they remove the arbitrary and heuristic formulations of the previous codes based on the older work [69]. Somewhat after the previously cited developments is the seminal work of Burton [74, 75], which discretizes the fluid equations in Lagrangian form on a staggered spatial grid utilizing subgrid quantities termed subcell masses and subcell forces, from which the cell and point masses, and the total force acting on a point, are constructed. A two-level time integration scheme is also utilized so that both kinetic energy and internal energy are defined at the same time level. The basic reasoning used by Burton to demonstrate conservation of total energy is the same as that employed in the method of support operators [73], and thus incorporates the important features of the previous works [73, 72, 74]. However, Burton's formulation is more general in that he does not consider forces, or differential operators, of any specific origin. Instead, he utilizes an arbitrary subcell force that allows the specification of forces of any forces from functional form. The associated work is completely defined, and, total energy is also exactly conserved. The only restriction on the discrete form of the subcell force is constraint of momentum conservation. He also notes [75] that this formulation of the Lagrangian hydrodynamics equations contains two distinct definitions of cell volume, and considers this difference to be a form of entropy error. It is this latter work of Burton that we refer as the "discrete, compatible formulation of Lagrangian hydrodynamics", and which was initially constructed on arbitrary polyhedral grids [75]. The word "discrete" has been inserted in [12] to emphasize that these equations are essentially created in discrete form, as opposed to being the discretization of a system of PDE's. As such, one may or may not be able to rigorously take the continuum limit to obtain the latter ; this depends on the kinds of forces that are employed, as instance artificial viscosity and anti-hourglass forces.
Finally, the discrete, compatible formulation of Lagrangian hydrodynamics was developed to be an algebraic identity : this identity consists of two arbitrary scalars, the cell and point masses, and one arbitrary vector, the subcell force, such that given the usual definition of total energy conservation is always fulfilled. As such, it describes a priori truth that cannot be confuted, since in primitive form

it makes no assertion about any physical system. The quality with which the discrete, compatible formulation of Lagrangian hydrodynamics may describe certain physical situations is mostly, if not entirely, dependent on the quality of the specification of the three abstract quantities that compose it.

In next section we state the governing equation, notation and derive the compatible staggered Lagrangian scheme and some of its associated properties.

1.2 Presentation of the compatbile staggered Lagrangian scheme

1.2.1 Governing equations and notation

In this chapter we mainly focus on two dimension space $\mathbb{R}^2$ paved with polygonal cells. The model equations under consideration are the hydrodynamics equations for which we neglect viscous stress and heat conduction. In other words we mainly focus on the gas dynamics equations expressed as conservation laws of mass, momentum and total energy.

In Lagrangian framework, the two-dimensional gas dynamics equations write

$$\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) - \nabla \cdot \boldsymbol{U} = 0, \tag{1.1}$$

$$\rho \frac{d}{dt}\boldsymbol{U} + \boldsymbol{\nabla} P = \boldsymbol{0}, \tag{1.2}$$

$$\rho \frac{d}{dt} E + \nabla \cdot (P\boldsymbol{U}) = 0, \tag{1.3}$$

where ρ is the density, $\boldsymbol{U}$ the velocity, E the specific total energy and $\frac{d}{dt}$ denotes the material derivative. The first equation expresses the volume conservation equation, whereas the second and third ones are the momentum and total energy conservation equations. Volume conservation equation is often referred to as the Geometric Conservation Law (GCL). The previous system is equipped with a thermodynamics closure (equation of state EOS) $P = P(\rho, \varepsilon)$, where the specific internal energy is given by $\varepsilon = E - \frac{\boldsymbol{U}^2}{2}$. Note that for smooth solutions energy equation can be rewritten as

$$\rho \frac{d}{dt}\varepsilon + P\nabla \cdot \boldsymbol{U} = 0, \tag{1.4}$$

and, substituting volume equation yields

$$\rho \frac{d}{dt}\varepsilon + P\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) = 0. \tag{1.5}$$

Recalling Gibbs relation for temperature T and specific entropy S : $TdS = d\varepsilon + Pd\left(\frac{1}{\rho}\right)$, and the second law of thermodynamics, namely $T\frac{dS}{dt} \geq 0$, implies that for non-smooth flows the following relation holds :

$$\rho \frac{d}{dt}\varepsilon + P\nabla \cdot \boldsymbol{U} \geq 0. \tag{1.6}$$

As a consequence, internal energy equation can be viewed as an entropy evolution equation since

$$\rho \frac{d}{dt}\varepsilon + P\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) \geq 0. \tag{1.7}$$

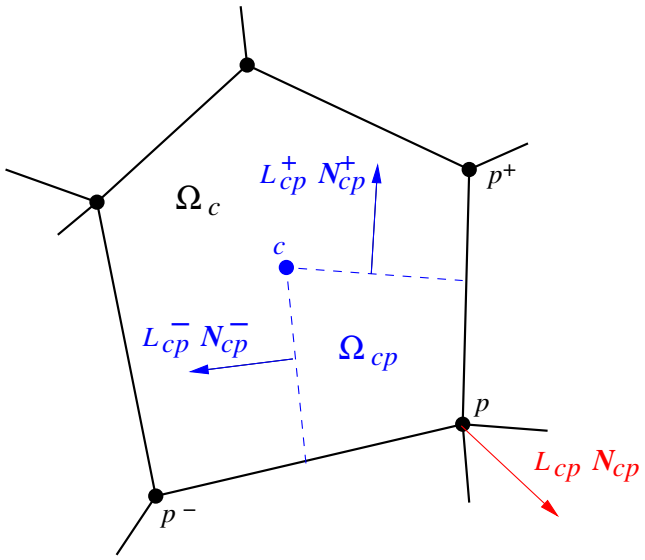


FIGURE 1.1 – *Fragment of a polygonal grid. Position and velocity are defined at grid points while thermodynamic variables are located at cell centers. A polygonal cell, Ω_c, is subdivided into subcells Ω_{cp}. Points are denoted by subscript p and counterclockwise ordered p^-, p, p^+.*

The previous system (1.1-1.3) can therefore be rewritten as a non-conservative system by replacing the energy equation by (1.6). The last equations are the trajectory equations

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{U}(\boldsymbol{X}(t), t), \qquad \boldsymbol{X}(0) = \boldsymbol{x}, \tag{1.8}$$

expressing the Lagrangian motion of any point initially located at position $\boldsymbol{x}$.

We use a staggered placement of variables in which position and velocity are defined at grid points while thermodynamic variables are located at cell centers, refer to Fig. 1.1. An unstructured grid consisting of a collection of non-overlapping polygons is considered. Each polygonal cell is assigned a unique index c and is denoted Ω_c. Each vertex/point of the mesh is assigned a unique index p and we denote $\mathcal{C}(p)$ the set of cells sharing a particular vertex p. Each polygonal cell is subdivided into a set of subcells ; each being uniquely defined by a pair of indices c and p and denoted Ω_{cp}. This subcell is constructed by connecting the cell center of Ω_c to the mid-points of cell edges impinging at point p. The union of subcells Ω_{cp} that share a particular vertex p allows to define the dual vertex-centered cell Ω_p related to point p with $\Omega_p = \bigcup_{c\in\mathcal{C}(p)} \Omega_{cp}$. Using the previous notation, we can define the primary grid $\bigcup_c \Omega_c$ and the dual grid $\bigcup_p \Omega_p$. The volumes of the primary and dual cells are functions of time t. For a vertex p of cell Ω_c we denote its previous and next vertices by p^- and p^+. Here, following [55], we make the fundamental but questionable assumption that the subcells are Lagrangian volumes. This means that the subcell mass m_{cp} is constant in time. Therefore, being given the initial density field $\rho^0(\boldsymbol{x})$ one deduces the initial mean density in cell c

$$\rho_c^0 = \int_{\Omega_c(0)} \rho^0(\boldsymbol{x}) d\boldsymbol{x} / V_c^0, \tag{1.9}$$

where V_c^0 is the volume of cell Ω_c at time $t = 0$. Subcell mass is defined as $m_{cp} = \rho_c^0\ V_{cp}^0$, where V_{cp}^0 is the initial volume of subcell Ω_{cp}. By summation of Lagrangian subcell masses one defines

Lagrangian cell/point masses as

$$m_c = \sum_{p \in \mathcal{P}(c)} m_{cp}, \qquad m_p = \sum_{c \in \mathcal{C}(p)} m_{cp}, \tag{1.10}$$

where $\mathcal{P}(c)$ is the set of counterclockwise ordered vertices of cell c.

1.2.2 Compatible discretization

We construct staggered Lagrangian schemes using the well known methodology of compatible discretization which has been presented in [74, 55, 12]. The cornerstone of this type of discretization is the subcell force that acts from subcell cp onto point p, see Fig. 1.1. In this approach, the discretization of the internal energy equation in terms of subcell forces is deduced from total energy conservation. Here, we reproduce the derivation of Maire [42] starting from a generic abstract form of the subcell force so that an entropy inequality is satisfied, which ensures that kinetic energy is dissipated into internal energy through shock waves. The subcell force writes as a pressure contribution plus a viscous contribution also known as artificial viscosity or pseudo-viscosity.

Geometric Conservation Law (GCL). Here, we use a discretization of the volume equation (1.1) that is compatible with the GCL. By GCL compatibility we mean that we are deriving a discrete divergence operator for the volume equation by requiring consistency of the divergence of the velocity field with the time rate of change of volume of the cell, refer to [76]. By noticing that $m_c = \rho_c V_c$, where $\rho_c = \rho_c(t)$ and $V_c = V_c(t)$ are the cell density and volume, we can write

$$m_c \frac{d}{dt}\left(\frac{1}{\rho_c}\right) = \frac{d}{dt} V_c,$$

using the fact that the cell mass is constant in time. Moreover, remarking that the cell volume can be expressed as a function of the position vectors of its vertices as follows

$$V_c(t) = \sum_{p \in \mathcal{P}(c)} \frac{1}{2} \left(\boldsymbol{X}_p \times \boldsymbol{X}_{p^+} \right) \cdot \boldsymbol{e}_z,$$

where $\boldsymbol{e}_z$ is the unit vector of the canonical basis in z direction. We deduce that the time rate of change of the cell volume writes

$$\frac{d}{dt} V_c = \sum_{p \in \mathcal{P}(c)} \boldsymbol{\nabla}_{\boldsymbol{X}_p} V_c \cdot \frac{d}{dt} \boldsymbol{X}_p.$$

Here, we have simply applied the chain rule differentiation. Setting $\frac{d}{dt}\boldsymbol{X}_p = \boldsymbol{U}_p$ where $\boldsymbol{U}_p$ is the vertex velocity, we rewrite this last equation as

$$\boxed{\frac{d}{dt} V_c - \sum_{p \in \mathcal{P}(c)} L_{cp} \boldsymbol{N}_{cp} \cdot \boldsymbol{U}_p = 0,} \tag{1.11}$$

where $L_{cp}\boldsymbol{N}_{cp}$, with $\boldsymbol{N}_{cp}^2 = 1$, stands for the corner vector defined by

$$L_{cp}\boldsymbol{N}_{cp} = \boldsymbol{\nabla}_{\boldsymbol{X}_p} V_c. \tag{1.12}$$

This corner vector is a fundamental geometric object which is nothing but the gradient of the cell volume at point p. Its explicit expression in terms of points coordinates writes

$$L_{cp}\boldsymbol{N}_{cp} = \frac{1}{2}\begin{pmatrix} Y_{p^+} - Y_{p^-} \\ -(X_{p^+} - X_{p^-}) \end{pmatrix},$$

where (X_p, Y_p) denote the coordinate of the position vector $\boldsymbol{X}_p$. This kind of formalism is well known and has been used in staggered and cell-centered (free Lagrange) discretizations long time ago [76, 77]. We note that (1.11) is compatible with the discrete version of the trajectory equation (1.8)

$$\frac{d}{dt}\boldsymbol{X}_p = \boldsymbol{U}_p, \quad \boldsymbol{X}_p(0) = \boldsymbol{x}_p.$$

This leads to a compatible definition of the discrete divergence operator over cell c as

$$(\nabla \cdot \boldsymbol{U})_c = \frac{1}{V_c}\sum_{p\in\mathcal{P}(c)} L_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p. \tag{1.13}$$

We also emphasize that the corner vector $L_{cp}\boldsymbol{N}_{cp}$ satisfies the fundamental geometric identity

$$\sum_{p\in\mathcal{P}(c)} L_{cp}\boldsymbol{N}_{cp} = \boldsymbol{0}, \tag{1.14}$$

which is equivalent to the result that the summation of the outward normals to a closed polygonal contour is equal to zero.

Finally, we have obtained a compatible discretization of the volume equation (1.1) which writes

$$m_c\frac{d}{dt}\left(\frac{1}{\rho_c}\right) - \sum_{p\in\mathcal{P}(c)} L_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p = 0. \tag{1.15}$$

Momentum equation. The semi-discrete momentum equation over the dual cell Ω_p writes

$$\boxed{m_p\frac{d}{dt}\boldsymbol{U}_p + \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp} = \boldsymbol{0}.} \tag{1.16}$$

Here, $\boldsymbol{F}_{cp}$ is the subcell force from cell c that acts on point p, which is defined by

$$\boldsymbol{F}_{cp} = \int_{\partial\Omega_p(t)\cap\Omega_c(t)} P\boldsymbol{N}\mathrm{d}l, \tag{1.17}$$

where $\mathrm{d}l$ is an infinitesimal length. Momentum equation (1.16) is nothing but the Newton law applied to particle of mass m_p moving with velocity $\boldsymbol{U}_p$.

Specific internal energy equation. Here we derive a semi-discrete internal energy equation that ensures total energy conservation using the concept of subcell force, following the approach initially described in [55]. Let us introduce total kinetic energy and total internal energy

$$\mathcal{K}(t) = \sum_p \frac{1}{2}m_p\boldsymbol{U}_p^2(t), \tag{1.18}$$

$$\mathcal{E}(t) = \sum_c m_c\varepsilon_c(t), \tag{1.19}$$

where ε_c is the cell averaged specific internal energy. Total energy is then defined as

$$E(t) = \mathcal{K}(t) + \mathcal{E}(t). \tag{1.20}$$

The conservation of total energy without taking into account boundary conditions simply writes

$$\frac{d}{dt}E = \frac{d}{dt}\mathcal{K} + \frac{d}{dt}\mathcal{E} = 0. \tag{1.21}$$

The substitution of kinetic and internal energies recalling that cell/point masses are Lagrangian objects, i.e. they not depend on time, yields

$$\frac{d}{dt}\mathcal{K} + \frac{d}{dt}\mathcal{E} = \sum_c m_c \frac{d}{dt}\varepsilon_c + \sum_p m_p \frac{d}{dt}\boldsymbol{U}_p \cdot \boldsymbol{U}_p,$$

then using (1.21) one deduces

$$\sum_c m_c \frac{d}{dt}\varepsilon_c + \sum_p m_p \frac{d}{dt}\boldsymbol{U}_p \cdot \boldsymbol{U}_p = 0.$$

Using the semi-discrete momentum equation (1.16) yields

$$\sum_c m_c \frac{d}{dt}\varepsilon_c - \sum_p \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp} \cdot \boldsymbol{U}_p = 0,$$

and interchanging the order in the double sum one finally gets

$$\sum_c \left(m_c \frac{d}{dt}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp} \cdot \boldsymbol{U}_p \right) = 0. \tag{1.22}$$

A sufficient condition for total energy conservation is obtained by requiring the previous equation to hold in each cell c

$$\boxed{m_c \frac{d}{dt}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp} \cdot \boldsymbol{U}_p = 0.} \tag{1.23}$$

Notice that this choice is not unique and other discretizations would provide the total energy conservation given a definition of total enery such as (1.20). Any of such discretization is refered to as a "compatible discretization" under Caramana's appelation. The word "discrete" also used in Caramana's phrasing refers to the fact that the discrete equations are rather deduced than derived from the constinuous equations.
Once the subcell force is known, then momentum and internal energy can be updated using equations (1.16) and (1.23).

Summary of the compatible discretization. We summarize the semi-discrete equations that govern the time rate of change of the primary variables $(\frac{1}{\rho}_c, \boldsymbol{U}_p, \varepsilon_c)$

$$m_c \frac{d}{dt}(\frac{1}{\rho}_c) - \sum_{p\in\mathcal{P}(c)} L_{cp}\boldsymbol{N}_{cp} \cdot \boldsymbol{U}_p = 0, \tag{1.24}$$

$$m_p \frac{d}{dt}\boldsymbol{U}_p + \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp} = \boldsymbol{0}, \tag{1.25}$$

$$m_c \frac{d}{dt}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp} \cdot \boldsymbol{U}_p = 0. \tag{1.26}$$

We point out that the mesh motion is given by the trajectory equations

$$\frac{d}{dt}\boldsymbol{X}_p = \boldsymbol{U}_p(\boldsymbol{X}_p(t), t), \qquad \boldsymbol{X}_p(0) \quad = \boldsymbol{x}_p, \tag{1.27}$$

which is compatible with the GCL. The thermodynamic closure is given by the equation of state which writes $P_c = P(\rho_c, \varepsilon_c)$. We emphasize that this subcell-based compatible discretization ensures total energy conservation regardless of the subcell force form.
Although our description of the staggered compatible Lagrangian scheme is different from the descriptions of Burton [74, 78] or Caramana [55, 12] it shares with them the same fundamental objects : Cell/point masses and subcell force such that the compatible discretization intrinsically leads to the conservation of total energy by construction. What must be the components of a sucell force is almost left to the developer (or user). Under this subcell force concept many different physical or numerical effects are in fact gathered. First the pressure force takes into account the $\boldsymbol{\nabla} P$ term in (1.2). Then the artificial viscosity force is designed to handle shock wave and steep fronts and as such stabilizes the scheme. It also assures (1.6) to hold. The anti-hourglass force is a pure numerical concept which is meant to fight back parasitical grid motion known as "hourglass modes" [56]. Elasto-plasticity terms can be expressed into this force formalism [79], slide-line [80, 25] or internal boundary conditions also. In fact many physical models can be recast into this fruitful compatible discretization [1], and, no matter what is put under this definition, conservation is preserved.

1.2.3 Subcell forces

Let us provide in this section a definition of the subcell force invoking Galilean invariance and thermodynamic consistency. Subcell pressure force is then deduced and several artificial viscous forces and anti-hourglass subpressure force are further described.

Galilean invariance is a principle of relativity which states that the fundamental laws of physics are the same in all inertial frames. It is one of the key requirements of many physical models adopted in theoretical and computational mechanics. To fulfill Galilean invariance, the previously derived specific internal energy equation (1.23) must remain unchanged under a uniform translation of frame. Let $\boldsymbol{A}$ denote the uniform translation velocity. Then equation (1.23) transforms into

$$m_c \frac{d}{dt}\varepsilon_c - \sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{cp} \cdot (\boldsymbol{U}_p + \boldsymbol{A}) = 0.$$

By substituting (1.23) into this last equation leads to

$$\sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{cp} \cdot \boldsymbol{A} = 0,$$

which must hold for all vectors $\boldsymbol{A}$. Therefore, specific internal energy equation remains invariant under uniform translation if and only if

$$\boxed{\sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{cp} = \boldsymbol{0}.} \tag{1.28}$$

1. This is probably one reason why this discretization has been successful amongst physicists from national laboratories along with the fact that using artificial viscosity methods are inherently simpler than operator splitting methods (such as Godunov methods) in that the level of numerical complexity does not increase as the number of dimensions and/or the amount of physics included increases.

We note that this result has been already quoted in [12] page 576 and probably elsewhere before. This condition also implies total momentum conservation without taking into account boundary conditions. To demonstrate this, it suffices to time-differentiate the global momentum defined as

$$\boldsymbol{\mathcal{Q}} = \sum_p m_p \boldsymbol{U}_p, \tag{1.29}$$

to obtain

$$\begin{aligned} \frac{d}{dt}\boldsymbol{\mathcal{Q}} &= \sum_p m_p \frac{d}{dt}\boldsymbol{U}_p \\ &= -\sum_p \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp}, \quad \text{thanks to momentum equation,} \\ &= -\sum_c \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}, \quad \text{by interchanging the double sums.} \end{aligned} \tag{1.30}$$

Thus, $\frac{d}{dt}\boldsymbol{\mathcal{Q}} = \boldsymbol{0}$ due to condition (1.28), which completes the proof.

A corollary of the Galilean invariance condition is that specific internal energy equation (1.23) can also be rewritten into

$$m_c \frac{d}{dt}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}\cdot(\boldsymbol{U}_p - \boldsymbol{U}_c) = 0, \tag{1.31}$$

where $\boldsymbol{U}_c$ is any arbitrary piecewise constant cell based velocity. This equation will be used in the next section.

Subcell pressure force

Let us investigate the thermodynamic consistency of the semi-discrete scheme by computing the time rate of change of entropy in a cell c. Using Gibbs formula, one gets

$$m_c T_c \frac{d}{dt} S_c = m_c \left[\frac{d}{dt}\varepsilon_c + P_c \frac{d}{dt}\left(\frac{1}{\rho_c}\right)\right], \tag{1.32}$$

where S_c and T_c are the specific entropy and temperature of cell c. Substituting into (1.32) the specific internal energy equation (1.23) and the volume equation (1.15) yields

$$m_c T_c \frac{d}{dt} S_c = \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p + P_c \left(\sum_{p\in\mathcal{P}(c)} L_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p \right) \tag{1.33}$$

$$= \sum_{p\in\mathcal{P}(c)} (\boldsymbol{F}_{cp} + L_{cp}P_c\boldsymbol{N}_{cp})\cdot\boldsymbol{U}_p. \tag{1.34}$$

For smooth flow the right hand side of the last equation must be zero leading to the form of the *subcell pressure force* as

$$\boxed{\boldsymbol{F}_{cp}^{\text{press}} = -L_{cp}\ P_c\ \boldsymbol{N}_{cp},} \tag{1.35}$$

which corresponds to the discretization of (1.17). One trivially verifies that

$$\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}^{\text{press}} = -P_c \sum_{p\in\mathcal{P}(c)} L_{cp}\boldsymbol{N}_{cp} = \boldsymbol{0}, \tag{1.36}$$

thanks to identity (1.14), which, as a side effect implies that momentum conservation is ensured.

Artificial viscous force

While cell-centered Lagrangian schemes rely on some sort of Riemann solvers to add numerical viscosity [43, 50, 51], staggered Lagrangian schemes historicaly rely on artificial viscosity [67]. The artificial viscosity, otherwize called pseudo-viscosity, is historically refered to as the 'q' term. The illuminating idea of von Neumann and Richtmyer was to introduce a purely artificial dissipative mechanism of such a form and strength that the shock transition would be a smooth one extending over a small number of cell length, and then to include this dissipation into the finite difference equations, [67] page 312 CHAP. 12 SEC. 12.10. Only a linear term introduced by Landshoff in [81] was present in the form of the original artificial viscosity. As a consequence the thickness of the 'transition layer' (i.e. the shock spreading) was varying with the shock strength approaching zero for a very strong shock and tending to infinity for a very weak one. However their wish was to have a constant thickness of the shock spreading, so von Neumann and Richtmyer added a quadratic term which they interpret to be as "using a small viscosity coefficient for weak shocks" ([67] page 312). In an unpublished work from Los Alamos from the 50's Rosenbluth suggested that the artificial viscosity should be zero when the fluid is undergoing an expansion, this 'trick' is nowadays known as the 'artificial viscosity switch'.
Since the seminal work of von Neumann and Richtmyer there is still no universally satisfactory form of the artificial viscosity suitable for all problems although many authors contributed to the subject. As instance Schultz [82, 83] introduced the nowadays known "edge-based" artificial viscosity further popularized by Caramana in [57], Richards [84], Wilkins analysed the viscosity coefficients [85] in the U.S.A, this work was also previously achieved by Kurapatenko in Soviet Union [86], Christensen interpreted the artificial viscosity as a Riemann solver [87] related to the works of Dukowicz [88, 89], Noh studied the errors that arise when using artificial viscosity in [90], Benson revised most of these works in the review paper [52] and also contributed to flux-limited shock viscosity in [91], Shashkov proposed a tensor extension of the artificial viscosity using mimetic finite difference method in [53, 92] even if several attempts of 'tensorization' have been tried before [93, 83] and, more recently, by Rieben and Kolev [62] and Owen [94] contributed to the subject in slighly different contexts [2]
The fact that different forms for this force are often utilized depending on the type of problem being studied is the major remaining deficiency of this class of hydrodynamics methods. In the following we briefly present three models to compute *artificial viscous subcell forces*.

Bulk viscosity based on original works [67, 81] considers a cell centered "pseudo-pressure"

$$q_c = c_1 \rho_c a_c^* |\Delta \boldsymbol{U}| + c_2 \rho_c (\Delta \boldsymbol{U})^2, \tag{1.37}$$

where $c_1 \leq c_2$ are two constants of the order of unity and $\Delta \boldsymbol{U}$ is a measure of the velocity difference over the cell and a_c^* and ρ_c are respectively the sound-speed and density in cell c. However the Kurapatenko combinaison [86] of linear and non-linear terms for material with ratio of specific heats γ is often used instead of (1.37)

$$q_c^{\text{Kur}} = \rho_c \left(c_2 \frac{\gamma - 1}{4} |\Delta \boldsymbol{U}| + \sqrt{c_2^2 \left(\frac{\gamma - 1}{4} \right)^2 (\Delta \boldsymbol{U})^2 + c_1^2 (a_c^*)^2} \right) |\Delta \boldsymbol{U}|. \tag{1.38}$$

This expression has been derived for an ideal equation of state to determine the form of the term that, quoting Caramana in [57] "must be added to the pressure in front of a steady-state

2. This list of published works is not entended to be exhautive, rather the works have been chosen to spread along the years from the 50's up to now and the authors have been cited as to give to the reader key names in the field. Following the cited works from these researchers surely provides an almost exhaustive view of the artificial viscosity quest, the rest being unpublished and classified works kept in library of national laboratories.

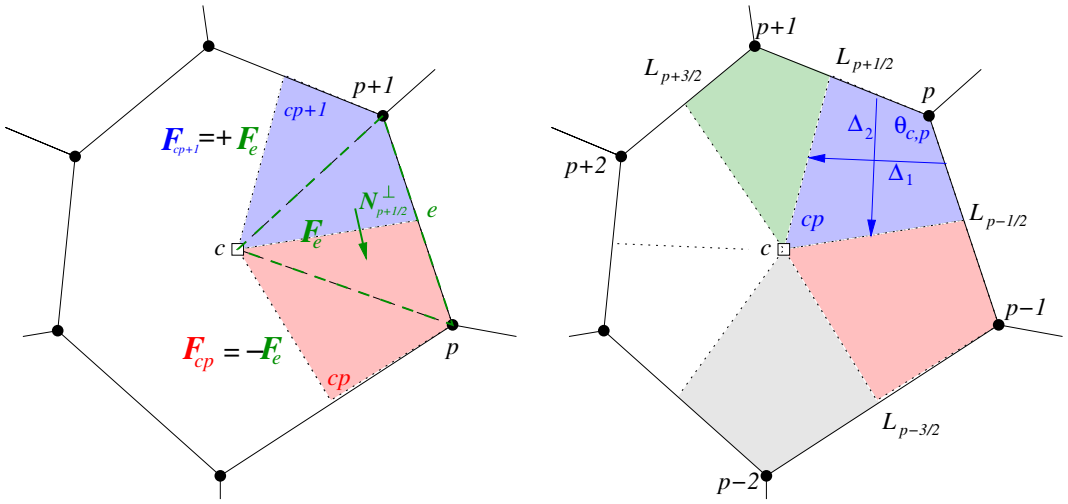


Figure 1.2 – *Notation involved for the artificial viscosity models — Left : the edge viscous force is first computed related to the triangular zone associated to an edge e (green triangle). Then this edge-based viscous force is associated to subcell cp with a + sign and cp′ with a minus sign — Right : three tensor viscous force*

shock in order to achieve the pressure behind the shock, using the jump conditions. In this instance ρ_c and a_c^* are the density and sound speed ahead of the shock, and $\Delta \boldsymbol{U}$ is the velocity jump across it". The use of (1.37) or (1.38) consists of adding the 'q' force to the pressure force (1.35) to get the following viscous subcell force

$$\boxed{\boldsymbol{F}_{cp}^{\mathrm{q}} = \begin{cases} -L_{cp}\, q_c^{\mathrm{Kur}}\, \boldsymbol{N}_{cp} & \text{if} \quad (\nabla \cdot \boldsymbol{U})_c \leq 0 \\ 0 & \text{else} \end{cases}} \tag{1.39}$$

Note that either q or q^{Kur} are positive constants over the cell. This form of artificial viscosity is dissipative because recalling (1.34) with the artificial viscous force (1.39) we have that for a non-smooth flow the right hand side of this equation must be positive (that is to say the scheme must be dissipative), and we convince ourselves that

$$\begin{aligned} m_c T_c \frac{d}{dt} S_c &= \sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{cp}^{\mathrm{q}} \cdot \boldsymbol{U}_p \\ &= \sum_{p \in \mathcal{P}(c)} -L_{cp}\, q_c^{\mathrm{Kur}}\, \boldsymbol{N}_{cp} \cdot \boldsymbol{U}_p \\ &= -q_c^{\mathrm{Kur}} \sum_{p \in \mathcal{P}(c)} L_{cp} \boldsymbol{N}_{cp} \cdot \boldsymbol{U}_p \\ &= -q_c^{\mathrm{Kur}}\, V_c (\nabla \cdot \boldsymbol{U})_c \geq 0, \end{aligned}$$

due to the fact that $(\nabla \cdot \boldsymbol{U})_c \leq 0$.
This formulation has been widely used but its main drawbacks lay in its inability to vanish for rigid rotation or uniform compression or (sometimes depending on the implementation) along a front of constant phase, see discussion in [57] page 85 for more details.

Edge viscosity based on [82, 83] and popularized in [57] is based on the computation of a 'q' term for each edge of a cell. A clear description of the edge viscosity forces is provided in Appendix A of [57]. Here we only para-phrase this Appendix. Let us consider one edge e defined by two

succesive points $p, p+1$ of cell c, see Fig.1.2, such that the velocity difference over this edge is $\Delta \boldsymbol{U}_e = \boldsymbol{U}_{p+1} - \boldsymbol{U}_p$ and the associated unit vector is $\widehat{\Delta \boldsymbol{U}}_e$. Let us first define the edge-based viscous force as

$$\boxed{\boldsymbol{F}_e = \begin{cases} (1-\psi_e) q_e^{\text{Kur}} (\Delta \boldsymbol{U}_e \cdot \boldsymbol{N}_{p+1/2}^{\perp}) \widehat{\Delta \boldsymbol{U}}_e & \text{if} \quad \Delta \boldsymbol{U}_e \cdot \boldsymbol{N}_{p+1/2}^{\perp} \leq 0, \\ 0 & \text{if} \quad \Delta \boldsymbol{U}_e \cdot \boldsymbol{N}_{p+1/2}^{\perp} > 0 \end{cases}} \tag{1.40}$$

where

$$q_e^{\text{Kur}} = \rho_e \left(c_2 \frac{\gamma - 1}{4} |\Delta \boldsymbol{U}_e| + \sqrt{c_2^2 \left(\frac{\gamma - 1}{4} \right)^2 (\Delta \boldsymbol{U}_e)^2 + c_1^2 (a_e^*)^2} \right), \tag{1.41}$$

and $\boldsymbol{N}_{p+1/2} = \frac{\boldsymbol{X}_{p+1} - \boldsymbol{X}_p}{|\boldsymbol{X}_{p+1} - \boldsymbol{X}_p|}$ is the unit normal along the edge direction and $\boldsymbol{N}_{p+1/2}^{\perp}$ is the perpendicular unit vector to $\boldsymbol{N}_{p+1/2}$. ρ_e and a_e^* are edge-based density and sound-speed respecively which can be computed as

$$\rho_e = \frac{2 \rho_p \rho_{p+1}}{\rho_p + \rho_{p+1}}, \qquad a_e^* = \min(a_p^*, a_{p+1}^*). \tag{1.42}$$

Moreover $0 \leq \psi_e \leq 1$ is the edge limiter dedicated to make the artificial viscosity to vanish for uniform compression, rigid rotation, and along a front of constant phase. We refer the reader to [57] for the exact definition and calculation of ψ_e. Finally the sign of $(\Delta \boldsymbol{U}_e \cdot \boldsymbol{N}_{p+1/2}^{\perp})$ represents the "switch" to turn off the artificial viscosity for expansion seen from the edge e. For a zone under compression, $(\nabla \cdot \boldsymbol{U})_c \leq 0$, for the triangular subzonal edge of cell c to be under compression we postulate the condition $\Delta \boldsymbol{U}_e \cdot \boldsymbol{N}_{p+1/2}^{\perp} \leq 0$. To build a subcell viscous force it remains to distribute $\boldsymbol{F}_e$ between the two subcells cp and $cp+1$, this is brought about by setting $\boldsymbol{F}_{cp+1}^q = -\boldsymbol{F}_e$ and $\boldsymbol{F}_{cp}^q = +\boldsymbol{F}_e$ noticing that a contribution with a minus sign from left neighbor edge is also associated to subcell force $\boldsymbol{F}_{cp}^q$, see Fig.1.2-left.
Dissipativity in this semi-discrete form is ensured because

$$\begin{aligned}
m_c T_c \frac{d}{dt} S_c &= \sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{cp}^{\text{q}} \cdot \boldsymbol{U}_p \\
&= \sum_{p \in \mathcal{P}(c)} (\boldsymbol{F}_e^{\text{right}} - \boldsymbol{F}_e^{\text{left}}) \cdot \boldsymbol{U}_p \qquad \longleftarrow \text{two edges imping. on } p \\
&= \sum_{e \in \mathcal{E}(c)} \boldsymbol{F}_e \cdot (\boldsymbol{U}_p - \boldsymbol{U}_{p+1}) \qquad \longleftarrow \text{switch to sum over edges} \\
&= \sum_{e \in \mathcal{E}(c)} (1-\psi_e) q_e^{\text{Kur}} (\Delta \boldsymbol{U}_e \cdot \boldsymbol{N}_{p+1/2}^{\perp}) \widehat{\Delta \boldsymbol{U}}_e \cdot (-\Delta \boldsymbol{U}_e) \\
&= -\sum_{e \in \mathcal{E}(c)} \underbrace{\frac{(1-\psi_e) q_e^{\text{Kur}}}{|\Delta \boldsymbol{U}_e|}}_{\geq 0} \underbrace{(\Delta \boldsymbol{U}_e \cdot \boldsymbol{N}_{p+1/2}^{\perp})}_{\leq 0} \quad \longleftarrow \text{because } \widehat{\Delta \boldsymbol{U}}_e = \frac{\Delta \boldsymbol{U}_e}{|\Delta \boldsymbol{U}_e|} \\
&\geq 0.
\end{aligned}$$

The main drawback of the artificial viscosity model is the occurence of "spurious jets" along axes as instance in the Noh problem on Cartesian grid, see Fig.1.8 left panel in section 1.4.1.

Tensor viscosity based on [53] where the entire theory is described. Instead of reproducing the derivation of such tensorial subcell based artificial viscosity force we give its final form

$$\boxed{\boldsymbol{F}_{cp}^{q} = V_c \left[\frac{1}{L_{p+1/2}} \left\{ \boldsymbol{R}_{p+1} + \boldsymbol{R}_p \right\} - \frac{1}{L_{p-1/2}} \left\{ \boldsymbol{Q}_p + \boldsymbol{Q}_{p-1} \right\} \right],} \tag{1.43}$$

where

$$\begin{aligned}
\boldsymbol{R}_{p+1} &= \frac{W_{c,p+1}}{\sin^2 \theta_{c,p+1}} \left(\mu_{c,p+1} \boldsymbol{G}_{p+1/2} + \cos\theta_{c,p+1}\ \mu_{c,p+1} \boldsymbol{G}_{p+3/2} \right), \\
\boldsymbol{R}_{p} &= \frac{W_{c,p}}{\sin^2 \theta_{c,p}} \left(\mu_{c,p} \boldsymbol{G}_{p+1/2} + \cos\theta_{c,p}\ \mu_{c,p} \boldsymbol{G}_{p-1/2} \right), \\
\boldsymbol{Q}_{p} &= \frac{W_{c,p}}{\sin^2 \theta_{c,p}} \left(\mu_{c,p} \boldsymbol{G}_{p-1/2} + \cos\theta_{c,p}\ \mu_{c,p} \boldsymbol{G}_{p+1/2} \right), \\
\boldsymbol{Q}_{p-1} &= \frac{W_{c,p-1}}{\sin^2 \theta_{c,p-1}} \left(\mu_{c,p-1} \boldsymbol{G}_{p-1/2} + \cos\theta_{c,p-1}\ \mu_{c,p-1} \boldsymbol{G}_{p-3/2} \right).
\end{aligned}$$

Here the notation is relative to point p which is a vertex of cell c, see Fig. 1.2-right, the previous points are indexed $p-2, p-1$, the next ones $p+1, p+2$. The edge connecting p and $p+1$ is indexed $p+1/2$ and the unit vector along this edge is refered to as $\boldsymbol{T}_{p+1/2}$ and its length is $L_{p+1/2}$. $\theta_{c,p}$ is the angle between the two edges of cell c meeting at point p. Moreover for all edge we define

$$\boldsymbol{G}_{p+1/2} = \frac{\boldsymbol{U}_{p+1} - \boldsymbol{U}_p}{L_{p+1/2}}, \tag{1.44}$$

and W's are some weights satisfying $W_{c,p} \geq 0$ and $\sum_{p \in \mathcal{P}(c)} W_{c,p} = 1$. Usually, for a quadrilateral cell, $W_{c,p}$ is defined as one half the area of the triangle in cell c which contains the angle at point p divided by the cell volume. For non-quadrilateral cells normalization is needed. It remains to define the μs which are some viscosity coefficients, a kind of Kurapatenko q term,

$$\mu_{cp} = (1 - \psi_{cp}) \rho_{cp} \left(c_2 \frac{\gamma+1}{4} |\Delta \boldsymbol{U}_{cp}| + \sqrt{c_2^2 \left(\frac{\gamma+1}{4} \right)^2 (\Delta \boldsymbol{U}_{cp})^2 + c_1^2 (a_c^*)^2} \right) l_{cp}. \tag{1.45}$$

This expression requires the definition of a velocity jump $\Delta \boldsymbol{U}_{cp}$ and a characteristic length l_{cp} in subcell cp. The definition of these values is a major source of difficulties for multi-dimensional artificial viscosity. Naive definitions result in instabilities for large aspect ratio cells. In [53] the authors have found a length definition that does not cause any problem for large aspect ratios, nor when a small change in velocity or geometry can result in large change in the length of velocity terms,

$$l_{cp} = \begin{cases} 2\sqrt{V_{cp}} \sqrt{\frac{|\Delta_1|}{|\Delta_2|}} & \text{if} \quad \widehat{\Delta_1} \cdot \widehat{\boldsymbol{U}_{av}} > \widehat{\Delta_2} \cdot \widehat{\boldsymbol{U}_{av}} \\ 2\sqrt{V_{cp}} \sqrt{\frac{|\Delta_2|}{|\Delta_1|}} & \text{if} \quad \widehat{\Delta_1} \cdot \widehat{\boldsymbol{U}_{av}} \leq \widehat{\Delta_2} \cdot \widehat{\boldsymbol{U}_{av}} \end{cases} \tag{1.46}$$

where Δ_1, Δ_2 are the lengths across the subcell and the hat symbol refers to the associated unit vectors, see Fig. 1.2-right. Moreover we define $\boldsymbol{U}_{av} = \frac{1}{4}(\boldsymbol{U}_{p-1/2} + \boldsymbol{U}_p + \boldsymbol{U}_{p+1/2} + \boldsymbol{U}_c)$ where

$\boldsymbol{U}_{p+1/2} = \frac{1}{2}(\boldsymbol{U}_p + \boldsymbol{U}_{p+1})$ and the cell-centered velocity is given by $\boldsymbol{U}_c = \frac{1}{|\mathcal{P}(c)|} \sum_{p\in\mathcal{P}(c)} \boldsymbol{U}_p$. Finally the velocity jump is taken as the maximum velocity jump across the subcell

$$\Delta \boldsymbol{U}_{cp} = 2\max(|\Delta \boldsymbol{U}_1|, |\Delta \boldsymbol{U}_2|), \tag{1.47}$$

where $\Delta \boldsymbol{U}_{1,2}$ are the velocity jumps along $\Delta_{1,2}$ respectively. Finally a switch is added the following way

$$\Delta \boldsymbol{U}_{cp} = \begin{cases} 2\max(|\Delta \boldsymbol{U}_1|, |\Delta \boldsymbol{U}_2|) & \text{if} \quad (\nabla \cdot \boldsymbol{U}_{cp}) \leq 0 \\ 0 & \text{else} \end{cases}, \tag{1.48}$$

where $\nabla \cdot \boldsymbol{U}_{cp}$ is a measure of the velocity divergence within subcell cp. The brute force proof of dissipativity of tensor viscosity is demanding and we end up with conditions which are not explicitely set in the original work [53]

$$\begin{aligned} m_c T_c \frac{d}{dt} S_c &= \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}^{\mathrm{q}} \cdot \boldsymbol{U}_p \\ &= -\sum_{e\in\mathcal{E}(c), e=[\boldsymbol{X}_p, \boldsymbol{X}_{p+1}]} (\boldsymbol{U}_{p+1} - \boldsymbol{U}_p) \cdot \Big(\\ &\quad (\boldsymbol{U}_p - \boldsymbol{U}_{p-1}) \frac{W_{cp}\mu_{cp}\cos\theta_{cp}}{L_{p-1/2}\sin^2\theta_{cp}} \\ &\quad + (\boldsymbol{U}_{p+1} - \boldsymbol{U}_p) \left[\frac{W_{cp}\mu_{cp}\cos\theta_{cp}}{L_{p+1/2}\sin^2\theta_{cp}} \frac{W_{cp+1}\mu_{cp+1}\cos\theta_{cp+1}}{L_{p-1/2}\sin^2\theta_{cp+1}} \right] \\ &\quad + (\boldsymbol{U}_{p+2} - \boldsymbol{U}_{p+1}) \frac{W_{cp+1}\mu_{cp+1}\cos\theta_{cp+1}}{L_{p+3/2}\sin^2\theta_{cp+1}} \Big), \end{aligned}$$

which is positive for each edge only if simultaneously

$$(\boldsymbol{U}_{p+1} - \boldsymbol{U}_p) \cdot (\boldsymbol{U}_p - \boldsymbol{U}_{p-1}) \leq 0, \quad \text{and} \quad (\boldsymbol{U}_{p+1} - \boldsymbol{U}_p) \cdot (\boldsymbol{U}_{p+2} - \boldsymbol{U}_{p+1}) \leq 0, \tag{1.49}$$

because all terms $\frac{W\mu\cos\theta}{L\sin^2\theta}$ are positive. Reasonnably we adopt (1.49) as definition of the switch $(\nabla \cdot \boldsymbol{U}_{cp}) \leq 0$ which appears in (1.48).

In the last two models of artificial viscosity, edge based and tensorial, a limiter ψ has been introduced to fulfill the wave-front invariance property. However, it has been shown in [95] that such a limiter, even very well adapted when the grid is aligned with the flow, produces some numerical artifacts for grids non aligned with the flow. Such artifacts generaly leads to numerical instabilities and evident loss of symmetry. The design of a valid limiter for non-aligned grid is still an open problem for such artificial viscosity models.

Anti-hourglass subcell force

As already quoted, following [55], we make the fundamental assumption that the subcells are Lagrangian volumes. Consequently following the compression or expansion of the cell, the subcell volume $V_{cp}(t)$ may change. Being Lagrangian, the subcell preserves its mass m_{cp}, hence its density varies as : $\rho_{cp}(t) = m_{cp}/V_{cp}(t)$. As the cell-centered specific internal energy ε_c is constant inside the cell, we use the equation of a state to define the subcell pressure as

$$P_{cp} = P(\rho_{cp}, \varepsilon_c), \qquad \text{with} \quad \rho_{cp} = m_{cp}/V_{cp}. \tag{1.50}$$

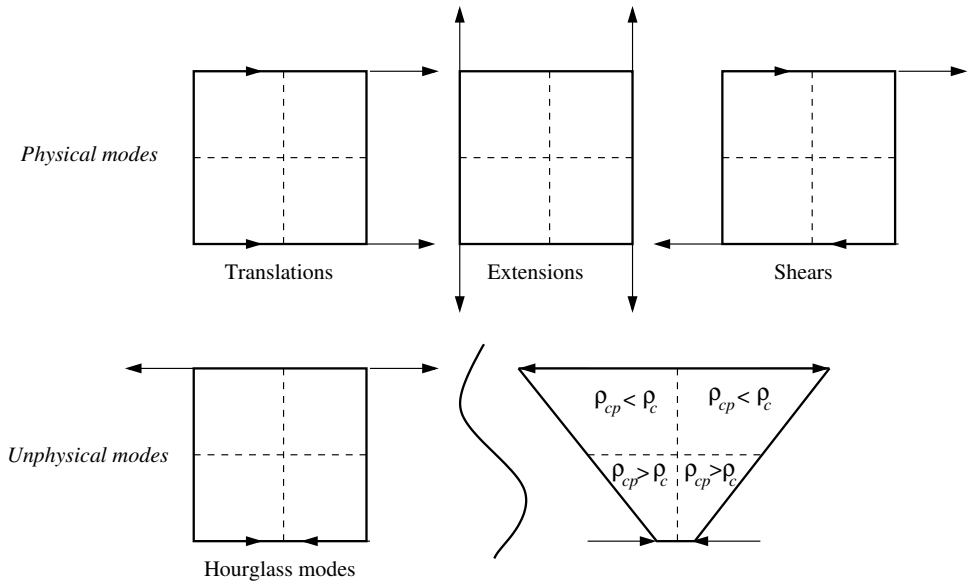


Figure 1.3 – *Physical and unphysical modes of a quadrilateral cell. Top : translations, extensions and shears (considering symmetry) — Bottom : unphysical hourglass modes. All but the two hourglass modes are physical but only for the hourglass modes do the subcell densities differ from the cell density.*

Subcell pressure force has been initially introduced by Caramana and Shashkov [56] to control artificial grid distortions, such as the hourglass modes[3]. In order to illustrate this effect let us recall that a quadrilateral cell has eight degrees of freedom : two translations, two extensions, two shears and two hourglass modes, see figure 1.3. All but the two hourglass modes are physical but only for the hourglass modes does the subcell density differ from the cell density to which it belongs. Within a cell we observe $\rho_{cp} > \rho_c$ for several subcells and $\rho_{cp} < \rho_c$ for others. The subcell pressures P_{cp} also differ from the cell pressure P_c. The subcell pressure method uses this effect to calculate subcell forces that are proportional to the difference between the subcell and the cell pressures, and oppose the hourglass motion. In this approach, the *subcell anti-houglass* force is defined as

$$\boxed{\boldsymbol{F}_{cp}^{\Delta P} = L_{cp}(P_{cp} - P_c)\boldsymbol{N}_{cp} + \frac{1}{2}\left[\left(P_{cp} - P_{cp^-}\right) L_{cp}^{-}\boldsymbol{N}_{cp}^{-} + \left(P_{cp} - P_{cp^+}\right) L_{cp}^{+}\boldsymbol{N}_{cp}^{+}\right]} \tag{1.51}$$

where P_{cp^-} and P_{cp^+} are the previous and next neighbor subcell pressures with respect to subcell cp and $L_{cp}^{\pm}\boldsymbol{N}_{cp}^{\pm}$ are the internal geometrical vector to subcell cp see Fig. 1.1. The subcell force $\boldsymbol{F}_{cp}^{\Delta P}$ is usually multiplied by a merit factor z_{merit} which ranges from 0 to 1.
Recall that conservation of global momentum, $\mathcal{Q}$ see (1.29), is implied by the relation on subcell forces (1.28), that is to say $\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp} = \boldsymbol{0}$. Let us prove that anti-hourglass subcell forces also verify

$$\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}^{\Delta P} = \boldsymbol{0}. \tag{1.52}$$

3. Other attempts to damp such artificial grid distortions can be found with artificial viscosity-like terms in Wilkins or using SUPG stabilized formulation [96].

Let us start by summing (1.51) for all points p of cell c

$$\begin{aligned}\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}^{\Delta P} &= \sum_{p\in\mathcal{P}(c)} L_{cp}(P_{cp}-P_c)\boldsymbol{N}_{cp} + \frac{1}{2}\left[\left(P_{cp}-P_{cp^-}\right)L_{cp}^-\boldsymbol{N}_{cp}^- + \left(P_{cp}-P_{cp^+}\right)L_{cp}^+\boldsymbol{N}_{cp}^+\right] \\ &= \sum_{p\in\mathcal{P}(c)} L_{cp}P_{cp}\boldsymbol{N}_{cp} + \frac{1}{2}\left[\sum_{p\in\mathcal{P}(c)}\left(P_{cp}-P_{cp^-}\right)L_{cp}^-\boldsymbol{N}_{cp}^- + \sum_{p\in\mathcal{P}(c)}\left(P_{cp}-P_{cp^+}\right)L_{cp}^+\boldsymbol{N}_{cp}^+\right],\end{aligned} \tag{1.53}$$

where we have used the geometrical identity (1.14) to deduce $\sum_{p\in\mathcal{P}(c)} L_{cp}P_c\boldsymbol{N}_{cp} = \boldsymbol{0}$. Let us focus on the square bracket terms

$$\mathcal{SB} = \frac{1}{2}\left[\sum_{p\in\mathcal{P}(c)} P_{cp}\left(L_{cp}^-\boldsymbol{N}_{cp}^- + L_{cp}^+\boldsymbol{N}_{cp}^+\right) - \sum_{p\in\mathcal{P}(c)} P_{cp^-}L_{cp}^-\boldsymbol{N}_{cp}^- - \sum_{p\in\mathcal{P}(c)} P_{cp^+}L_{cp}^+\boldsymbol{N}_{cp}^+\right],$$

Assuming periodic boundary conditions we can use index shifts in the second and third sums (p^- becomes p and p^+ becomes p) to obtain

$$\begin{aligned}\mathcal{SB} &= \frac{1}{2}\left[\sum_{p\in\mathcal{P}(c)} P_{cp}\left(L_{cp}^-\boldsymbol{N}_{cp}^- + L_{cp}^+\boldsymbol{N}_{cp}^+\right) - \sum_{p\in\mathcal{P}(c)} -P_{cp}L_{cp}^+\boldsymbol{N}_{cp}^+ - \sum_{p\in\mathcal{P}(c)} -P_{cp}L_{cp}^-\boldsymbol{N}_{cp}^-\right], \\ &= \sum_{p\in\mathcal{P}(c)} P_{cp}\left(L_{cp}^-\boldsymbol{N}_{cp}^- + L_{cp}^+\boldsymbol{N}_{cp}^+\right) \\ &= -\sum_{p\in\mathcal{P}(c)} L_{cp}P_{cp}\boldsymbol{N}_{cp}.\end{aligned} \tag{1.54}$$

Back substituting this last equation into (1.53) yields the expected result

$$\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}^{\Delta P} = \boldsymbol{0}.$$

Total subcell/nodal force

The total subcell force that applies onto point p from cell c is constituted of the pressure, artificial viscosity and anti-hourglass forces

$$\boxed{\boldsymbol{F}_{cp} = \boldsymbol{F}_{cp}^{\text{press}} + \boldsymbol{F}_{cp}^{q} + \boldsymbol{F}_{cp}^{\Delta P}.} \tag{1.55}$$

Recall that many other physical models could be added to the system of equations and, thanks to the compatible discretization, this would induce other types of subcell forces to be added to (1.55). By summation of subcells cp around a point p we construct the total nodal force that applies onto p as

$$\boldsymbol{F}_p = \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp}, \tag{1.56}$$

which is further used in (1.25) to compute the evolution of the momentum equation.

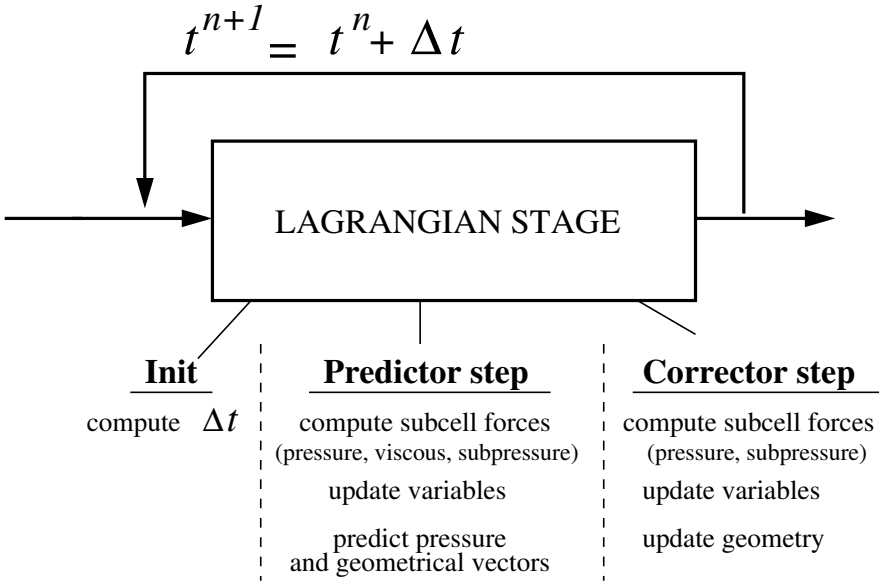


Figure 1.4 – *Simplistic view of the predictor-corrector temporal scheme used in the compatible staggered Lagrangian stage. See the algorithm in the text for an exhaustive description.*

1.2.4 Time discretization

The time discretization is obtained by means of a classical two-step predictor-corrector scheme to gain second-order accuracy as presented in Fig. 1.4 and in the following algorithm.
Being given geometric quantities and physical variables at time t^n, we first predict the time centered geometrical quantities and pressures that are later used in the corrector step to update physical and geometric variables. There exist several other ways to exhibit a predictor-corrector scheme for this system, each having some interesting properties and drawbacks (as instance one can avoid to call the equation of state routine in the predictor phase or one can avoid to update the velocity in the predictor phase). Nevertheless this algorithm is symmetric (apart from steps 0. and 9. which are specific to the predictor step) which simplifies its implementation and the associated code maintenance.

Predictor step.

0. Compute subcell artificial viscous force $\boldsymbol{F}_{cp}^{q,n}$, deduce the time step Δt
1. Compute subcell pressure force $\boldsymbol{F}_{cp}^{\text{press},n} = -L_{cp}^{n} P_c^n \boldsymbol{N}_{cp}^{n}$
2. Compute subcell anti-hourglass force $\boldsymbol{F}_{cp}^{\Delta P,n}$
3. Compute total subcell forces

$$\boldsymbol{F}_{cp}^{n} = \boldsymbol{F}_{cp}^{\text{press},n} + \boldsymbol{F}_{cp}^{q,n} + \boldsymbol{F}_{cp}^{\Delta P,n}$$

4. Update momentum equation

$$\begin{aligned} m_p \left(\boldsymbol{U}_p^{n+1} - \boldsymbol{U}_p^{n} \right) &= -\Delta t \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{cp}^{n} \\ \boldsymbol{U}_p^{n+1/2} &= \frac{1}{2} \left(\boldsymbol{U}_p^{n+1} + \boldsymbol{U}_p^{n} \right) \end{aligned}$$

5. Update internal energy equation

$$m_c \left(\varepsilon_c^{n+1} - \varepsilon_c^{n} \right) = \Delta t \sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{cp}^{n} \cdot \boldsymbol{U}_p^{n+1/2}$$

6. Update vertex position

$$\boldsymbol{X}_p^{n+1} = \boldsymbol{X}_p^{n} + \Delta t \, \boldsymbol{U}_p^{n+1/2}$$

7. Recompute cell/subcell volumes, geometrical entities and densities at t^{n+1}

$$\rho_c^{n+1} = \frac{m_c}{V_c^{n+1}}, \qquad \rho_{cp}^{n+1} = \frac{m_{cp}}{V_{cp}^{n+1}}$$

8. Compute updated pressures

$$\begin{aligned} P_c^{n+1} &= P\left(\rho_c^{n+1}, \varepsilon_c^{n+1} \right) \\ P_{cp}^{n+1} &= P\left(\rho_{cp}^{n+1}, \varepsilon_{cp}^{n+1} \right) \end{aligned}$$

9. Compute time centered geometrical entities and predicted pressures

$$\begin{aligned} P_c^{n+\frac{1}{2}} &= \frac{1}{2} \left(P_c^{n+1} + P_c^{n} \right) \\ P_{cp}^{n+\frac{1}{2}} &= \frac{1}{2} \left(P_{cp}^{n+1} + P_{cp}^{n} \right) \end{aligned}$$

Corrector step.

0. —
1. Compute subcell pressure force $\boldsymbol{F}_{cp}^{\text{press},n+1/2} = -L_{cp}^{n+1/2} P_c^{n+1/2} \boldsymbol{N}_{cp}^{n+1/2}$
2. Compute subcell anti-hourglass force $\boldsymbol{F}_{cp}^{\Delta P,n+1/2}$
3. Compute total subcell forces

$$\boldsymbol{F}_{cp}^{n+1/2} = \boldsymbol{F}_{cp}^{\text{press},n+1/2} + \boldsymbol{F}_{cp}^{q,n} + \boldsymbol{F}_{cp}^{\Delta P,n+1/2}$$

4. Update momentum equation

$$\begin{aligned} m_p \left(\boldsymbol{U}_p^{n+1} - \boldsymbol{U}_p^{n} \right) &= -\Delta t \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{cp}^{n+1/2} \\ \boldsymbol{U}_p^{n+1/2} &= \frac{1}{2} \left(\boldsymbol{U}_p^{n+1} + \boldsymbol{U}_p^{n} \right) \end{aligned}$$

5. Update internal energy equation

$$m_c \left(\varepsilon_c^{n+1} - \varepsilon_c^{n} \right) = \Delta t \sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{cp}^{n+1/2} \cdot \boldsymbol{U}_p^{n+1/2}$$

6. Update vertex position

$$\boldsymbol{X}_p^{n+1} = \boldsymbol{X}_p^{n} + \Delta t \, \boldsymbol{U}_p^{n+1/2}$$

7. Recompute cell/subcell volumes, geometrical entities and densities at t^{n+1}

$$\rho_c^{n+1} = \frac{m_c}{V_c^{n+1}}, \qquad \rho_{cp}^{n+1} = \frac{m_{cp}}{V_{cp}^{n+1}}$$

8. Compute updated pressures

$$\begin{aligned} P_c^{n+1} &= P\left(\rho_c^{n+1}, \varepsilon_c^{n+1} \right) \\ P_{cp}^{n+1} &= P\left(\rho_{cp}^{n+1}, \varepsilon_{cp}^{n+1} \right) \end{aligned}$$

End of time step.
Final data $\rho_c^{n+1}, \rho_{cp}^{n+1}$, ε_c^{n+1} and P_c^{n+1}, then $\boldsymbol{U}_p^{n+1}$, $\boldsymbol{X}_p^{n+1}$ and mesh related entities (volumes, lengths, corner vectors, etc.).

In the following we focus on important details of several steps of the algorithm.

0. Compute subcell artificial viscous force $\boldsymbol{F}_{cp}^{q,n}$ and deduce the time step Δt.
This step is only performed for the prediction. The time step is constrained by the classical CFL condition which roughly states that any information can not travel across more than one cell during the time step. In other words

$$\Delta t \leq \min_c \left(\frac{L_c^{\text{characteristic}}}{S_c} \right), \tag{1.57}$$

where $L_c^{\text{characteristic}}$ is a cell-based characteristic length. We take $L_c^{\text{characteristic}} = \min_{e \in \mathcal{E}(c)} L_e$ where $\mathcal{E}(c)$ is the set of edges of cell c. Moreover S_c is the characteristic sound-speed of cell c which is constituted of a_c the actual sound-speed in cell c and a "viscous" sound speed obtained from the artificial viscous model a_c^{viscous}, therefore $S_c = \sqrt{a_c^2 + (a_c^{\text{viscous}})^2}$. The occurence of a "viscous" sound speed is mandatory to allow a valid definition of a time step when the initial specific internal energy is zero or close to zero, which in the case of a perfect gas leads to a sound-speed a_c close to zero, hence a time step tending to infinity. In addition to this we classicaly multiply the time step with a security coefficient of $1/4$ hence the time step is given by

$$\Delta t \leq \frac{1}{4} \min_c \left(\frac{\min_{e \in \mathcal{E}(c)} L_e}{\sqrt{a_c^2 + (a_c^{\text{viscous}})^2}} \right). \tag{1.58}$$

We notice that the viscous subcell forces are only computed at time t^n and not updated for the corrector step. Up to our knowledge there is no rigorous reason to justify this point. The fact that updating the viscous force does not drastically modify the results and the relative cost of such step seem two acceptable points in favour of such legacy.
Finally in order to avoid too violent cell volume change during one single time step, Δt is not allowed to increase by more than 20% percents compared to its previous value.

1- Compute subcell pressure force $\boldsymbol{F}_{cp}^{\text{press},*} = -L_{cp}^* P_c^* \boldsymbol{N}_{cp}^*$.
Note that the time centering of the geometrical entities $-L_{cp}^* \boldsymbol{N}_{cp}^*$ and the pressure P_c^* are the same. This is the reason why at the end of the predictor step all geometrical entities must be recomputed to match the time centering of the predicted pressure (at time $t^{n+1/2}$ in our algorithm).

5- Update internal energy equation

$$m_c \left(\varepsilon_c^{n+1} - \varepsilon_c^n \right) = \Delta t \sum_{p \in \mathcal{P}(c)} \boldsymbol{F}_{cp}^* \cdot \boldsymbol{U}_p^{n+1/2}. \tag{1.59}$$

Using $\boldsymbol{U}_p^{n+1/2} = \frac{1}{2} \left(\boldsymbol{U}_p^{n+1} + \boldsymbol{U}_p^n \right)$ in the equation above is mandatory to fulfill the total energy conservation. To prove it let us start from the momentum equation

$$m_p \left(\boldsymbol{U}_p^{n+1} - \boldsymbol{U}_p^n \right) = -\Delta t \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{cp}^*. \tag{1.60}$$

Multiplying (1.60) by $\frac{1}{2} \left(\boldsymbol{U}_p^{n+1} + \boldsymbol{U}_p^n \right)$ yields the evolution of the kinetic energy equation for point p

$$\begin{aligned} \frac{1}{2} m_p \left(\boldsymbol{U}_p^{n+1} - \boldsymbol{U}_p^n \right) \left(\boldsymbol{U}_p^{n+1} + \boldsymbol{U}_p^n \right) &= -\Delta t \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{cp}^* \cdot \frac{1}{2} \left(\boldsymbol{U}_p^{n+1} + \boldsymbol{U}_p^n \right), \\ \frac{1}{2} m_p \left(\boldsymbol{U}_p^{n+1} \right)^2 &= \frac{1}{2} m_p \left(\boldsymbol{U}_p^n \right)^2 - \Delta t \sum_{c \in \mathcal{C}(p)} \boldsymbol{F}_{cp}^* \cdot \boldsymbol{U}_p^{n+1/2}. \end{aligned}$$

By suming over all points p (and neglecting boundary conditions or assuming periodic boundary conditions) one gets

$$\frac{1}{2}\sum_p m_p \left(\boldsymbol{U}_p^{n+1}\right)^2 = \frac{1}{2}\sum_p m_p \left(\boldsymbol{U}_p^{n}\right)^2 - \Delta t \sum_p \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp}^{*}\cdot\boldsymbol{U}_p^{n+1/2}. \tag{1.61}$$

On the other hand rewritting (1.59) under the form

$$m_c\varepsilon_c^{n+1} = m_c\varepsilon_c^{n} + \Delta t \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}^{*}\cdot\boldsymbol{U}_p^{n+1/2},$$

and suming over all cells c yields

$$\sum_c m_c\varepsilon_c^{n+1} = \sum_c m_c\varepsilon_c^{n} + \Delta t \sum_c \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}^{*}\cdot\boldsymbol{U}_p^{n+1/2}. \tag{1.62}$$

At last adding (1.61) and (1.62) gives

$$\begin{aligned}\sum_c m_c\varepsilon_c^{n+1} + \frac{1}{2}\sum_p m_p \left(\boldsymbol{U}_p^{n+1}\right)^2 &= \sum_c m_c\varepsilon_c^{n} + \frac{1}{2}\sum_p m_p \left(\boldsymbol{U}_p^{n}\right)^2 \\ &\quad -\Delta t\left(\sum_p\sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp}^{*}\cdot\boldsymbol{U}_p^{n+1/2} - \sum_c\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}^{*}\cdot\boldsymbol{U}_p^{n+1/2}\right)\end{aligned}$$

switching the sum signs in one of the two terms in parenthesis (i.e. $\sum_p\sum_{c\in\mathcal{C}(p)} \equiv \sum_c\sum_{p\in\mathcal{P}(c)}$) must convince the reader that the term in parenthesis is indeed equal to zero. Therefore the previous equation reduces to

$$\sum_c m_c\varepsilon_c^{n+1} + \frac{1}{2}\sum_p m_p \left(\boldsymbol{U}_p^{n+1}\right)^2 = \sum_c m_c\varepsilon_c^{n} + \frac{1}{2}\sum_p m_p \left(\boldsymbol{U}_p^{n}\right)^2,$$

which clearly states that if total energy is defined as $E = \sum_c m_c\varepsilon_c + \frac{1}{2}\sum_p m_p \left(\boldsymbol{U}_p\right)^2$, then this quantity is conserved only if the subcell force is dot-producted with $\boldsymbol{U}_p^{n+1/2}$ in equation (1.59).

8- Compute updated pressures

$$P_c^{n+1} = P\left(\rho_c^{n+1}, \varepsilon_c^{n+1}\right), \qquad P_{cp}^{n+1} = P\left(\rho_{cp}^{n+1}, \varepsilon_{cp}^{n+1}\right).$$

The subcell pressures could instead be computed in steps 2. if the anti-hourglass forces $\boldsymbol{F}_{cp}^{\Delta P,n}$ are to be evaluated. (The subcell densities can be also computed during this step as both subcell densities and subcell pressures are only utilized to compute anti-hourglass forces). Consequently, depending on implementation or efficiency reasons the computation of subcell entities can be removed from steps 7. and 8. and moved to steps 2. of the algorithm.

9- Compute time centered geometrical entities and predicted pressures

$$P_c^{n+\frac{1}{2}} = \frac{1}{2}\left(P_c^{n+1} + P_c^{n}\right), P_{cp}^{n+\frac{1}{2}} = \frac{1}{2}\left(P_{cp}^{n+1} + P_{cp}^{n}\right).$$

The goal of the prediction step is to center all entities at time $t^{n+1/2}$ which are later used to compute forces and advance point position (and by association geometrical entities like cell volume and cell density), point velocity and cell centered specific internal energy. These pressures contribute to the subcell pressure forces computed in step 1. of the corrector stage.

1.2.5 Boundary conditions

Boundary conditions in a Lagrangian formulation are of two different types : prescribed normal velocity or pressure.
If a prescribed normal velocity is enforced along two edges impinging at a boundary point p, say $v_p^* = \boldsymbol{U}_p \cdot \boldsymbol{N}_p$ where $\boldsymbol{N}_p$ is a unit outward normal at point p, then it is usually considered as sufficient to add a velocity correction between the two equations of step 4. in the algorithm of section 1.2.4 both for predictor and corrector steps. This sub-step consists of modifying the point velocity $\boldsymbol{U}_p^{n+1}$ after its evaluation in such a way that its normal component be equal to v_p^* and its tangential component remain unchanged. Indeed if we call $\boldsymbol{T}_p$ the unit tangential vector at point p, this amounts to find the components of velocity $\widehat{\boldsymbol{U}}_p$ such that :

$$\widehat{\boldsymbol{U}}_p \cdot \boldsymbol{N}_p = v_p^*, \tag{1.63}$$

$$\widehat{\boldsymbol{U}}_p \cdot \boldsymbol{T}_p = \boldsymbol{U}_p^{n+1} \cdot \boldsymbol{T}_p, \tag{1.64}$$

knowing v_p^*, $\boldsymbol{U}_p^{n+1}$, $\boldsymbol{N}_p$ and $\boldsymbol{T}_p$. The boundary condition friendly velocity $\widehat{\boldsymbol{U}}_p$ is given by

$$\widehat{\boldsymbol{U}}_p = v_p^* \boldsymbol{N}_p - (\boldsymbol{U}_p^{n+1} \cdot \boldsymbol{T}_p)\boldsymbol{T}_p. \tag{1.65}$$

For pressure boundary condition P^* we usually define ghost subcells around boundary points for which we set the pressure to be P^*. This modification is to be operated before the update of momentum equation, say between step 3. and 4. This amounts to modify (1.35) for any subcell having an edge on the boundary line as if a subcell from a ghost cell c' is present

$$\boldsymbol{F}_{cp}^{\text{press}} = -L_{cp}\, P_c\, \boldsymbol{N}_{cp} - L_{c'p}\, P^*\, \boldsymbol{N}_{c'p}. \tag{1.66}$$

By construction of the ghost subcell we have $L_{c'p}\boldsymbol{N}_{c'p} = -L_{cp}\boldsymbol{N}_{cp}$, hence

$$\boldsymbol{F}_{cp}^{\text{press}} = -L_{cp}\, (P_c - P^*)\, \boldsymbol{N}_{cp}. \tag{1.67}$$

Obviously v^* and P^* may be space/time dependent boundary conditions so that accelerated piston, spacial varying pressure boundary conditions as well as no-slip boundary conditions and constant pressure can be easily applied.

Note that most of the previous proofs (conservation, Galilean invariance, etc) where periodic boundary conditions were assumed can be revamped using more complex boundary conditions, the principles of the proof being the same, only the equations are more involved, see [42] as instance.

1.2.6 Cylindrical $r-z$ geometry

This compatible staggered Lagrangian scheme has also been extended to 2D $r-z$ cylindrical geometry either using a so-called *control volume* (CV) discretization, which does not maintain cylindrical symetry, or a so-called *area-weighted* (AW) discretization which can [55]. Note that our section is freely inspired from Section 3 of paper [55], and, undoubtedly, the reader must compulse the aforementioned paper.

Control volume formulation

The description of control volume (CV) formulation can be found in [55] but its formulation surely dates back to the 50's. Let us provide a brief description of this formulation. First we call r_p, z_p the coordinates of a point $\boldsymbol{X}_p$. The formulation starts with a true cylindrical cell volume definition

$$V_c = \int_{\Omega_c} r \, dr \, dz. \tag{1.68}$$

This volume integral is indeed a function of point positions. Any geometrical entities such as length, normals, surfaces can be also derived in cylindrical geometry. (A more advanced description of all algebraic manipulations involved in this formulation can also be found in [97].) Then using (1.13), a discrete divergence operator can be defined. Accordingly the associated discrete gradient can be derived. This essentially determines the form of the geometrical corner vector $L_{cp}\boldsymbol{N}_{cp}$. Subcell mass is also computed taking into account the cylindrical volumes which following (1.10) determines the cell and point masses. Nodal force can then be constructed and finally the energy equation is updated in a compatible way. As we can observe the same 'volume' definition is used both to compute density and cell volume but also the work within the energy equation. Consequently total energy and momentum are conserved.
The problem of CV formulation is that it will not preserve two-dimensional cylindrical $r-z$ symetry of a spherical flow. As quoted by Caramana in [55] and further observed in simple numerical examples, *one can specify one-dimensional, spherically symmetric initial and boundary conditions and the numerical solution, computed with the control volume scheme described above, will not remain spherical in time*. Spherical symmetry CV formulation is violated because the areas along the angular direction are not equal even when the angles between the radial lines are equal. Therefore for symmetric pressure distribution along radius the calculated force can not be radial, leading to symmetry violation. However for an equal angular mesh, 2D cylindrical symmetry is preserved as in Cartesian geometry because the lengths along the angular direction are equal. Consequently the computed nodal force perpendicular to the radial direction vanishes for a spherically symmetric distribution of pressures [63].

Area-weighted formulation

The description of area-weighted (AW) formulation can be found in [55] but it dates back to the time of the Green Book [69, 98, 82] and revamped in [99, 100]. This method is an example of discretization of axisymetrical equations which preserves spherical symmetry of the numerical solution on equal angular mesh. As a consequence AW is often the prefered Lagrangian discretization for problem with cylindrical geometry. For this type of scheme one begins by postulating the form of the gradient operator, based on physical reasoning of what is necessary for symmetry preservation for an equal angular mesh. This implicitly determines the zone volume definition. Strictly speaking such schemes violate momentum conservation ; in compatible form they may give rise to entropy errors. As seen previously for an equal angular mesh, 2D cylindrical symmetry is preserved in Cartesian geometry. This property is used to construct the area-weighted schemes in cylindrical geometry focusing on preserving this spherical symmetry. To obtain the area-weighted schemes one simply multiplies the vector lengths, as deïňĄned in Cartesian geometry, of the entire force contour defined with respect to a given point, p, by the value of the radius r_p. Then the Lagrangian nodal mass m_p is also defined at point p as an effective local inertia $(\rho A)_p$ times r_p

$$m_p = r_p(\rho A)_p, \tag{1.69}$$

so that the momentum equation (1.16) in cylindrical geometry becomes

$$r_p(\rho A)_p \frac{d}{dt}\boldsymbol{U}_p = -\sum_{c\in\mathcal{C}(p)} r_p \boldsymbol{F}_{cp},$$

where $\boldsymbol{F}_{cp}$ is the subcell Cartesian force. Then r_p cancels in the previous equation yielding

$$(\rho A)_p \frac{d}{dt}\boldsymbol{U}_p + \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp} = \boldsymbol{0}, \tag{1.70}$$

which is essentially the same as (1.16) for Cartesian geometry. The area inertia is further defined using the true cylindrical initial mass m_p using definition (1.69). Caramana provides in [55] a simple way to compute nodal, cell and subcell masses and volumes in cylindrical geometry : in short the cell volume is split into $\mathcal{P}(c)$ Cartesian triangular volumes [4] denoted $A_{pp'}$ multiplied by the "radius" $\frac{1}{3}(r_p + r_{p'} + r_c)$, so that the AW cylindrical cell volume is

$$V_c = \sum_{p\in\mathcal{P}(c)} A_{pp'} \frac{1}{3}(r_p + r_{p'} + r_c). \tag{1.71}$$

Finally the specific internal energy evolution is computed by the work obtained by multiplying each subcell force by the nodal radius times the nodal velocity

$$m_c \frac{d}{dt}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} r_p \boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p = 0, \tag{1.72}$$

where once again $\boldsymbol{F}_{cp}$ is the subcell Cartesian force. Due to the presence of r_p in front of $\boldsymbol{F}_{cp}$ in (1.72) one deduces that momentum can not be conserved [5]
Unfortunately keeping in mind the future use of this Lagrangian scheme within a conservative ALE code the AW formulation can not be considered. Because the conservative remapping step relies on integral definition of volume and mass, but, as already seen, the AW cell volume (1.71) can not be reinterpreted into a proper integral formula contrarily to CV formulation. This enforces us to abandon the AW formulation in favour of the CV formulation for a conservative ALE code.

1.2.7 Discussion

The central feature of this more modern form of Lagrangian hydrodynamics is its ability to exactly conserve mass, momentum, and total energy without the need to use these quantities directly as variables. It instead retains density, velocity, and specific internal energy as dependent variables as did the earlier version of this algorithm [65]. Total energy conservation is ensured by the use of a "compatible" discretization while the conservation of momentum is obtained by assuring that subcell forces, no matter which physical process they may represent, do sum to zero within a cell. Finally mass conservation is trivially fulfilled due to the Lagrangian formalism.
Even if this is not presented in this short description, many different physical models can and have been coupled to this staggered Lagrangian scheme : elasto-plasticity, radiative transport, diffusion equation, multi-material treatment, etc. Consequently this Lagrangian scheme has been used and still is in many Lagrangian or ALE simulation codes. Nevertheless analysis and understanding of its intricate nature is still an on-going work even if the scheme is ancient.
Some of such investigations are presented in the next section.

4. Such triangle is the zone defined with two adjacent points p and p' and the cell center c.
5. Indeed because of factor r_p, (1.28) turns into $\sum_{p\in\mathcal{P}(c)} r_p \boldsymbol{F}_{cp} \neq \boldsymbol{0}$, in general.

1.3 Numerical analysis

This section presents some results on the numerical analysis of the compatible staggered Lagrangian scheme. More specifically the internal consistency, accuracy and stability issues are addressed in the first subsection as paper [12] is presented. Then the proof of a result from [12] which has been published in [15] is summarized in the second subsection. Finally in the third subsection the volume consistency of the scheme is investigated by briefly reviewing the results of paper [16].

1.3.1 Internal consistency, accuracy and stability

Led by E.J. Caramana, a joined effort with several colleagues from LANL gave rise in 2006 to paper [12] the title of which is *"The internal consistency, accuracy and stability of the discrete compatible Formulation of Lagrangian Hydrodynamics"*. The goals of this work were

To study the internal consistency of the scheme by analyzing the difference between the two definitions of cell volume the scheme utilizes : A compatible cell volume V_c^{comp} deduced from the discrete version of the divergence of the velocity (1.11) and a cell volume obtained from point coordinates $V_c^{coord} = f\left(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{|\mathcal{P}(c)|}\right)$ [6] The derivatives of the cell volume with respect to its coordinate dependence is used to define the geometrical vectors (1.12) associated with point p, as was done in the work of Favorskii [71], and Margolin and Adams [72]. Thus the geometrical vectors used to construct the subcell forces are not arbitrarily specified, as with the older versions of this type of hydrodynamics [69], but are a consequence of the chosen volume definitions. The analysis in the article shows that if the geometrical vectors are time-centered and for zero force then the two volumes in 2D Cartesian geometry are equal, *a posteriori* justifying the $t^{n+1/2}$ time centering at the end of the predictor phase (step 9. of the predictor stage of the algorithm in section 1.2.4). When the forces are not zero then it is shown that the difference in coordinate and compatible volumes is of order Δt^3 on a single timestep and the time accuracy (globally integrated up to a final time t^n) is of the order Δt^2. It was shown that this difference can be used to ascertain many properties of a simulation, and as such has direct and practical significance.

To construct non-dimensional internal consistency norms based on the difference in these two aforementioned volumes. These can be used to operationally measure the non-dimensional magnitude of the truncation error of a calculation by placing the geometrical vectors from which the subcell force is calculated at the fully advanced time level on the corrector step, that is to say t^{n+1} instead of $t^{n+1/2}$.

To validate the error indicators on a set of numerical tests (Guderley, Noh, Sedov in 2D cylindrical geometry and also in 3D). They also serve to illustrate how they can be utilized to assess the quality of numerical simulations. We have demonstrated that the size of the error associated with the coordinate and compatible volumes is significant only when severe numerical difficulties, such as numerical instability, arise. The accuracy in both space and time were also measured, and results were found to correspond to the first or second order accuracy that one expects in space or time with the type of finite-volume differencing employed.

To analyse the numerical stability of the two level predictor-corrector time integration scheme employed : First with a simplified set of model equations using standard Fourier stability analysis, and then using a properly constructed test problem that verifies this analysis for the

6. In the article a cell is refered to as a "zone" z, a subcell to a "corner", and these two volumes are denoted V_z^{crd} and V_z^{cmp}.

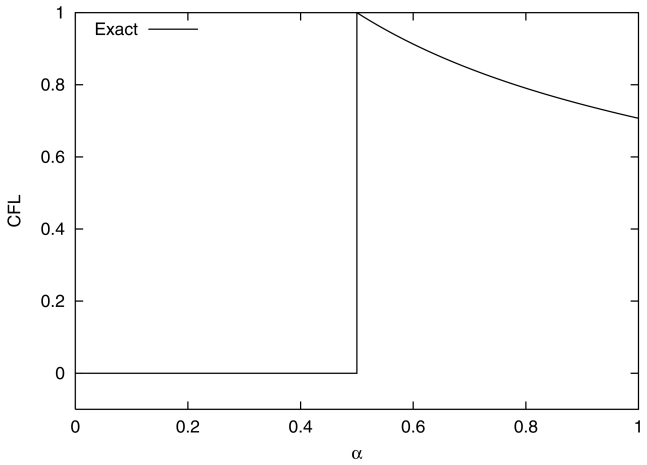


Figure 1.5 – *Numerical results from paper [12]. Maximum CFL number as a function of parameter α.*

actual system of equations in all three spatial dimensions. In this stability analysis we studied the effect of the time centering parameter α of the pressure on the corrector step (step 9. of the predictor step of the algorithm in section 1.2.4) when written as :

$$P_c^{n+\frac{1}{2}} = \alpha P_c^{n+1} + (1-\alpha) P_c^n.$$

We numerically showed that our compatible system of equations with predictor-corrector time integration is stable for CFL number $CFL \leq 1/\sqrt{2\alpha}$ with $\alpha \geq 1/2$; otherwise, it is unconditionally unstable (see the Fig. 1.5). This somewhat justifies the optimal choice of $\alpha = 1/2$ which maximizes the usable CFL coefficient.

This paper is reproduced in the following pages.

Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 218 (2006) 572–593
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics

A.L. Bauer [a,b,*], D.E. Burton [c], E.J. Caramana [d], R. Loubère [e], M.J. Shashkov [b], P.P. Whalen [f]

[a] Department of Mathematics, University of Michigan, 2074 East Hall, 530 Church Street, Ann Arbor, MI 48109-1043, USA
[b] T-7, MS B284, Los Alamos National Laboratory, Los Alamos, NM, USA
[c] X-3, MS T086, Los Alamos National Laboratory, Los Alamos, NM, USA
[d] CCS-2, MS D413, Los Alamos National Laboratory, Los Alamos, NM, USA
[e] MIP, University of Toulouse, 31062 Toulouse Cedex 9, France
[f] X-7, MS F644, Los Alamos National Laboratory, Los Alamos, NM, USA

Received 30 November 2005; received in revised form 21 February 2006; accepted 23 February 2006
Available online 19 April 2006

Abstract

This work explores the somewhat subtle meaning and consequences of the salient properties of the discrete, compatible formulation of Lagrangian hydrodynamics. In particular, since this formulation preserves total energy to roundoff error, the amount of error in the conservation of total energy cannot be used to gauge the internal consistency of calculations, as is often done with the older forms of this algorithm. However, the compatible formulation utilizes two definitions of zone volume: the first is the usual definition whereby the volume of a zone is defined as some prescribed function of the coordinates of the points that define it; the second is given as the integration in time of the continuity equation for zone volume as expressed in Lagrangian form. It is the use of this latter volume in the specific internal energy equation that enables total energy to be exactly conserved. These two volume definitions are generally not precisely equal. It is the analysis of this difference that forms the first part of this study. It is shown that this difference in zone volumes can be used to construct a practical internal consistency measure that not only takes the place of the lack of total energy conservation of the older forms of Lagrangian hydrodynamics, but is more general in that it can be defined on a single zone basis. It can also be used to ascertain the underlying spatial and temporal order of accuracy of any given set of calculations. The difference in these two definitions of zone volume may be interpreted as a type of entropy error. However, this entropy error is found to be significant only when a given calculation becomes numerically unstable, otherwise it remains at or far beneath truncation error levels. In fact, it can be utilized to provide an upper bound on the size of the spatial truncation error for a stable computation. It is also shown how this volume difference can be used as an indicator of numerical difficulties, since exact local conservation of total energy does not guarantee numerical stability or the quality of any numerical calculation. The discrete, compatible formulation of Lagrangian hydrodynamics utilizes a two level predictor/corrector-type of time integration scheme; a stability analysis, both analytical and numerical, is given. This analysis reveals a novel stability diagram that has not been heretofore published, and gives definitive information as to how the stabilizing corrector step should be centered in time.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Lagrangian scheme; Stability; Compressible fluids; Compatible discretization; Support operator

* Corresponding author. Tel.: +1 734 936 4557/997 0509; fax: +1 734 763 0937.
E-mail addresses: albauer@umich.edu (A.L. Bauer), burton@lanl.gov (D.E. Burton), ed.caramana@gmail.com (E.J. Caramana), loubere@mip.ups-tlse.fr (R. Loubère), shashkov@lanl.gov (M.J. Shashkov), ppw@lanl.gov (P.P. Whalen).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.02.024

1. Introduction

The Lagrangian formulation of the equations of hydrodynamics has a very old and venerable history. Indeed, the very first large-scale numerical calculations that resemble modern computer simulations in both complexity and in the numerical issues considered utilized fluid equations in the Lagrangian frame of reference in one-dimension [1]. If one considers an arbitrary fluid velocity v and distinct sound speed c_s, then in one-dimension the characteristic trajectories of the fluid equations have associated eigenvalues $v \pm c_s$ and v. In the Lagrangian frame that follows the fluid velocity v, these eigenvalues transform into the Galilean invariant values $\Delta v \pm c_s$ and Δv, where Δv is the difference in velocity between two adjacent discrete spatial locations in the flow field. Thus, the Lagrangian frame of reference is unique, and one characteristic is chosen and followed exactly, except for spatial gradients in the velocity field. The fact that the Lagrangian description of fluid dynamics is automatically adaptive makes it the preferred representation in one-dimension. Newton's second law of motion, $\vec{F} = M\vec{a}$, where M is the mass of a point particle (or fluid element), $\vec{F}$ is the total force acting on it, with the acceleration $\vec{a}$ given as the second derivative with respect to time of its displacement vector $\vec{r}$, is a statement set in the Lagrangian frame of reference. Discretizing this equation directly with respect to time utilizing $\vec{r}$ as a dependent variable results quite naturally in the three-level leapfrog scheme. These time levels are usually denoted as $n-1$, n, and $n+1$, with the force $\vec{F}$ spatially differenced in some manner but placed at time level n. All early Lagrangian algorithms in both 1D [2] and in 2D [3] utilized this kind of discretization with respect to time. Although this forms a simple and intuitive numerical integration scheme, it leaves the velocity of a mass point or fluid element defined only as the difference between its displacement vector at two different time levels, and therefore the velocity is defined only at the $n \pm \frac{1}{2}$ time levels. When one then considers the total energy of a fluid as a sum of both kinetic energy and internal energy that can be exchanged between each other by the action of forces, this sum is difficult to conserve exactly in discrete form owing to the fact that the two components that comprise it are defined at different time levels. When velocity dependent forces are explicitly added to this model, as with the artificial viscosity [4], this type of time integration becomes somewhat clumsy and looks even contrived [2], since the artificial viscosity terms must be lagged in time to preserve numerical stability.

The spatial discretization of the force in all early versions of Lagrangian hydrodynamics [2,3] is some form of what is presently known as finite-volume differencing. That is, these various forms calculate force as a stress (scalar pressure plus deviators) times a normal surface-area vector. The most modern of these older force calculations is the diamond differencing scheme due to Wilkins [3], which uses closed surface area contours to calculate the force acting on a point, and thus properly conserves linear momentum in a trivial manner. Others piece the force contributions together in various ways to form the total force acting on each fluid element such that strict conservation of linear momentum may, or may not, be obtained. Most Lagrangian hydrodynamics codes employ a spatially staggered placement of dependent variables with stress, density, and specific internal energy given in zones surrounded by points that have associated position and velocity vectors. This enables the calculation of forces by means of the various kinds of finite-volume differencing referenced above, with masses and volumes ascribed to both zones and points in an interleaved manner. This also avoids the "grid-decoupling" instability that is the bane of non-staggered forms of this algorithm. A difficulty with the older work is that there was no agreement amongst the various authors of these different algorithms as to how these schemes, aside from the noted common features, should be constructed. The choices made were largely arbitrary and not tied to solid mathematical concepts.

An early attempt to remedy this lack of a sound theoretical basis is the work of Goad [5], who used the method of virtual work to derive a form of finite-volume force differencing of the stress in 2D cylindrical

geometry. This work was little noted, partly because this type of scheme does not yield the limit of 1D spherical geometry from 2D cylindrical geometry that is important in some applications [6]. Next, finite element spatial discretizations were utilized. However, these must be of low order accuracy to prevent the occurrence of non-physical oscillations about shock discontinuities that appear if the support of the basis functions is too large. Thus, finite element spatial representations only give the appearance of a rigorous foundation while sacrificing the flexibility that a finite-volume discretization affords.

The work that first places this type of finite-volume algorithm on a firm theoretical basis is that of Favorskii [7], and independently, Margolin and Adams [8]. The first paper shows that the discrete equations in Lagrangian form can be generally derived from a variational principle. It also justifies the use of the surface area vectors of closed volumes as proper discretization objects, a practice that was previously employed, but not always correctly, in that the surface areas about a point did not in all cases sum to zero. The second paper parallels this work. Its central thrust is to use the continuity equation in discrete form to derive finite-volume differencing given a discrete expression for the volume of a zone. This again results in, and justifies, the surface areas utilized to calculate the force employed. It emphasizes that the difference formulas that are derived are "operator" expressions that can be used to calculate discrete derivatives of any function, and not just of the velocity field that they originally act upon. It is the discrete form of the continuity equation as emphasized in [8] that is central to our discussion of internal consistency. A further extension of the work of Favorskii results in what has become known as the "method of support operators" [9]. Although this work is more general than just its application to the equations of Lagrangian hydrodynamics, it is this system of equations that is used in its original exposition. This method also utilizes the continuity equation in discrete form to derive the divergence "operator" and then uses the vector identities in summation form to derive discrete versions of all other operators. It emphasizes the relation in discrete form of the divergence and gradient operators as negative adjoints of each other as in the continuum case. The paper of Margolin and Tarwater [10] again parallels this work. It derives the gradient operator from the divergence operator by using the same requirement – that they must be negative adjoints. This is done in the context of deriving a compatible expression for the diffusion operator as the direct product of gradient and divergence, and requires additionally that this diffusion operator be exact for a linear function. These publications all revolve around the central idea that the discrete equations must obey the global properties of the continuum ones in order to be considered as valid discretizations that will then mirror continuum conservation properties in their discrete analogs. They thus remove the arbitrary and heuristic formulations of the previous, but well-used, codes based on the older work [3].

Somewhat after the previously cited developments is the seminal work of Burton [11,12], which discretizes the fluid equations in Lagrangian form on a staggered spatial grid utilizing subgrid quantities termed corner masses and corner forces, from which the zone and point masses, and the total force acting on a point, are constructed. A two-level time integration scheme is also utilized so that both kinetic energy and internal energy are defined at the same time level. The basic reasoning used by Burton to demonstrate conservation of total energy is the same as that employed in the method of support operators [9], and thus incorporates the important features of the previous works [8–10]. However, Burton's formulation is more general in that he does not consider forces, or differential operators, of any specific origin. Instead, he utilizes an arbitrary corner force that allows the specification of forces of any functional form, with the work that they perform completely defined and total energy exactly conserved. The only restriction on the discrete form of the corner force is the relatively mild physical constraint of momentum conservation. He also notes [12] that this formulation of the Lagrangian hydrodynamics equations contains two distinct definitions of zone volume, and considers this difference to be a form of "entropy" error.

It is this latter work of Burton that we refer to herein as the "discrete, compatible formulation of Lagrangian hydrodynamics", and which was initially constructed on arbitrary polyhedral grids [12]. The word "discrete" has been inserted to emphasize that these equations are essentially created in discrete form, as opposed to being the discretization of a system of PDE's. As such, one may or may not be able to rigorously take the continuum limit to obtain the latter; this depends on the kinds of forces that are employed to resolve shocks and to counteract spurious grid motions. It is the subject of this paper to explore important consequences of this formulation. To this end a very brief statement of it is given that includes the definition of the two zone volumes that are central to investigating its internal consistency. A study of this novel internal

consistency feature is then developed theoretically and illustrated with numerical examples. A stability analysis of the time integration scheme utilized follows; last, our principal conclusions are briefly summarized.

2. Discrete, compatible Lagrangian hydrodynamics

Since the numerical algorithm that we term the discrete, compatible formulation of Lagrangian hydrodynamics has been given elsewhere [12,13], we repeat only that skeleton portion that is necessary to explore the features of it that are the subject of this work. To this end, consider the momentum equation given in discrete form as

$$M_p(\vec{v}_p^{\,n+1} - \vec{v}_p^{\,n}) = \sum_z \vec{f}_z^{\,p,*}\Delta t \equiv \vec{F}_p^{\,*}\Delta t. \tag{2.1}$$

To define the terms that occur in this equation we make reference to Fig. 1. This shows a small section of a quadrilateral grid in 2D Cartesian geometry with a single zone labelled by the integer index z, and with its four defining points labelled by the integer index p; these indices range over all zones and points in the computational domain, respectively. Unique to each zone and each point, considered together as a pair-object, is a corner mass m_z^p and a corner force $\vec{f}_z^{\,p}$. The indices of these objects are placed so that the lower one is always summed with the upper one fixed; the asterisk used as a superscript on the force in Eq. (2.1) denotes time-centering, an issue discussed further on. The point mass M_p is formed by summing all m_z^p's that have the same index p, where a particular m_z^p corresponds to the grey region in Fig. 1; likewise, the zone mass M_z that enters Eq. (2.2), is defined as the sum of all m_p^z that have the same zone index z. On this staggered spatial grid the points p carry a velocity $\vec{v}_p$ that, when multiplied by the point mass M_p, represents an average momentum over the median mesh volume, depicted in Fig. 1 as the closed, dashed curve. The position vector $\vec{r}_p$ of the point p is advanced from time level n to time level $n+1$ in a timestep Δt as, $\Delta\vec{r}_p \equiv (\vec{r}_p^{\,n+1} - \vec{r}_p^{\,n}) = (\vec{v}_p^{\,n+1} + \vec{v}_p^{\,n})\Delta t/2$, a form that is unique [14]. As is seen in Eq. (2.1) the total force $\vec{F}_p^{\,*}$ that acts on mass M_p is made up of the sum of all corner forces $\vec{f}_z^{\,p,*}$ with common point index p. The corner force $\vec{f}_z^{\,p,*}$ mediates the exchange of kinetic energy and internal energy between a zone z and a point p; its construction is at this stage completely general.

The most important result of this hydrodynamics formulation, and that which is termed "compatible", is that if at any time t^n a definition of total energy is given as, $E_T(t^n) \equiv \frac{1}{2}\sum_p M_p(\vec{v}_p^{\,n})^2 + \sum_z M_z e_z^n$, then in conjunction with the momentum equation, Eq. (2.1), an equation in discrete form for the evolution of the specific internal energy e_z, defined in the zones z, results and is given as

$$M_z(e_z^{n+1} - e_z^n) \equiv -\sum_p \vec{f}_p^{\,z,*} \cdot \Delta\vec{r}_p. \tag{2.2}$$

In Eq. (2.2) the corner force $\vec{f}_p^{\,z,*}$ is the same as that used in the momentum equation above except that it is now summed with respect to fixed zone index z; all other quantities in Eq. (2.2) have been defined. As noted in

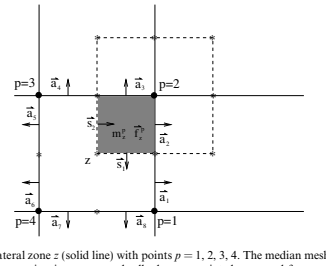


Fig. 1. Notation for a given quadrilateral zone z (solid line) with points $p = 1, 2, 3, 4$. The median mesh is drawn with a dashed line. $\vec{S}_1, \vec{S}_2$ are the median mesh vectors. The gray region is a corner subcell whose associated mass and force are m_z^p and $\vec{f}_z^{\,p}$, respectively.

previous work [13,14], this formulation of Lagrangian hydrodynamics consists of three discrete equations: momentum, specific internal energy, and total energy, such that the specification of any two results in the third. These three equations form an algebraic identity. Also, as discussed previously [13,14], for these equations to make physical sense the arbitrary corner force $\vec{f}_p^{z,*}$ must sum to zero in any zone z; this represents Galilean invariance of the discrete equations.

The primary quantity that must be computed with this form of hydrodynamics to advance the dependent variables in time is the corner force. To advance a timestep Δt this force is first calculated at the n time level where all quantities are known; the dependent variables are then advanced to the $n+1$ time level. However, to obtain numerical stability, as discussed in Section 4, this cycle is repeated with the corner forces recalculated at some time level between half-forward or time centered ($* = n + \frac{1}{2}$), and full-forward ($* = n + 1$). The exact functional form of the corner forces can be quite complicated, and is not at all unique. This has been discussed in depth elsewhere [6,12,14–17]. Here we are concerned with Eq. (2.2) for pressure forces, and how it represents a discrete form of what is generally known as $P\,\mathrm{d}V$ work; that is, how does this equation calculate the work done by a zone pressure P_z^* that is constant throughout the given zone z? The work performed by the "generic" RHS of Eq. (2.2) can be written for pressure forces as:

$$-\sum_p \vec{f}_p^{z,*} \cdot \Delta\vec{r}_p \equiv -P_z^*(\Delta V)_z = -P_z^* V_z(\nabla \cdot \vec{v})_z \Delta t, \tag{2.3}$$

where $(\Delta V)_z$ is the change in volume of zone z in timestep Δt. That this second term is equal to the third term in the above equation is a consequence of the continuity equation for volume in Lagrangian form. More explicitly, if we consider the equation of continuity for the zone density (where zone density ρ_z is defined as $\rho_z \equiv M_z/V_z(t)$) as $\partial\rho/\partial t + \nabla \cdot \rho\vec{v} = 0$, then using the definition given for ρ_z one obtains

$$\frac{\mathrm{d}V_z}{\mathrm{d}t} = V_z(\nabla \cdot \vec{v})_z, \tag{2.4}$$

which in the "support operators" terminology [9] yields a "discrete" definition of $(\nabla \cdot \vec{v})_z$, or of the divergence operator. To see this, consider that the zone volume depends on some arbitrary number of coordinates d ($p = 1, \ldots, d$), so that $V_z^n = V_z(\vec{r}_1^n, \ldots, \vec{r}_d^n)$; putting this form into Eq. (2.4) results in

$$\frac{\mathrm{d}V_z}{\mathrm{d}t} = \sum_{p=1}^{d}\sum_{i=1}^{3} \frac{\partial V_z}{\partial x_{p,i}} v_{p,i} \equiv \sum_{p=1}^{d} \vec{a}_p^z \cdot \vec{v}_p = V_z(\nabla \cdot \vec{v})_z, \tag{2.5}$$

where in the first part of this equation $\vec{r}_p$ consists of the three Cartesian components $x_{p,i}$, and the velocity components $v_{p,i}$ are defined by the fact that the points are Lagrangian, $v_{p,i} \equiv \mathrm{d}x_{p,i}/\mathrm{d}t$. The derivatives of the zone volume with respect to its coordinate dependence is used to define the surface vectors associated with point p, as was done in the work of Favorskii [7], and Margolin and Adams [8]. Thus the grid vectors used to construct the corner forces are not arbitrarily specified, as with the older versions of this type of hydrodynamics [3], but are a consequence of the chosen volume definitions. (One interesting case is area-weighted differencing, see [13].) Under this construction, we can write the equations for the grid vector and corner force associated with $P\,\mathrm{d}V$ work of point p and zone z as

$$a_{p,i}^z = \frac{\partial V_z}{\partial x_{p,i}}, \quad \vec{f}_z^p = P_z\vec{a}_p^z. \tag{2.6}$$

Referring to the definitions illustrated in Fig. 1 for point $p = 2$ of zone z we have that $\vec{a}_{p=2}^z = \vec{a}_2 + \vec{a}_3$, where for the quadrilateral shown the $\vec{a}_j$ vectors ($j = 1, \ldots, 8$) are outward normal to zone z with a magnitude of one-half of their respective edge length.

The preceding discussion justifies the claim that the work done by a zone pressure is a proper differencing of $P\,\mathrm{d}V$. However, Eq. (2.2) calculates the work done by all zone forces of any origin. This is an important generalization when one needs to calculate the work done by the artificial stresses used with this kind of hydrodynamics. Defining the surface area vectors of a prescribed volume as given by the derivative of that volume with respect to coordinates is essentially a generalization of the result in elementary calculus where the derivative of the area of a circle, or of the volume of a sphere, yields its surface area to the case where the volume in question does not have a simply differentiable boundary.

That the use of two different definitions of zone volume results in what may be described as an entropy error can be seen by considering the first law of thermodynamics for a zone z, written as

$$M_z\Delta e_z = -P_z(\rho_z = M_z/V_z^{\mathrm{crd}}; e_z)\Delta V_z^{\mathrm{cmp}} + (T\Delta S)_z, \tag{2.7}$$

where S is the entropy and T is the temperature. In this equation the density dependence of the zone pressure is a function of the coordinate volume V_z^{crd}, while the $P\,\mathrm{d}V$ work term involves the change in compatible volume V_z^{cmp} on a timestep. The entropy term completes the equation, and for adiabatic flow will not be exactly zero unless $V_z^{\mathrm{crd}} = V_z^{\mathrm{cmp}}$. The magnitude and implications of this difference are explored in the next section. However, it will be shown that using V_z^{cmp} to compute the density in Eq. (2.7) is not a valid way to make this equation internally consistent.

What has been demonstrated is that the volume change utilized in the discrete version of the specific internal energy written in compatible form is not the change in volume deduced from the difference in a zone volume calculated from its coordinates at two different times; but rather, this change in zone volume is calculated from the discrete evolution in time of the continuity equation for volume, Eq. (2.5). These two definitions of zone volume are not generally equal. The zone volume that is computed directly from its coordinates is hereon referred to as the "coordinate volume", while that computed from integrating Eq. (2.5) in time is termed the "compatible volume". How these two volumes differ and the consequences of this difference are the subject of the next section.

3. Internal consistency and accuracy

Given the previous discussion, it is convenient for the analysis that follows to contrast the change in volume of a zone calculated from its coordinates at two different times to that of the increment in zone volume as calculated from the evolution equation for volume. To this end, the change in coordinate zone volume $(\Delta V)_z^{\mathrm{crd}}$ during a timestep, where the coordinates are incremented by an amount $\Delta\vec{r}_p$ and $p = 1, \ldots, d$, is defined as

$$(\Delta V)_z^{\mathrm{crd}} \equiv V_z^{n+1} - V_z^n = V_z[(\vec{r}_1^n + \Delta\vec{r}_1) \cdots (\vec{r}_d^n + \Delta\vec{r}_d)] - V_z[\vec{r}_1^n \cdots \vec{r}_d^n]. \tag{3.1}$$

From the previous arguments, the change in compatible zone volume $(\Delta V)_z^{\mathrm{cmp}}$ is given by

$$(\Delta V)_z^{\mathrm{cmp}} = \sum_{p=1}^{d} \vec{a}_p^{z,*} \cdot \Delta\vec{r}_p, \tag{3.2}$$

where the time centering of the surface vectors $\vec{a}_p^{z,*}$ of a zone remains to be specified. Since we are interested in only the change in compatible zone volume on the final corrector step, these vectors are centered somewhere between time levels $* = n + \frac{1}{2}$ and $* = n + 1$, and not at time level $* = n$ as is the case for the predictor step. Note that the displacement vectors $\Delta\vec{r}_p$ of the grid points are the same in both of these equations.

A global internal consistency criterion can be constructed from a knowledge of both the coordinate and compatible zone volumes at any given time. The coordinate zone volume, $V_z^{\mathrm{crd}}(t^n)$, is always given as a known function of the coordinates; however, the compatible zone volume must be constructed as a diagnostic on each timestep. That is, to obtain $V_z^{\mathrm{cmp}}(t^n)$ the continuity equation (2.5) must be integrated with respect to time, and $V_z^{\mathrm{cmp}}(t^n)$ updated at the end of every timestep as an additional dependent variable. Thus we have that

$$V_z^{\mathrm{cmp}}(t^n) = V_z^{\mathrm{crd}}(t = 0) + \sum_{k=1}^{n} (\Delta V)_{z,k}^{\mathrm{cmp}}, \tag{3.3}$$

where the sum is over all timesteps k to time t^n. Given these volume definitions, two useful norms can be defined as

$$E_1(t^n) \equiv \left(\sum_{z=1}^{N_z} |V_z^{\mathrm{crd}}(t^n) - V_z^{\mathrm{cmp}}(t^n)|/V_z^{\mathrm{crd}}(t^n)\right) \Big/ N_z, \tag{3.4}$$

$$E_{\max}(t^n) \equiv \max_z |V_z^{\mathrm{crd}}(t^n) - V_z^{\mathrm{cmp}}(t^n)|/V_z^{\mathrm{crd}}(t^n), \tag{3.5}$$

where the sum in the first equation above, and the max operation in the second, is over all N_z zones of the grid. The average norm $E_1(t^n)$ and the max norm $E_{\max}(t^n)$ serve to measure different and complimentary properties

of a computation. Also note that the kernel, $|V_z^{\mathrm{crd}}(t^n) - V_z^{\mathrm{cmp}}(t^n)|/V_z^{\mathrm{crd}}(t^n)$, of these two norms is the same and is defined for every zone z. These values can be contoured spatially at any given time to yield additional detailed information about any simulation.

Our discussion of the difference in these two volume definitions begins in one dimension. Consider a set of zones in 1D with a numerical index j that ranges over the entire domain, where a single zone z (with index j) is defined by boundary points with index labels j and $j+1$ that lie to the left and right, respectively. Next the difference in these two volumes is considered in 1D Cartesian, cylindrical, and spherical geometry. The case of 1D Cartesian geometry is trivial in that, in this instance, the grid vectors $a_p^{z,*}$ are equal to unity and fixed in time. Thus both definitions of zone volume are exactly equal.

The first non-trivial case is 1D cylindrical geometry. Let the grid point coordinates be denoted as R_j, then for a zone with index $z = j$ it follows from Eq. (3.1) that the change in coordinate volume in a timestep is given by

$$\Delta V_z^{\mathrm{crd}} = \pi\left[(R_{j+1}^{n+1})^2 - (R_j^{n+1})^2 - (R_{j+1}^n)^2 + (R_j^n)^2\right] = 2\pi\left[R_{j+1}^n\Delta R_{j+1} + (\Delta R)_{j+1}^2/2 - R_j^n\Delta R_j - (\Delta R)_j^2/2\right], \tag{3.6}$$

where we have used the fact that $R_j^{n+1} = R_j^n + \Delta R_j$. To compute the corresponding compatible volume using Eq. (3.2), we specify the grid vectors as $a_{p=j}^{z,*} = 2\pi R_j^* = 2\pi(R_j^n + \alpha\Delta R_j^*)$, where ΔR_j^* is the displacement of point j from the predictor step, and $0 \leqslant \alpha \leqslant 1$. This yields the result

$$\Delta V_z^{\mathrm{cmp}} = a_{j+1}^{z,*}\Delta R_{j+1} - a_j^{z,*}\Delta R_j = 2\pi\left[(R_{j+1}^n\Delta R_{j+1} - R_j^n\Delta R_j) + \alpha(\Delta R_{j+1}^*\Delta R_{j+1} - \Delta R_j^*\Delta R_j)\right]. \tag{3.7}$$

By inspection these two equations are equal only if $\alpha = 1/2$ and $\Delta R_j^* = \Delta R_j$ for all points j. The first condition says that the grid vectors should be time-centered on the corrector step; the second is more severe and requires that the forces be exactly equal on both the predictor and corrector steps of a timestep cycle. This latter requirement is only true if the force is zero, since roundoff error in the calculation of the force will prevent this equality from holding in the case of non-zero forces.

If 1D spherical geometry is considered with the volume of a spherical zone given by $4\pi(R_{j+1}^3 - R_j^3)/3$, and with time-centered grid vectors $a_j^{z,*} = 4\pi(R_j^n + \Delta R_j^*/2)^2$, then repeating the calculation given above one finds that the change in these volumes is also not identical; their difference is given by

$$\Delta V_z^{\mathrm{crd}} - \Delta V_z^{\mathrm{cmp}} = -4\pi\left[\Delta R_{j+1}((\Delta R^*)_{j+1}^2/4 - (\Delta R)_{j+1}^2/3) - \Delta R_j((\Delta R_j^*)^2/4 - (\Delta R_j)^2/3)\right]. \tag{3.8}$$

This difference is seen to arise from terms that are cubic in ΔR_j. In the case of 1D spherical geometry, this defect can be remedied by a direct decomposition of the volume change of a spherical zone between time levels n and $n+1$ to yield an alternative form for the grid vectors in Eq. (3.8) as

$$a_j^{z,*} = 4\pi[(R_j^{n+1,*})^2 + R_j^{n+1,*}R_j^n + (R_j^n)^2]/3. \tag{3.9}$$

In the above equation, $R_j^{n+1,*}$ is the value of the coordinate of grid point j at the end of the predictor step, $R_j^{n+1,*} = R_j^n + \Delta R_j^*$. So if $\Delta R_j^* = \Delta R_j$, then $R_j^{n+1,*} = R_j^{n+1}$ and the compatible volume calculated by using the grid vectors defined in Eq. (3.9) exactly matches the coordinate volume. The type of truncation error manipulation involved in the derivation of Eq. (3.9) cannot always be effected in the multi-dimensional case. The fact that coordinate and compatible volume definitions are not always equal for uniform drift is not a real defect with compatible Lagrangian hydrodynamics since, as will be shown, this difference remains at very small values, and both volumes are by themselves exactly conserved.

Next we continue our comparison of coordinate and compatible volume definitions by considering the case of 2D Cartesian geometry with quadrilateral zones. For coordinates $\vec{r}_p^n$, the volume of a quadrilateral using the notation of Fig. 1 is $(\vec{r}_2 - \vec{r}_4) \times (\vec{r}_3 - \vec{r}_1)$. From Eq. (3.1) this results in

$$\begin{aligned}\Delta V_z^{\mathrm{crd}} &\equiv V_{\mathrm{crd}}^{n+1} - V_{\mathrm{crd}}^n \\ &= \frac{1}{2}[(\vec{r}_2 - \vec{r}_4)^n \times \Delta\vec{r}_3 + (\vec{r}_4 - \vec{r}_2)^n \times \Delta\vec{r}_1 + (\vec{r}_3 - \vec{r}_1)^n \times \Delta\vec{r}_4 + (\vec{r}_1 - \vec{r}_3)^n \times \Delta\vec{r}_2] \\ &\quad + \frac{1}{2}\{\Delta\vec{r}_2 \times (\Delta\vec{r}_3 - \Delta\vec{r}_1) + \Delta\vec{r}_4 \times (\Delta\vec{r}_1 - \Delta\vec{r}_3)\}.\end{aligned} \tag{3.10}$$

Using Eq. (3.2), where the grid vectors $\vec{a}_p^{z,*}$ are shown in Fig. 1 ($\vec{a}_{p=1} = \vec{a}_1 + \vec{a}_8$, etc.), results in

$$\begin{aligned}\Delta V_z^{\mathrm{cmp}} &\equiv \sum_{p=1}^{4} \vec{a}_p^{z,*} \cdot \Delta\vec{r}_p \\ &= \frac{1}{2}[\cdots \text{same as above} \cdots] + \frac{\alpha}{2}\{(\Delta\vec{r}_2^n \times \Delta\vec{r}_3 + \Delta\vec{r}_2 \times \Delta\vec{r}_3^n) - (\Delta\vec{r}_2^n \times \Delta\vec{r}_1 + \Delta\vec{r}_2 \times \Delta\vec{r}_1^n) \\ &\quad + (\Delta\vec{r}_4^n \times \Delta\vec{r}_1 + \Delta\vec{r}_4 \times \Delta\vec{r}_1^n) - (\Delta\vec{r}_4^n \times \Delta\vec{r}_3 + \Delta\vec{r}_4 \times \Delta\vec{r}_3^n)\},\end{aligned} \tag{3.11}$$

where the terms in square brackets are the same in both equations. Thus it is once again seen that if the grid vectors are time-centered ($\alpha = 1/2$), and also for zero force, then these equations and their associated time integrated volumes are identical.

In order to examine the difference in these two volumes further, consider how coordinates are advanced in time. With $\Delta\vec{r}_p^* = (\vec{v}_p^{n+1,*} + \vec{v}_p^n)\Delta t/2$ we can write

$$\begin{aligned}\Delta\vec{r}_p^* &= \vec{v}_p^n\Delta t + \vec{F}_p^n(\Delta t)^2/(2M_p), \\ \Delta\vec{r}_p &= \vec{v}_p^n\Delta t + \vec{F}_p^*(\Delta t)^2/(2M_p).\end{aligned} \tag{3.12}$$

Now let us pick out the first term in curly brackets in Eq. (3.10) and in Eq. (3.11) that refer to the points 2 and 3 of the quadrilateral in Fig. 1, and subtract these terms with $\alpha = 1/2$ to obtain

$$\begin{aligned}&[\Delta\vec{r}_2 \times \Delta\vec{r}_3 - \frac{1}{2}(\Delta\vec{r}_2^* \times \Delta\vec{r}_3 + \Delta\vec{r}_2 \times \Delta\vec{r}_3^*)] \\ &\quad = \frac{(\Delta t)^3}{4M}\left[\vec{v}_2^n \times (\vec{F}_3^* - \vec{F}_3^n) - \vec{v}_3^n \times (\vec{F}_2^* - \vec{F}_2^n)\right] + \frac{(\Delta t)^4}{4M^2}\left[\vec{F}_2^* \times \vec{F}_3^* - \frac{1}{2}(\vec{F}_2^n \times \vec{F}_3^* + \vec{F}_2^* \times \vec{F}_3^n)\right].\end{aligned}$$

This shows that the difference in coordinate and compatible volumes for 2D Cartesian geometry is of order $(\Delta t)^3$ on a single timestep. When integrating this difference to some global time t^n one factor of Δt is absorbed into the multiplying constant so that we deduce that the accuracy in time should be of order $(\Delta t)^2$.

The preceding arguments justify writing the average error norms defined in Eqs. (3.4), (3.5) in an approximate global form as

$$E_1(t^n) \approx \mathcal{O}\left(\sum_{z=1}^{N_z} (\Delta t)^q \frac{(\Delta x)^r}{V_z}\right) \Big/ N_z, \tag{3.13}$$

$$E_{\max}(t^n) \approx \mathcal{O}\left(\max_z (\Delta t)^q \frac{(\Delta x)}{V_z}\right). \tag{3.14}$$

In the above expressions, the powers q and r denote global orders of accuracy in space and time, respectively; notice that for the max-norm, r is set to unity in anticipation of the expected result. The zone dimensions may not be even close to uniform, so one may have quite different values of Δx and zone volume V_z across the grid at any given time. Because of how the truncation error enters in the preceding discussion, the approximate error estimators given in Eq. (3.13) are a product of space and time error, instead of a separate sum of such terms as usually results from a direct Taylor series expansion of a system of PDE's. The operational use of these error estimators requires some further explanation. As will be seen, although they can indicate the relative quality of one or a set of simulations, they do not always yield an unambiguous value for the spatial accuracy r, and thus require some care in their utilization.

To determine the powers in the error indicators defined above one performs a series of computations for a set of problems with different space and time resolution. Then by comparing the values of a given error norm at the same time for a given computation, one can calculate the values of the power law dependence, the values of q and r. There are essentially three different strategies for doing this calculation. First, with spatial resolution fixed, one scales the CFL number down by factors of two (starting with CFL = 0.25) and runs each simulation to the same final time; this freezes any dependence on the spatial factors in our error norms. Second, the Δt factors in this equation can be effectively held fixed by increasing the CFL number by a factor of two (while staying beneath the 0.25 level) and decreasing the grid spacing by a factor of two in the spatial direction

that determines the CFL number. Then one compares the values of Eq. (3.13) for different runs at the same final time; these runs will have executed approximately the same number of time cycles. Third, one can scale the spatial dimensions holding the CFL number fixed. Next we detail what one must consider when implementing these three strategies.

The first of the above strategies is the only one that is straightforward. By fixing the spatial resolution and scaling the CFL number, each time decreasing the CFL number by a factor of two, usually yields decreasing values of Eq. (3.13) on each iteration by two or four indicating a value of q of one or two, respectively. This is expected from the previous arguments that show that one should obtain either first or second order accuracy in time depending upon whether the grid vectors are placed full-forward or time-centered on the corrector step. The second strategy attempts to eliminate time dependence from our error estimators; however, for most Lagrangian calculations, the timestep varies significantly throughout any run, so what is fixed is the cycle number to reach a final time, and it is assumed that this eliminates the time dependence between runs. One must then carefully consider how the zone volume V_z changes with the scaled dimensions in order to conclude an order of spatial accuracy r. For instance, if the zone size decreases as $(\Delta x)^2$ and $E_1(t^n)$ remains constant, then one concludes second order spatial accuracy ($r = 2$). Alternatively, if $E_1(t^n)$ increases by the same factor that is used to decrease Δx, then one would conclude first order accuracy in space($r = 1$). The third strategy requires the most scrutiny in its use in that changing a spatial dimension may or may not affect timestep; additionally, the timestep may be determined first by one dimension and later by another, as can happen when a center of convergence is approached by a shock front. However, if the scaled dimension does determine the timestep and if the order of accuracy with respect to time is known (usually second order for centered grid vectors), then one can cancel a factor of $(\Delta t/\Delta x)^2$ in $E_1(t^n)$ and use the remainder to determine r. Again, one must carefully consider how V_z changes with the scaled spatial dimensions.

Before proceeding to the numerical tests, we wish to note that the local spatial accuracy of the type of finite-volume differencing used with Lagrangian hydrodynamics calculations is usually first order, and never higher than second order. This has been demonstrated previously in the appendix of [13] for 2D Cartesian geometry, but applies generally in any number of dimensions. A geometric illustration of this can be seen in 1D by referring to Fig. 2, in which is depicted a simple 1D grid that is uniform but with different zone spacing to the right and left of a central displacement. Such a grid represents a shock wave travelling from right to left with the shock position at the central two zones that are non-uniform. The spatial gradients that are calculated in this simple case are just spatial differences of quantities such as density and velocity centered at zones and points on a uniform grid, and are thus second order accurate for simple two-point differences. Where the grid is non-uniform the accuracy drops to first order, which is the shock position S in Fig. 2. This simple case often extends quite naturally to 2D and 3D in the case where the velocity field is said to be grid-aligned. Here the velocity difference $\Delta\vec{v}$ between any two adjacent grid points is parallel (or possibly zero) to the displacement vector between them. Configurations of this type in 2D or in 3D often have approximately uniform grid spacing except near the location of shocks. In these cases one can often operationally observe second order spatial accuracy with respect to the E_1 error indicator. Theoretically, these calculations are still first order in space with respect to any part of the grid that a shock wave has traversed, since a shock acts as an internal boundary condition applied by means of the artificial viscosity with first order spatial accuracy.

While it has been noted that the order of accuracy of Lagrangian calculations is usually low, it is not the case that our error estimators are meant to simply verify the obvious. They may do this to the extent that they verify that runs are reasonable in this regard. However, their most common use (mainly $E_1(t^n)$) is to indicate when numerical problems arise, and to show in more detail how convergence to a solution is reached. Some examples are given in the next section that illustrate these points.

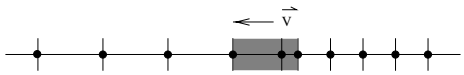


Fig. 2. 1D path of a shock (gray region) travelling from right to left with velocity $\vec{V}$. The left part of the domain is unperturbed, whereas the right part has been compressed due to the shock wave.

3.1. Numerical tests

A set of numerical tests are given to validate the error indicators previously defined, and to illustrate how they can be utilized to assess the quality of numerical simulations. These also serve to demonstrate that the size of the entropy error associated with the coordinate and compatible volumes is significant only when severe numerical difficulties, such as numerical instability, arise. The discussion is meant to show that the error estimators are of both a practical and a pedagogical utility. The test problems employed have all been published elsewhere, and thus their setups are not repeated; instead, the previously published results are referenced. Only enough detail is given to make the discussion of the tables given herein intelligible.

Our first example shows how the internal consistency estimator $E_1(t)$ can be used as a highly sensitive indicator of the development of numerical difficulties during any calculation. For this purpose the Guderley implosion problem is chosen with a resolution of 1000 zones radially and three 30° angles on a 90° quadrant in cylindrical geometry. We use area-weighted differencing to preserve 1D spherical symmetry (see Fig. 3 of [6] for both setup and results), and a CFL number of 0.25. For such resolution this problem has very long-thin zones that eventually develop hourglass difficulties that result in run termination unless subzone pressure forces that counteract this instability are employed [16]. Table 1 gives the cycle number, time, timestep, $E_1(t)$ norm, and total energy balance for this test case: first, using no stabilizing subzone pressure forces, and then with stabilizing forces and a merit factor of unity. The fact that the calculation has developed numerical difficulties in the first instance can be seen by the sudden increase in $E_1(t)$ by an order of magnitude, from 10^{-9} to 10^{-8}, between the time of 0.3 and 0.4; $E_1(t)$ then increases by three additional orders of magnitude at a time of 0.5, and the run terminates with timestep collapse about 1300 cycles later. Notice that the value of total energy balance, given in non-dimensional form in the last column of this table, remains at roundoff error level and that $E_1(t)$ never reaches a magnitude of order unity. In the second part of Table 1, the results are shown for the case where subzone pressure forces are utilized. This case runs to the final time of 0.8 without any hourglass problems. Although $E_1(t)$ starts out with a slightly larger value than that of the first run at an earlier time, it shows only a gradual increase as the shock wave travels inward, activating more cells of the grid which can then contribute a non-zero value to $E_1(t)$. When the shock wave reflects from the center of convergence just after a time of 0.75, the value of $E_1(t)$ increases by approximately an order of magnitude, and remains close to constant at the final value shown as the shock wave moves outward. In these runs the grid vectors are time-centered on the corrector step. Therefore, the magnitude of $E_1(t)$ is small compared to what one would expect the spatial truncation error to be for this grid resolution.

Table 1
Guderley problem in cylindrical geometry with CFL = 1/4

Cycle	Time t	Δt	$E_1(t)$	Total energy balance
Without anti-hourglass forces				
1709	0.2	1.113E − 4	3.242E − 9	−5.276E − 16
2635	0.3	1.046E − 4	3.835E − 9	5.921E − 17
3627	0.4	0.969E − 4	3.204E − 8	−2.419E − 15
4713	0.5	0.867E − 4	1.187E − 5	−3.612E − 16
6092	0.558	0.97E − 12	8.330E − 4	−1.827E − 15
⋯	⋯	Fails	⋯	⋯
With anti-hourglass forces				
1708	0.2	1.113E − 4	6.353E − 9	7.912E − 16
2634	0.3	1.046E − 4	7.260E − 9	2.368E − 15
3626	0.4	9.698E − 5	9.221E − 9	2.329E − 15
4710	0.5	8.744E − 5	1.153E − 8	3.106E − 15
⋮	⋮	⋮	⋮	⋮
13,122	0.8	1.515E − 5	5.526E − 5	7.332E − 16

We show results with and without anti-hourglass subzone pressure forces: without these forces the problem fails at time $t = 0.558$; with them the problem runs to completion. That the calculation develops numerical difficulties in the first instance is seen by the increase in $E_1(t)$. Total energy balance remains at roundoff error level.

In Fig. 3 the log of $E_1(t)$ versus time is plotted for the cases given in Table 1. The unstable case without anti-hourglass forces increases sharply, and eventually with a large linear slope, indicating exponential growth of the instability after $t \approx 0.4$. The stable case shows only linear growth (approximately constant on a semi-log plot), and increases significantly only after shock reflection from the center of convergence, an expected behavior that is due to the wall heating difficulty of all forms of artificial viscosity. Thus the $E_1(t)$ error norm should be monitored closely for all simulations. Any sudden increase in its value very often indicates a developing numerical problem that is cause for closer scrutiny. Subzone anti-hourglass pressure forces with merit factor, $M_f = 1$, are employed in all remaining simulations detailed in this paper.

The next example illustrates how the $E_1(t)$ error norm varies in both magnitude and in scaling when the grid vectors are time-centered or full-forward on the corrector step, and with varying CFL number between runs. Again the Guderley implosion problem is used with 400 radial zones and three 30° angles. In Table 2 results are given for four different runs. The left side of this table shows the approximate time and number of cycles for CFL numbers 0.25 and 0.125, and the $E_1(t)$ error norm for the case where the grid vectors are centered at the $n+\frac{1}{2}$ time level on the corrector step; on the right $E_1(t)$ is given when the grid vectors are full-forward at the $n+1$ time level on the corrector step. Note that there is approximately a three to four order of magnitude increase in the size of $E_1(t)$ between these two cases. This is expected from the analysis given earlier that showed that for grid vectors placed full-forward, the two definitions of zone volume differ at spatial truncation error level without consideration of any applied forces; the magnitude of $E_1(t)$ in this case gives an operational measure of spatial truncation error. Also, from the preceding analysis we expect to observe first order scaling in time with grid vectors placed full-forward, and second order accuracy in the time-centered case. Table 2 shows the ratio of the values of $E_1(t)$ between runs where the CFL number has been halved (cycle number to same final time doubled). One expects a ratio of about two for first order accuracy, and about four for second order accuracy. It is seen that for full-forward grid vectors, this is what is obtained; for grid vectors time-centered, values somewhat larger than four are observed. In general, and with other test problems, one sees this type of behavior as well, usually very close to factors of two or factors of four for these respective grid vector time centerings. The general conclusion is that if the grid vectors are time centered on the corrector step, second order accuracy in time is achieved. It is the spatial accuracy that invariably dominates and determines the quality of the solutions in Lagrangian hydrodynamics calculations. In the remaining simulations,

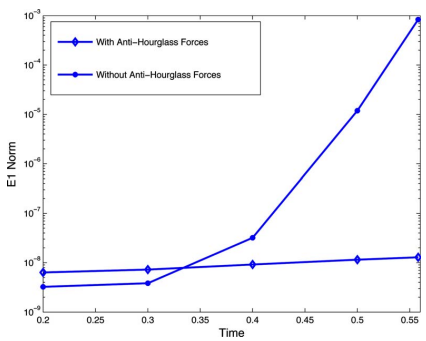


Fig. 3. Semi-log plot of the E_1 norm in time for the Guderley problem in cylindrical geometry (CFL = 1/4). Without anti-hourglass forces, the E_1 norm increases sharply showing exponential growth of the instability after $t \approx 0.4$. With anti-hourglass forces, the stable case, growth is linear and a significant increase in $E_1(t)$ occurs only after shock reflection from the center of convergence.

Table 2
Guderley problem in cylindrical geometry

CFL	Cycle	Time t	Centered grid vectors		Forward grid vectors	
			$E_1(t)$	Ratio	$E_1(t)$	Ratio
1/4	2960	0.7	**1.71E − 7**	–	1.17E − 3	–
	3433	0.75	3.00E − 6	–	5.13E − 3	–
	5054	0.8	1.01E − 5	–	1.03E − 2	–
1/8	5922	0.7	**2.97E − 8**	**5.75**	5.54E − 4	1.37
	6874	0.75	4.73E − 7	6.34	2.55E − 3	2.01
	10087	0.8	1.34E − 6	7.53	5.12E − 3	2.01
				⩾4		∼2

When the grid vectors are time-centered, the ratios $E_1(\text{CFL} = 1/4)/E_1(\text{CFL} = 1/8)$ are larger than four (e.g., **Ratio = 1.71E − 7/2.97E − 8 = 5.75**). Whereas when the grid vectors are full-forward, the ratios are approximately two.

the grid vectors are always time-centered on the corrector step. The data presented throughout this paper constitute a representative sample taken from many simulations.

Before proceeding, we wish to mention that in the context of area-weighted differencing, multiple correctors can and have been used to advance a timestep. In this instance the forces used to compute both $\Delta\vec{r}_p^n$ and $\Delta\vec{r}_{p'}$ in Eq. (3.13) are formally at the $n+\frac{1}{2}$ time level, and we observe a decrease in the values of $E_1(t)$ by about a factor of two with two correctors over that seen with one time-centered corrector step. The use of more than two correctors does not result in any further decrease in $E_1(t)$ as roundoff error is always present to seed this error. The use of multiple correctors in this context has not been found to improve the quality of any numerical results and is not recommended.

Our third example contrasts the scaling of the $E_1(t)$ error norm for the Noh (see [15] for setup) and Guderley problems with time-centered grid vectors but with increasing radial resolution and increasing CFL number so that the number of cycles to run the Guderley problem to the quoted time of 0.74 is about 12,000, and the Noh problem to a time of 0.05 is about 1000. This is an example of the second manner in which $E_1(t)$ can be utilized, in this case with the dependence on timestep frozen out. Here we attempt to assess the order of spatial accuracy by considering the scaling of the $E_1(t)$ error norm. In cylindrical geometry the grid is scaled in the radial direction, using 400, 800, and 1600 radial zones, respectively, with three fixed 30° angles. It is seen from Table 3 that the $E_1(t)$ error norm increases only modestly as the grid is refined radially for the Guderley problem, while it increases by somewhat more than a factor of two for the Noh problem. The Guderley problem is a running shock wave that has not reflected by a time of 0.74, while the Noh problem is a stagnation shock that shows a severe wall heating problem at the center of convergence of the grid. If one looks at the value of the kernel of the error norm $E_1(t)$ defined for each zone, one finds that as the Noh problem is refined in the radial direction, this error is constant as a function of logical radial index number from the origin. This is a numerical example of non-uniform spatial convergence, and we expect this problem to show only first order spatial accuracy with respect to the $E_1(t)$ error norm. Thus, if we conclude that the zone volume V_z in $E_1(t)$

Table 3
Guderley and Noh problems in cylindrical geometry

Grid	CFL	Cycle	Time t	$E_1(t)$	Ratio
Guderley problem					
400 × 3	1/16	∼12,000	0.74	**1.90E − 8**	**1.26**
800 × 3	1/8			**2.39E − 8**	1.39
1600 × 3	1/4			3.32E − 8	–
Noh problem					
400 × 3	1/32	∼1000	0.05	3.18E − 7	2.37
800 × 3	1/16			7.53E − 7	2.43
1600 × 3	1/8			1.83E − 6	–

The grid is refined as the CFL increases such that the same number of time steps is performed to reach the final time (12,000 for Guderley problem, 1000 for Noh). Ratios are computed by dividing two consecutive norms.

decreases as the square of the radial zone spacing, then for first order accuracy of the force differencing in the Noh problem ($r = 1$), one expects that $E_1(t)$ should increase by factors of two as the number of radial zones is successively doubled. Likewise, if the force is effectively calculated with second order spatial accuracy ($r = 2$), then it is expected that $E_1(t)$ should remain approximately constant. This is close to what is seen for the Guderley problem; these results are given in Table 3. We conclude that this running shock wave problem shows approximate second order spatial accuracy with respect to the $E_1(t)$ error norm. This is because the zones that contain the shock wave contribute a negligible contribution to $E_1(t)$ because they are few in number. Order of accuracy is always a function of the norm used to measure it [2].

In the following example we employ the third strategy that was mentioned for utilizing our internal consistency estimators. The Guderley problem is computed in 2D cylindrical geometry with a CFL number of 0.25; the number of radial zones is scaled from 800 to 1600 with three 30° angles. The scaling of both the $E_1(t)$ and the $E_{\max}(t)$ norms is examined; these results are given in Table 4. Notice that the number of cycles needed to reach a given time has doubled with an increase in grid size since it is the radial spacing of grid points that determines the timestep for 30° angles before the shock wave intersects the center of convergence. If we make the the same assumption as in the previous example regarding the scaling of the zone size V_z with the number of radial zones, then the spatial part of the $E_1(t)$ norm should be close to constant. Thus a factor of approximately four in the ratio of these norms comes from their $(\Delta t)^2$ time dependence. This reasoning and the results in Table 4 indicate second order accuracy in both space and time. Likewise, the decrease in the $E_{\max}(t)$ norm by about a factor of two indicates a situation where there is second order accuracy in time but only first order in space, since the growth in the spatial part of this norm cancels one-half of the decrease due to second order time dependence. First order accuracy is what one expects with respect to the $E_{\max}(t)$ norm. If one locates the position in the grid where the value of the max-norm is achieved, one finds that it tracks the shock location exactly. That is, the value of the kernel that makes up these norms has its maximum at the shock location, and for a given zone, relaxes in magnitude by about a factor of 20 after the shock wave has heated this zone and moved by it. The type of reasoning just employed must be used carefully in that the conclusions depend on both the scaling of V_z with grid dimensions, and that the timestep is determined by the scaled dimension, which is certainly not always the case.

The Sedov blast wave problem is now considered. The precise setup used is given in [13]; it is run in 2D cylindrical geometry using area-weighted differencing with grids consisting of square zones of size 45×45 and 90×90, and with CFL numbers of 0.25 and 0.125. The internal energy is zero in all zones except the single zone at the origin. This zone contains the same total internal energy in all four cases, so the specific internal energy is a factor of eight larger when going from a 45×45 to a 90×90 grid size. The Sedov problem has always represented a particular challenge for all of the older non-energy-preserving Lagrangian algorithms in that they show a loss of total energy of about 10%. This energy is lost in the first few timesteps and is large enough to raise serious questions about the quality of the solution. For our compatible formulation, total energy conservation is achieved, but the $E_{\max}(t)$ norm has a relatively high value for these cases. If we consider only the error kernel for the inner zone where all internal energy is initially located, we find that this norm oscillates in time with an approximate period of about 100 cycles and an amplitude of about 1–0.5% and does not rapidly decay. This is a rather large value for this quantity compared to our other cases; however this inner

Table 4
Guderley problem in cylindrical geometry for CFL = 1/4

Grid	CFL	Cycle	Time t	$E_1(t)$	Ratio	$E_{\max}(t)$	Ratio
800×3	1/4	5973	0.7	**6.13E − 8**	–	1.52E − 5	–
		6964	0.75	8.43E − 7	–	2.77E − 4	–
		10,367	0.8	3.60E − 6	–	1.14E − 3	–
1600×3	1/4	14,992	0.7	**1.15E − 8**	**5.33**	8.74E − 6	1.74
		14,027	0.75	1.92E − 7	4.39	1.44E − 4	1.92
		21,211	0.8	9.28E − 7	3.88	8.41E − 4	1.36
					∼4		∼2

The results show that when the grid is refined, the ratios of E_1 norms (**Ratio = 6.13E − 8/1.15E − 8 = 5.33**) are near four, whereas they are approximately two for the ratios of the $E_{\max}$ norms.

zone expands by about a factor of ten in size (10^3 in volume), so interpreting this difference as a measure of local truncation error about this zone is not unreasonable. If we investigate the error kernel of other zones in the grid it is found that they all oscillate in time with an amplitude that decreases with radial location from the origin, but is constant with respect to logical grid index number when the number of initial zones is doubled in each dimension.

The results of this problem for the four cases mentioned are detailed for the $E_1(t)$ error norm in Table 5. Unlike in the other examples considered, it is seen that this norm does not change as the CFL number is halved for a fixed initial grid size. This appears to be connected to the oscillation of this error at all grid points, something that is not seen in our other calculations. There is a decrease in the $E_1(t)$ norm by a factor of between two and three when going from 45×45 to 90×90 zones with the CFL number held fixed. Notice from Table 5 that it also takes about three times the number of cycles in this instance to reach the same final time. So just as for the older Lagrangian algorithms, this problem is still somewhat of an enigma.

The last example that is considered is the Guderley implosion problem in 3D Cartesian geometry. This is run with a spherical initial grid on an octant with three 30° angles in both θ and ϕ, and with 100 zones in the radial coordinate with unit initial domain. The relevance of this problem is to show what happens when the symmetry correction factors given in [18] are turned on and off. Corrections to the grid vectors are necessary if a calculation in 3D Cartesian geometry is to preserve 1D spherical symmetry when present in the initial and boundary conditions, as is true for this case. The results from two simulations that are identical except for this symmetry correction procedure are given in Table 6. There it is seen that the magnitude of the $E_1(t)$ norm is about three to four orders larger for the case where the symmetry corrections are not utilized. With symmetry corrections off, the flow field deviates from being 1D spherical, the velocity vectors are no longer grid-aligned, and the grid distorts in space. This can be seen from the grid and density contours shown in Fig. 4.

Table 5
Sedov problem in cylindrical geometry

Grid	CFL	Cycle	Time t	$E_1(t)$	Ratio CFL	Ratio grid
45×45	1/4	369	0.5	*5.60E − 5*	–	–
		551	1.0	7.38E − 5	–	–
45×45	1/8	677	0.5	*5.43E − 5*	*1.03*	–
		996	1.0	**6.80E − 5**	1.09	–
90×90	1/4	1031	0.5	2.23E − 5	–	2.51
		1590	1.0	2.56E − 5	–	2.88
90×90	1/8	1846	0.5	2.07E − 5	1.08	2.62
		2843	1.0	**2.37E − 5**	1.08	**2.87**

Ratio CFL (Grid) is the ratio between the E_1 norms obtained for the same grid (CFL number) at the same time but for two different CFL numbers (grids). Notice that Ratio CFL is approximately one and does not improve.

Table 6
Guderley problem in 3D Cartesian geometry for CFL = 1/4

Cycle	Time t	Δt	$E_1(t)$	Total energy balance
With symmetry corrections				
297	0.30	1.012E − 3	2.757E − 7	−3.257E − 15
636	0.60	7.386E − 4	7.645E − 7	−4.312E − 15
897	0.75	0.510E − 4	1.215E − 5	−4.332E − 15
Without symmetry corrections				
299	0.30	1.000E − 3	8.839E − 4	−4.764E − 16
666	0.60	0.521E − 3	2.313E − 3	−9.746E − 16
1044	0.75	0.574E − 4	1.960E − 2	3.726E − 15

Although total energy is conserved to round-off error, the E_1 norm shows relatively large values when no symmetry corrections are applied; these numbers reflect the fact that 1D symmetry is broken, as seen in Fig. 4.

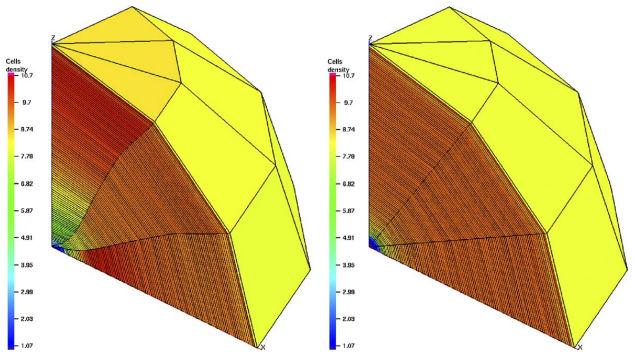


Fig. 4. 3D Guderley problem at $t = 0.75$ in Cartesian geometry. The mesh is $100 \times 3 \times 3$ on an octant of the unit sphere. Grid and density contours are shown. Left: 1044 time steps with symmetry corrections off (the loss of 1D symmetry is quantified in Table 6 where the E_1 norm is of truncation error size). Right: 897 time steps with symmetry corrections on.

The lack of 1D symmetry preservation in 3D Cartesian geometry with an initially spherical grid is not a numerical instability, as was the hourglassing case of our first example. This is seen by the fact that although the $E_1(t)$ norm is about three to four orders of magnitude larger with symmetry corrections off, compared to the case with them on, these norms scale with time in the same manner. These calculations illustrate the ability of the internal consistency error norms to detect numerical problems of differing origins with respect to any given simulation.

4. Numerical stability

For any computational algorithm to be useful it must be numerically stable. By numerical stability one means that if the underlying set of continuum of equations is stable, which says that the continuum system has no growing solutions, then the differenced form of this system should also contain no growing solutions. This is a necessary condition for a useful numerical algorithm. If the underlying continuum system of equations has energy source terms that allow growing solutions, then these terms are omitted from the stability analysis. That is, a numerical stability analysis only makes sense for a continuum system of equations that is already stable. For terms that produce growth of a solution one is only interested in the numerical accuracy with which these terms are differenced. Often very simple subsystems of equations yield the correct numerical stability results for much more complex ones; however, the selection of a proper subsystem can be a bit of an art form. In our case, the simple second order wave equation in one space dimension, written as two coupled first order equations, is utilized for this purpose. These equations are discretized in the next subsection, and the stability of the 3-step predictor–corrector scheme described earlier is analyzed using Fourier analysis [2]. The results reveal a stability diagram for constant space step and time step that has not been given previously.

It is difficult to empirically observe a precise stability boundary for a Lagrangian hydrodynamics algorithm because such calculations are typically run with a constant CFL number, but with a grid spacing and a time-step that can vary greatly during a simulation. Also, the definition of the scale length used in computing the CFL number with respect to non-uniform zones is not unique. In order to verify the results of the stability analysis derived using Fourier analysis of our simple system of equations, a numerical test problem is

constructed that eliminates the above noted difficulties. This test problem is then used to substantiate the analytical results for the full system of equations in all three spatial dimensions.

4.1. Stability utilizing the 1D second order wave equation

The 1D second order wave equation written as a coupled first order system with unit velocity is given by

$$\frac{\partial u}{\partial t} = \frac{\partial v}{\partial x}, \quad \frac{\partial v}{\partial t} = \frac{\partial u}{\partial x}, \tag{4.1}$$

where $u = u(x,t)$ can be thought of as the velocity, so that the first equation is that for momentum, and $v = v(x,t)$ is the pressure or density in the more general system given by Eqs. (2.1), (2.2). Note that in our simplified model there is no equation of state (EOS), so that if this model yields meaningful results we expect these results to be independent of the form of the EOS of the more general equations.

On a 1D domain $\Omega = [x_{\min}, x_{\max}]$, a mesh is defined by N_z zones/cells and $N_p = N_z + 1$ points such that $x_j = x_{\min} + (j-1)\Delta x$ where $\Delta x = x_{j+1} - x_j$ is the size of a cell $j + \frac{1}{2}$. Time is discretized using a time step $\Delta t = t^{n+1} - t^n$ and we define the CFL number as $r \equiv \Delta t/\Delta x$. For our stability analysis, we assume that Δx and Δt are constant in time. The velocity u is discretized on node j at time $t^n = n\Delta t$ as u_j^n, whereas v is placed at the center of the cell $j + \frac{1}{2}$ as $v_{j+\frac{1}{2}}^n$. The numerical scheme is denoted by $\mathbf{S}_\alpha$, where $0 \leqslant \alpha \leqslant 1$, and is defined by the following equations

- *Predictor step:* u_j^*, v_{j+1}^* at time $t^n + \Delta t$; $\bar{v}_{j+\frac{1}{2}}^*$ at time $t^n + \alpha\Delta t$

$$u_j^* = u_j^n + r\left(v_{j+\frac{1}{2}}^n - v_{j-\frac{1}{2}}^n\right), \tag{4.2}$$

$$v_{j+\frac{1}{2}}^* = v_{j+\frac{1}{2}}^n + r\left(\frac{1}{2}(u_{j+1}^* + u_{j+1}^n) - \frac{1}{2}(u_j^* + u_j^n)\right), \tag{4.3}$$

$$\bar{v}_{j+\frac{1}{2}}^* = \alpha v_{j+\frac{1}{2}}^* + (1-\alpha)v_{j+\frac{1}{2}}^n. \tag{4.4}$$

- *Corrector step:* uses the * values to increment u^n and v^n

$$u_j^{n+1} = u_j^n + r\left(\bar{v}_{j+\frac{1}{2}}^* - \bar{v}_{j-\frac{1}{2}}^*\right), \tag{4.5}$$

$$v_{j+\frac{1}{2}}^{n+1} = v_{j+\frac{1}{2}}^n + r\left(\frac{1}{2}(u_{j+1}^{n+1} + u_{j+1}^n) - \frac{1}{2}(u_j^{n+1} + u_j^n)\right). \tag{4.6}$$

This is a two parameter system: the CFL number r, and the parameter α that time-averages the pressure from the predictor step to the momentum equation on the corrector step. With $\alpha = 0$ the predictor and corrector steps are the same and the scheme is known to be unconditionally unstable. A few additional remarks on this discretization are in order; first, in the equations used to advance $v_{j+1/2}^*$ and $v_{j+1/2}^{n+1}$ in time, an average of the values of u at the advanced and old time levels is used on the RHS of Eqs. (4.3), (4.6) instead of a full-forward value. This is due to the fact that in the equations that are actually solved numerically, Eq. (2.2), also uses an average of old and advanced velocity to form the displacement $\Delta \vec{r}_p$. This time centering is necessary if we expect the simple model to accurately represent the compatible hydrodynamics equations that we wish to solve. The time centering parameter α in the corrector step, that is used in Eq. (4.5), has two important values: $\alpha = 1/2$ is referred to as time-centered, and $\alpha = 1$ is referred to as full-forward.

By substituting Eqs. (4.2)–(4.4) into (4.5) and (4.6) we obtain explicit formulas for u and v at time t^{n+1}:

$$u_j^{n+1} = u_j^n + r\left(v_{j+\frac{1}{2}}^n - v_{j-\frac{1}{2}}^n\right) + \alpha r^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n) + \frac{1}{2}\alpha r^3\left(v_{j+\frac{3}{2}}^n - 3v_{j+\frac{1}{2}}^n + 3v_{j-\frac{1}{2}}^n - v_{j-\frac{3}{2}}^n\right), \tag{4.7}$$

$$v_{j+\frac{1}{2}}^{n+1} = v_{j+\frac{1}{2}}^n + r(u_{j+1}^n + u_j^n) + \frac{1}{2}r^2\left(v_{j+\frac{3}{2}}^n - 2v_{j+\frac{1}{2}}^n + v_{j-\frac{1}{2}}^n\right) + \frac{1}{2}\alpha r^3(u_{j+2}^n - 3u_{j+1}^n + 3u_j^n - u_{j-1}^n) + \frac{1}{4}\alpha r^4\left(v_{j+\frac{5}{2}}^n - 4v_{j+\frac{3}{2}}^n + 6v_{j+\frac{1}{2}}^n - 4v_{j-\frac{1}{2}}^n + v_{j-\frac{3}{2}}^n\right). \tag{4.8}$$

A convenient way to analyze such schemes, and which is due to Von Neumann [2], is to assume a solution of the form: $u_j^n = u^0\lambda^n e^{ij\theta\Delta x}$ and $v_{j+\frac{1}{2}}^n = v^0\lambda^n e^{i\theta(j+\frac{1}{2})\Delta x}$ where θ is a real parameter, and $i = \sqrt{-1}$. The quantity $\lambda = \lambda(\theta)$ is determined by substituting these forms into Eqs. (4.7) and (4.8). After some algebraic manipulation, these equations can be written as the linear system

$$\mathbf{M}\begin{pmatrix} u^0 \\ v^0 \end{pmatrix} \equiv \begin{pmatrix} \lambda + A & iC \\ iC & \lambda + B \end{pmatrix}\begin{pmatrix} u^0 \\ v^0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{4.9}$$

where A, B, C are defined by

$$A = -1 + 4\alpha r^2 s^2, \quad B = -1 + 2r^2 s^2(1 - 2\alpha r^2 s^2), \quad C = 2rs(1 - 2\alpha r^2 s^2)$$

and $s = \sin(\theta/2)$. If this system is to be solved for any vector (u^0, v^0), the determinant of the matrix $\mathbf{M}$ must be zero, that is, $\det(\mathbf{M}) = \lambda^2 + \lambda(A + B) + AB - C^2 = 0$. Thus, $\det(\mathbf{M})$ is a polynomial in λ, the complex roots of which are

$$\lambda^{\pm} = -\frac{A+B}{2} \pm \sqrt{\frac{(A-B)^2}{4} + C^2}. \tag{4.10}$$

For numerical stability the solution vector must not increase in magnitude with time, or $|(u^{n+1}, v^{n+1})| \leqslant |(u^n, v^n)|$, must hold. Thus we require that $|\lambda^{\pm}| \leqslant 1$. Our objective is to characterize the maximum value of the CFL number r such that, given a particular value of α, $|\lambda^{\pm}| \leqslant 1$.

$\lambda^{\pm} \in \mathbb{C}$: In this case $\frac{(A-B)^2}{4} + C^2 < 0$, and we can instead consider

$$|\lambda^{\pm}|^2 = \left(\frac{A+B}{2}\right)^2 - \frac{(A-B)^2}{4} - C^2 = \frac{AB}{2} - C^2 = 1 - 2(2\alpha - 1)r^2 s^2 - 4\alpha r^4 s^4.$$

Two cases must be considered:

- if $\alpha < 1/2$, then $-2(2\alpha - 1)r^2 s^2 > 0$ and there exists r small enough such that $-2(2\alpha - 1)r^2 s^2 > 4\alpha r^4 s^4$ for any s leading to $|\lambda^{\pm}|^2 > 1$; the scheme $\mathbf{S}_{\alpha<1/2}$ is therefore unconditionally unstable;
- if $\alpha \geqslant 1/2$, then $-2(2\alpha - 1)r^2 s^2 - 4\alpha\, r^4 s^4 \leqslant 0$ for any s resulting in $|\lambda^{\pm}|^2 \leqslant 1$ independent of r.

$\lambda^{\pm} \in \mathbb{R}$: In this case $\frac{(A-B)^2}{4} + C^2 \geqslant 0$, and we need only to focus on the case $\alpha \geqslant 1/2$, as the case $\alpha < 1/2$ has just been shown to be unconditionally unstable (see above). Several examples are depicted in Fig. 5, where the discriminant (thick line) and $\lambda^{\pm} \in \mathbb{R}$ (thin lines) are plotted. These plots show for which θ the discriminant is positive. Moreover, for stability we need $-1 \leqslant \lambda^{\pm} \leqslant 1$. It can easily be shown that the maximum value of $|\lambda^{\pm}|$ is always reached for the largest wave number $\theta = \pm\pi$. Therefore, we compute the eigenvalues when $s = \sin(\frac{\theta}{2}) = 1$.

- $\alpha = 1$: the eigenvalues are given by (see Fig. 6 left curve)

  $$\lambda^{\pm} = 1 - r^2(3 - 2r^2) \pm r\sqrt{-4 + 17r^2 - 12r^4 + 4r^6},$$

  using $r = 1$, we obtain $|\lambda^{\pm}| = |\pm\sqrt{5}| > 1$. The scheme is unstable in this case. On this other hand, choosing the value $r = 1/\sqrt{2} = 1/\sqrt{2\alpha}$ yields $|\lambda^{\pm}| = 1$. (In fact, if $r > 1/\sqrt{2}$, then $|\lambda^{\pm}| > 1$ resulting in instability.) Therefore, for $\alpha = 1$ and $\lambda^{\pm} \in \mathbb{R}$ one expects stability for the CFL condition: $r \leqslant 1/\sqrt{2}$.
- $\alpha = 3/4$: the eigenvalues are given by (see Fig. 6 middle curve)

  $$\lambda^{\pm} = 1 - \frac{r^2}{2}(5 - 3r^2) \pm \frac{r}{2}\sqrt{-16 + 49r^2 - 30r^4 + 9r^6},$$

  which is exactly equal to ± 1 if $r = 1/\sqrt{2/3} = 1/\sqrt{2\alpha}$. (Again if $r > \sqrt{2/3}$, then $|\lambda^{\pm}| > 1$.) Therefore, for $\alpha = 3/4$ and $\lambda^{\pm} \in \mathbb{R}$ one expects stability for the CFL condition: $r \leqslant \sqrt{2/3}$.
- $\alpha = 1/2$: the eigenvalues are given by (see Fig. 6 right curve)

  $$\lambda^{\pm} = 1 - r^2(2 - r^2) \pm r\sqrt{-4 + 8r^2 - 4r^4 + r^6},$$

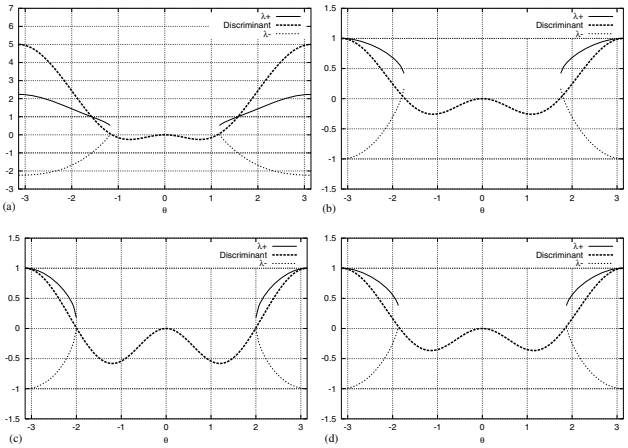


Fig. 5. Discriminant (thick dashed line) and real eigenvalues $\lambda^{\pm}$ (thin lines) as a function of $\theta \in [-\pi, \pi]$. (a) If $\alpha = 1$ and $r = 1$, then there exist $\theta \in [-\pi, \pi]$ such that $|\lambda^{\pm}| > 1$. (b) If $\alpha = 1$ and $r = 1/\sqrt{2\alpha}$, then $|\lambda^{\pm}| \leqslant 1$. (c) If $\alpha = 1/2$ and $r = 1$, then $|\lambda^{\pm}| \leqslant 1$. (d) If $\alpha = 3/4$ and $r = 1/\sqrt{2\alpha}$, then $|\lambda^{\pm}| \leqslant 1$.

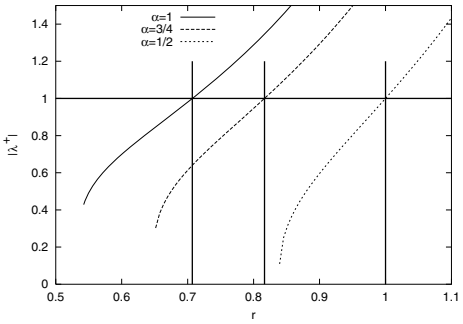


Fig. 6. $|\lambda^{+}|$ as a function of r for $\lambda^{+} \in \mathbb{R}$. For stability $|\lambda^{+}|$ must be less than 1 (thick horizontal line). For which $r_{\max}$ (vertical dashed line) do we have $\lambda^{+} = 1$? Left curve: for $\alpha = 1$, $r_{\max} = 1/\sqrt{2}$. Middle curve: for $\alpha = 3/4$, $r_{\max} = 1/\sqrt{2 \cdot 3/4}$. Right curve: for $\alpha = 1/2$, $r_{\max} = 1/\sqrt{2 \cdot 1/2} = 1$.

which is exactly equal to ± 1 if $r = 1 = 1/\sqrt{2\alpha}$. (As is now expected, if $r > 1$, then $|\lambda^{\pm}| > 1$, which results in numerical instability.) Therefore, for $\alpha = 1/2$ and $\lambda^{\pm} \in \mathbb{R}$ one expects stability for the classical CFL condition: $r \leqslant 1$.

In summary, the schemes $\mathbf{S}_{\alpha}$ are unconditionally unstable if $\alpha < 1/2$, and stable if $\alpha \geqslant 1/2$ and if the CFL condition $r \leqslant 1/\sqrt{2\alpha}$ is obeyed (see Fig. 7). Typically, a CFL number less than the maximum allowed for stability is used, most commonly the CFL number $r = 1/4$. In Fig. 8, $|\lambda^{+}|$ (for three schemes $\alpha = 1, \frac{3}{4}, \frac{1}{2}$) is plotted

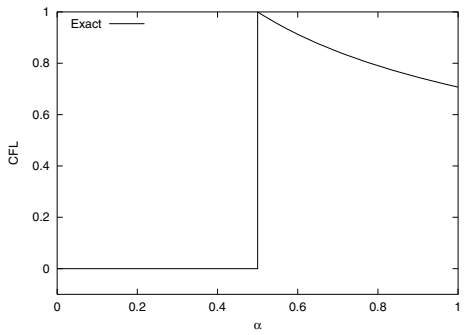


Fig. 7. CFL condition as a function of α for the family of schemes $\mathbf{S}_{\alpha}$. The schemes $\mathbf{S}_{\alpha}$ are unconditionally unstable for $\alpha < 1/2$; stable for $\alpha \geqslant 1/2$ and with CFL condition $r \leqslant 1/\sqrt{2\alpha}$.

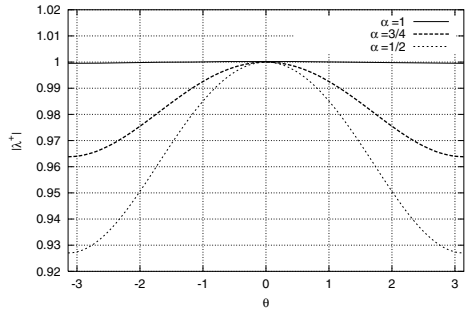


Fig. 8. Dissipation of the schemes $\mathbf{S}_1$, $\mathbf{S}_{\frac{3}{4}}$, $\mathbf{S}_{\frac{1}{2}}$ for a fixed CFL number $r = 1/4$. $|\lambda^{+}|$ is plotted as a function of $\theta \in [-\pi, \pi]$. ($\alpha = 1, \frac{3}{4}, \frac{1}{2}$ are denoted by straight, thick dashed, dashed lines, respectively).

as a function of $\theta \in [-\pi, \pi]$ for the fixed CFL number $r = \frac{1}{4}$. This plot shows that the scheme defined by $\alpha = 1/2$, which lies on the stability boundary, is the least dissipative of the family $\alpha \geqslant 1/2$. The full-forward scheme $\mathbf{S}_1$ damps more than any other.

4.2. Stability of the compatible hydrodynamics equations

The previous stability analysis has been performed for the 1D coupled system of wave equations using a discretization that, although simplified, does indeed correspond to that utilized for the more general system. In order to verify that these results are valid in the more general case, the full compatible hydrodynamic system of equations are solved for the following model problem. Consider the fluid equations in 1D, 2D, and 3D Cartesian geometry with the initial conditions: uniform density $\rho = 1$, ideal gas law EOS with $\gamma = 5/3$, a sound speed of $c_s = 1$, and velocity $\vec{v} = 0$. A unit domain with uniform zone size is constructed: 10, 10^2, 10^3 zones typically, but we also use slightly larger values to rule out surface effects since reflective boundary conditions are imposed on all faces of the domain. In 1D only, the pressure is seeded with a random perturbation at the 10^{-15} level whenever the EOS is called; in 2D and in 3D, the roundoff error that automatically occurs when calculating the grid vectors that are used to construct the corner force for each predictor and corrector stage of a timestep is sufficient to seed numerical instability (in 1D Cartesian geometry these are fixed at unity). Since for this problem no velocity should develop, simulations are run for a very large number of time cycles, typically 10^5, and with varying values for the CFL number r, and time-centering parameter α, in the corrector step. A sensitive gauge that provides a useful way to monitor the stability of any given simulation is to track the total kinetic energy $K^{\mathrm{CFL}}(t^n) \equiv \frac{1}{2}\sum_p M_p(\vec{v}_p^n)^2$. Since the density and domain size are scaled to unity, this number should remain at the square of machine precision, about 10^{-28}–10^{-30} in our case. For an unstable scheme it is observed that $K^{\mathrm{CFL}}(t^n)$ grows by several orders of magnitude long before the 10^5 cycle limit is reached. Using this test one can very accurately scope out the stability boundary in r and α space for all three dimensions. The 1D results are found to exactly match those produced by our analysis of the simplified 1D system of equations (4.1) (see Fig. 7); the 2D and 3D results are also found to be identical to those of the simplified 1D system of equations. Thus we conclude that the stability analysis performed on the 1D simplified system of equations is valid for the full hydrodynamic equations in all spatial dimensions. Although the artificial viscosity is turned off in this test problem, this term acts only as an addition to the pressure as far as stability is concerned. This point is stressed in the original paper that introduced this concept [4]. We also note that the same stability diagram (Fig. 7) is obtained for this test problem using an isothermal EOS, and further supports our conclusion that the numerical stability of the compatible Lagrangian hydrodynamics equations is EOS independent.

5. Conclusions

The theme of this paper is to clarify numerical issues concerning the internal consistency, stability, and accuracy of a recent discretization of the equations of fluid dynamics cast into Lagrangian form that we refer to as the discrete, compatible formulation. The proper resolution of the issues mentioned are central to the understanding of any numerical algorithm. Although Lagrangian hydrodynamics algorithms are very old, they languished until recently as attention in computational fluid dynamics turned to broader areas and domains of interest. The central feature of this more modern form of Lagrangian hydrodynamics is its ability to exactly conserve mass, momentum, and total energy without the need to use these quantities directly as dependent variables. It instead retains density, velocity, and specific internal energy as dependent variables as did the earlier version of this algorithm; these variables are more appropriate to compressible high speed flow calculations.

The internal consistency of this algorithm was investigated by analyzing the difference between the two definitions of zone volume that it utilizes. This difference comes about because of the subtle relationship of surface area to volume that characterizes any closed volume, and thus is seen to arise quite generally and naturally. It was shown that this difference can be used to ascertain many properties of a simulation, and thus has direct and practical significance. Non-dimensional internal consistency norms were constructed based on the difference in these two volumes. These can be used to operationally measure the non-dimensional

magnitude of the truncation error of a calculation by placing the grid vectors from which the corner force is calculated at the fully advanced time level on the corrector step. The accuracy in both space and time can also be measured, and results were found to correspond to the first or second order accuracy that one expects in space or time with the type of finite-volume differencing employed with Lagrangian schemes. It was also shown that some care must be exercised in the use and interpretation of these norms/estimators. More importantly, the difference in these two definitions of zone volume, which can be interpreted as a form of entropy error, was demonstrated to be significant only when numerical instability occurs, or other numerical difficulties, such as a loss of lower dimensional symmetry, appear. A precipitous increase in this quantity during any calculation is a sensitive indication of the development of numerical difficulties. Otherwise this difference remains at or far below truncation error levels. A numerical stability analysis of the two level predictor/corrector time integration scheme employed was also performed: first with a simplified set of model equations using standard Fourier stability analysis, and then using a properly constructed test problem that verifies this analysis for the actual system of equations in all three spatial dimensions. This stability analysis showed that our compatible system of equations with predictor–corrector time integration is stable for CFL number $r \leqslant 1/\sqrt{2\alpha}$ and $\alpha \geqslant 0.5$, where α is the time centering parameter of the pressure on the corrector step; otherwise, it is unconditionally unstable.

The principal conclusion of this study is that the corner force, which is the central feature of this numerical scheme, should be placed at the time-centered $n+\frac{1}{2}$ time level on the corrector step of a cycle. This centering should be performed with respect to both the grid vectors, to minimize the difference in the two zone volumes used in this formulation, and with respect to the total stress (pressure plus deviators, and artificial stresses such as viscosity and anti-hourglass forces). Corner force time-centering results in the least amount of numerical dissipation and the largest stability margin. However, this is also seen to be just at the stability boundary, and is thus a somewhat different choice than is often made with numerical schemes (particularly ODE solvers) where the timestep is chosen from within the stability region.

Acknowledgments

We wish to acknowledge useful discussions with B. Wendroff concerning the stability analysis of the predictor/corrector time integration scheme: specifically, to have correctly conjectured the $1/\sqrt{2\alpha}$ stability bound. This work was supported by the Advanced Strategic Computing Initiative (ASCI) of the US Department of Energy.

References

[1] H.A. Bethe, R. Christy, R.R. Davis, R.P. Feynman, S. Frankel, E.J. Konopinski, N. Livesay, N. Metropolis, E. Nelson, IBM calculations of implosion hydrodynamics, Los Alamos National Laboratory report, LA-94, June 20, 1944, unclassified, cover only.
[2] R.D. Richtmyer, K.W. Morton, Difference Methods for Initial-Value Problems, second ed., Interscience, New York, 1967 (Chapter 12).
[3] B. Alder, S. Fernbach, M. Rotenberg (Eds.), Methods in Computational Physics, vol. 3, 1964, various articles.
[4] J. VonNeumann, R.D. Richtmyer, A method for the calculation of hydrodynamic shocks, J. Appl. Phys. 21 (1950) 232–237.
[5] W.B. Goad, WAT: a numerical method for two-dimensional unsteady fluid flow, Los Alamos National Laboratory Report, LAMS-2365, 1960.
[6] E.J. Caramana, P.P. Whalen, Numerical preservation of symmetry properties of continuum problems, J. Comput. Phys. 141 (1998) 174–198.
[7] A.P. Favorskii, Variational-discrete models of hydrodynamics equations, Diff. Equations 16 (1980) 1308–1321.
[8] L.G. Margolin, T.F. Adams, Spatial differencing for finite difference codes, Los Alamos National Laboratory Report, LA-10249, 1985.
[9] A.A. Samarskii, V.F. Tishkin, A.P. Favorskii, M.J. Shashkov, Operational finite difference schemes, Diff. Equations 17 (1981) 854–862.
[10] L.G. Margolin, A.E. Tarwater, A diffusion operator for Lagrangian meshes, Lawrence Livermore National Laboratory Report, UCRL-95652, 1986.
[11] D.E. Burton, Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity, Advances in the Free-Lagrange Method, Springer, New York, 1990.
[12] D.E. Burton, Multidimensional discretizations of conservation laws for unstructured polyhedral grids, Lawrence Livermore National Laboratory Report, UCRL-JC-118306, 1994.

[13] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1998) 227–262.
[14] R. Loubère, E.J. Caramana, The force/work differencing of exceptional points in the discrete, compatible formulation of Lagrangian hydrodynamics, LAUR-04-8906, J. Comput. Phys., in press, doi:10.1016/j.jcp.2005.11.022.
[15] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (1998) 70–97.
[16] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distortion and Hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comput. Phys. 142 (1998) 521–561.
[17] E.J. Caramana, R. Loubère, Curl-q: a vorticity damping artificial viscosity for Lagrangian hydrodynamics calculations, LAUR-05-6618, J. Comput. Phys. 215 (2006) 385–391.
[18] E.J. Caramana, C.L. Rousculp, D.E. Burton, A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry, J. Comput. Phys. 157 (2000) 89–119.

1.3.2 Stability (again)

A non-classical stability bound has been uprised in [12]. However we were not able to rigorously prove it and we had to resort to numerical sampling only at the very end of the proof to validate this stability bound cleverly conjectured by B. Wendroff. This was unfortunate. Consequently with the help of B. Wendroff we attacked this proof again in paper [15] entitled *On stabiliy analysis of staggered schemes*.
Some years ago, M. Shashkov, proposed a simple problem for the purpose of testing the stability of Lagrangian hydrodynamics codes. The initial data are given as zero velocity, constant pressure and constant density. One then computes the total kinetic energy, which should be zero for all time. We have discovered in [12] that it was unstable for certain mesh ratios that "folklore" indicated should be otherwise — the instability manifested itself as an explosive growth of kinetic energy. The surprise here is not that a 1D symmetry preserving 2D code could be unstable for 1D data; the surprise lies in the contradiction of the "folklore". Briefly, the compatible staggered Lagrangian scheme has two parameters, α and β[7] such that if $\alpha = \beta = 1/2$, the scheme resembles a time centered (Crank-Nicolson) scheme, while if $\alpha = \beta = 1$, it looks like backward Euler. For the Courant-Friedrichs-Lewy (CFL) condition CFL $= 1$, the scheme is stable for $\alpha = \beta = 1/2$. However, for $\alpha = 1$, $\beta = 1/2$ the scheme is unstable for CFL > 0.71, indicating that as α increases, the CFL limit decreases.

In [12] we have created a 1D problem using the same data; we seed a 1D staggered-grid predictor-corrector compatible Hydrodynamics Lagrangian code with a small random perturbation of the pressure. Moreover, we have created the multi-dimension version of the problem and have run our 2D and Caramana's 3D code as well. We observed the same phenomenon in 1D, 2D, and 3D [12]. The stability limit was conjectured by B. Wendroff to be CFL $= 1/(2\sqrt{\alpha\beta})$, and numerical tests in [12] using the full nonlinear equations show that this is true provided $\alpha \geq 1/2$ and $\beta \geq 1/2$. In this paper we prove the above conjecture in 2D, which contains 1D as a special case. Of course, there is no possibility of doing this for the full nonlinear problem - the Euler equations. Ultimately, one applies a von Neumann analysis to the linearized system. In [12] we almost succeeded in doing this theoretical analysis in 1D. In this paper we have succeeded in 2D, by using the numerical radius of the amplification matrix as a tool, an idea apparently first applied in [101]. This proof is the goal of this paper.
We showed on wave equations as a model equation that the staggered implicit scheme is unconditionally stable for $\alpha \geq 1/2$, $\beta \geq 1/2$; moreover we showed that the predictor-corrector staggered scheme is stable for $\alpha \geq 1/2$, $\beta \geq 1/2$ and CFL $\leq 1/(2\sqrt{\alpha\beta})$. In a specific paragraph we demonstrated on a specific 1D example that the schemes for $\alpha < 1/2$ or $\beta < 1/2$ can not be stable as some Fourier components are amplified. Finally, we have showed 2D numerical results for the wave and Euler equations using a compatible Lagrangian Hydrodynamics code. In Fig. 1.6 we reproduce the graphics from [15] showing the numerical experiments made with the wave equations (panel **(a)**) and the hydrodynamics equation (panel **(b)**). The exact and experimental maximal CFL number as a function of α, β and only α are in perfect agreement.

This paper is reproduced in the following pages.

7. One parameter for each 'entity location', that is to say α for cell-centered variables, β for node-centered ones.

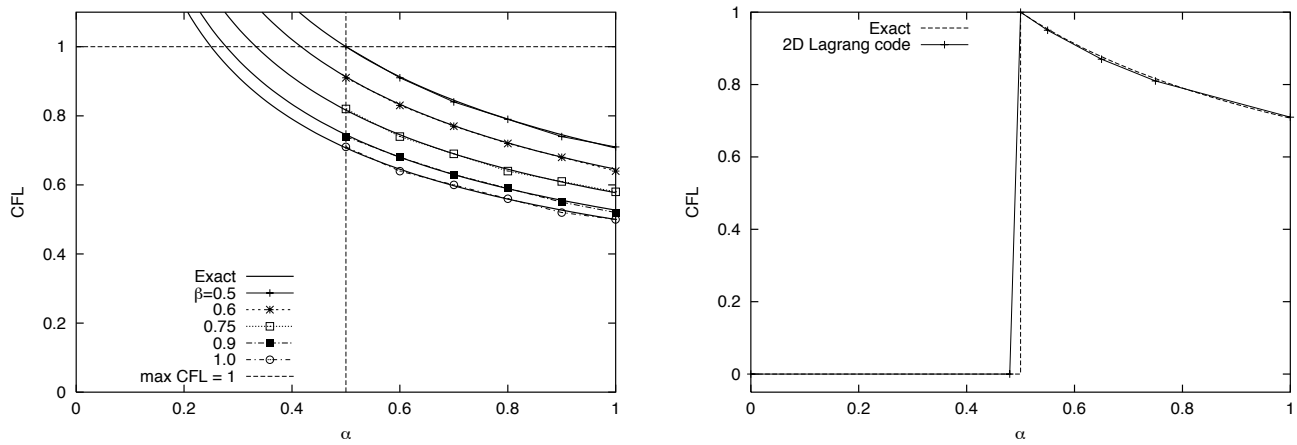


Figure 1.6 – *Numerical results from paper [15]. Maximum CFL number λ as a function of α (β fixed) for the kinetic energy to remain on the order of machine precision for the stability test case of M.Shashkov —* ***(a)*** *Wave equations. The vertical dashed line is the $\alpha = 1/2$ limit, the horizontal one is the CFL $= 1$ limit. The Theorem predicts the continuous thick lines, the code produces the data for different β. Any scheme defined by a value $\alpha \geq 1/2$ and $\beta \geq 1/2$ is stable with the following CFL number $\lambda \leq 1/(2\sqrt{\alpha\beta})$ —* ***(b)*** *Hydrodynamics equations for $\beta = 1/2$. The therorem predicts the dashed line ; the 2D code produces the continuous line. Any scheme defined by a value $\alpha \geq 1/2$ is stable with the CFL number $\lambda \leq 1/\sqrt{2\alpha}$ and is unstable otherwise.*

SIAM J. NUMER. ANAL.
Vol. 46, No. 2, pp. 996–1011
© 2008 Society for Industrial and Applied Mathematics

ON STABILITY OF STAGGERED SCHEMES*

AMY L. BAUER†, RAPHAËL LOUBÈRE‡, AND BURTON WENDROFF§

Abstract. This paper investigates the theoretical stability bound of a Lagrangian staggered scheme used to solve hydrodynamics equations. We present the two-dimensional (2D) wave equation as a possible model for this study and, by using the numerical radius of the amplification matrix, we prove that the family of schemes defined with two time-centering parameters is limited by a non-classical stability bound limit defined with an analytical curve. We further show that 2D numerical experiments agree with this theoretical result.

Key words. staggered schemes, stability, wave equations, Euler equations

AMS subject classifications. 65N12, 65N22, 65M12, 65J10, 65J15

DOI. 10.1137/060660151

1. Introduction. The concept of stability was first introduced in the seminal paper [1] of Courant, Friedrichs, and Lewy in 1928 where they discussed finite-difference methods for solving partial differential equations. Amazingly in the very same paper they studied one of the first staggered numerical schemes for the one-dimensional (1D) wave equation $v_t = cw_x, w_t = cv_x$, which was further developed in 1967 by Richtmyer and Morton in [6] and cast in its "modern" form as:

$$\frac{v_j^{n+1} - v_j^n}{\Delta t} = c\,\frac{w_{j+1/2}^n - w_{j-1/2}^n}{\Delta x}, \quad \frac{w_{j-1/2}^{n+1} - w_{j-1/2}^n}{\Delta t} = c\,\frac{v_j^{n+1} - v_{j-1}^{n+1}}{\Delta x}.$$

By considering these works as the genesis for numerical analysis and computational fluid dynamics, one would expect the stability of staggered numerical schemes to be at present well understood. However, this is far from being the case. Although staggered schemes have since been widely employed computationally, not much is known analytically about the stability of staggered schemes.

The use of staggered schemes also dates back to the inception of Los Alamos National Laboratory where the "calculation of certain time-dependent fluid flows played an important part in the wartime work of the laboratory" (preface to the first edition of [6]). Staggered schemes have frequently been used to solve the compressible fluid dynamics equations in their Lagrangian formulation. In this case, a staggered spatial placement of variables is used, where the position and velocity are defined at grid points and density, internal energy, and the pressure are defined at cell centers. Since this time, other Lagrangian hydrodynamics staggered numerical codes have been developed around the globe, but again little advancement has been made towards a deeper understanding and analytical proof of the stability of such schemes.

Some years ago, Shashkov proposed a simple problem for the purpose of testing the stability of Lagrangian hydrodynamics codes. The initial data are given as zero velocity, constant pressure, and constant density. One then computes the total kinetic energy, which should be zero for all time. Caramana recently tried this problem with his two-dimensional (2D) staggered grid Lagrangian code and discovered that it was unstable for certain mesh ratios that "folklore" indicated should be otherwise—the instability manifested itself as an explosive growth of kinetic energy. The surprise here is not that a 1D symmetry-preserving 2D code could be unstable for 1D data; the surprise lies in the contradiction of the folklore.

The precise numerical scheme we are referring to and which is the focus of our investigation is the staggered-grid predictor-corrector compatible Lagrangian hydrodynamics scheme. We refer the reader to [3], [4] for the details of this scheme which we omit here. Briefly, there are two parameters α and β such that, if $\alpha = \beta = 1/2$, the scheme resembles a time-centered (Crank–Nicolson) scheme, while if $\alpha = \beta = 1$, it looks like backward Euler. For the Courant–Friedrichs–Lewy (CFL) condition CFL $= 1$, the scheme is stable for $\alpha = \beta = 1/2$. However, for $\alpha = 1$, $\beta = 1/2$ the scheme is unstable for CFL > 0.71, indicating that, as α increases, the CFL limit decreases. Bauer et al. then created a 1D problem by using the same data; they seed their 1D staggered-grid predictor-corrector compatible hydrodynamics Lagrangian code with a small random perturbation of the pressure. Moreover, they created the multidimension version of the problem and ran their 2D and Caramana's 3D code as well. They observed the same phenomenon in 1D, 2D, and 3D [3]. The stability limit was conjectured to be CFL $= 1/(2\sqrt{\alpha\beta})$, and numerical tests using the full nonlinear equations show that this is true provided $\alpha \geq 1/2$ and $\beta \geq 1/2$ [3].

We prove the above conjecture in 2D, which contains 1D as a special case. Of course, there is no possibility of doing this for the full nonlinear problem—the Euler equations. Ultimately, one applies a von Neumann analysis to the linearized system. Bauer et al. [3] almost succeeded in doing this theoretical analysis in 1D, resorting to numerical sampling only at the very end of the proof. We have succeeded in 2D, by using the numerical radius of the amplification matrix as a tool, an idea apparently first applied in [2]. This proof is the goal of this paper, which is articulated as follows: First we present the Lagrangian coordinates and justify the use of the wave equation as a possible linear model for the Euler equations in 1D and 2D; second, we introduce the notation; third, we show that the staggered implicit scheme is unconditionally stable for $\alpha \geq 1/2$, $\beta \geq 1/2$; fourth, we show that the predictor-corrector staggered scheme is stable for $\alpha \geq 1/2$, $\beta \geq 1/2$, and CFL $\leq 1/(2\sqrt{\alpha\beta})$. In a paragraph we show on a specific 1D example that the schemes for $\alpha < 1/2$ or $\beta < 1/2$ cannot be stable as some Fourier components are amplified. Finally, we show 2D numerical results for the wave and Euler equations using a compatible Lagrangian hydrodynamics code.

2. Lagrangian coordinates and wave equation models.

2.1. 1D. The hydrodynamic equations in 1D Lagrangian coordinates, expressed in mass variables m, are

$$\frac{du}{dt} = -\frac{\partial p}{\partial m}, \qquad \frac{d\tau}{dt} = \frac{\partial u}{\partial m}, \qquad \frac{de}{dt} = -p\frac{\partial u}{\partial m},$$

*Received by the editors May 17, 2006; accepted for publication (in revised form) October 11, 2007; published electronically March 5, 2008. This work was performed under the auspices of the U.S. Department of Energy at Los Alamos National Laboratory, under contract W-7405-ENG-36. The authors acknowledge the partial support of the DOE/ASCR Program in the Applied Mathematical Sciences and the Laboratory Directed Research and Development program (LDRD). The authors also acknowledge the partial support of DOE's Accelerated Strategic Computing Initiative (ASCI).
http://www.siam.org/journals/sinum/46-2/66015.html
†Theoretical Division, T-14, Los Alamos National Laboratory MS-B262, Los Alamos, NM 87545 (albauer@lanl.gov).
‡Institut de Mathematiques et CNRS (UMR-5219), Université Paul-Sabatier Toulouse 3, 31062 Toulouse Cedex 9, France (loubere@mip.ups-tlse.fr).
§Theoretical Division, T-7, Los Alamos National Laboratory MS-B284, Los Alamos, NM 87545 (bbw@lanl.gov).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where u is the velocity, τ is the specific volume, e is the specific internal energy, and $p = p(\tau, e)$ is the pressure.

Total energy E and entropy S are conserved:

$$\begin{aligned}\frac{dE}{dt} &= \frac{1}{2}\frac{du^2}{dt} + \frac{de}{dt}\\ &= -u\frac{\partial p}{\partial m} - p\frac{\partial u}{\partial m} = -\frac{\partial pu}{\partial m},\end{aligned}$$

and

$$T\frac{dS}{dt} = \frac{de}{dt} + p\frac{d\tau}{dt} = 0.$$

Therefore for smooth flows the exact differential equations for u, p are

$$\begin{aligned}&\frac{du}{dt} = -\frac{\partial p}{\partial m},\\ &\frac{dp}{dt} = \frac{dp}{d\tau}\frac{d\tau}{dt} + \frac{dp}{de}\frac{de}{dt} = \left(\frac{dp}{d\tau} - p\frac{dp}{de}\right)\frac{\partial u}{\partial m},\end{aligned}$$

and taking the coefficients p, $\frac{dp}{d\tau}$, and $\frac{dp}{de}$ to be constant yields a linear system with constant coefficients. Now assume that $\frac{dp}{d\tau} < 0$ and $\frac{dp}{de} > 0$. As

$$c = \sqrt{\left|\frac{dp}{d\tau}\right| + p\frac{dp}{de}}$$

is the sound speed, the linearization of the system with constant coefficients is obtained by freezing the sound speed; that is, c is assumed to be constant in space and time. Consequently, the resulting system is just the wave equation

$$\begin{aligned}&\frac{du}{dt} = -\frac{\partial p}{\partial m},\\ &\frac{dp}{dt} = -c^2\frac{\partial u}{\partial m}.\end{aligned}$$

By using the new variables $u = u'/c$ and $dm/c = -da$ and dropping the prime notation, this linearized system can be written as

$$\frac{du}{dt} = \frac{\partial p}{\partial a}, \tag{2.1}$$

$$\frac{dp}{dt} = \frac{\partial u}{\partial a}. \tag{2.2}$$

2.2. 2D. The same linearized reduction can be performed for Lagrangian coordinates in 2D, but it is not quite as straightforward. In 2D the Euler equation system

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

is

$$\begin{aligned}\rho\frac{du}{dt} &= -\frac{\partial p}{\partial x},\\ \rho\frac{dv}{dt} &= -\frac{\partial p}{\partial y},\\ \rho\frac{d\tau}{dt} &= \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y},\\ \rho\frac{de}{dt} &= -p\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right),\\ \frac{dx}{dt} &= u,\\ \frac{dy}{dt} &= v.\end{aligned}$$

By following [7], [8] we make the change of variables:

$$\begin{aligned}x &= x(a, b, t),\\ y &= y(a, b, t),\end{aligned}$$

with the choice

$$\begin{aligned}a &= x(a, b, 0),\\ b &= y(a, b, 0).\end{aligned}$$

The Jacobian matrix of the transformation $(x, y) \leftrightarrow (a, b)$ is given by

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial a} & \frac{\partial x}{\partial b} \\ \frac{\partial y}{\partial a} & \frac{\partial y}{\partial b} \end{pmatrix},$$

the determinant of which is the Jacobian

$$J = \frac{\partial x}{\partial a}\frac{\partial y}{\partial b} - \frac{\partial y}{\partial a}\frac{\partial x}{\partial b}.$$

For the two velocity equations we now have

$$\begin{aligned}J\rho\frac{du}{dt} &= \frac{\partial y}{\partial a}\frac{\partial p}{\partial b} - \frac{\partial y}{\partial b}\frac{\partial p}{\partial a},\\ J\rho\frac{dv}{dt} &= \frac{\partial x}{\partial b}\frac{\partial p}{\partial a} - \frac{\partial x}{\partial a}\frac{\partial p}{\partial b}.\end{aligned}$$

But then, by starting the linearization by freezing the entries of the Jacobian, we obtain

$$\begin{aligned}J\rho\frac{d}{dt}\left(u\frac{\partial x}{\partial a} + v\frac{\partial y}{\partial a}\right) &= -J\frac{\partial p}{\partial a},\\ J\rho\frac{d}{dt}\left(u\frac{\partial x}{\partial b} + v\frac{\partial y}{\partial b}\right) &= -J\frac{\partial p}{\partial b}.\end{aligned}$$

Thus, by letting

$$\begin{aligned}u' &= u\frac{\partial x}{\partial a} + v\frac{\partial y}{\partial a},\\ v' &= u\frac{\partial x}{\partial b} + v\frac{\partial y}{\partial b},\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

we obtain

$$\rho\frac{du'}{dt} = -\frac{\partial p}{\partial a},$$
$$\rho\frac{dv'}{dt} = -\frac{\partial p}{\partial b}.$$

We next need to evaluate $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ in the energy equation. First,

$$Ju = \frac{\partial y}{\partial b}u' - \frac{\partial y}{\partial a}v',$$
$$-Jv = \frac{\partial x}{\partial b}u' - \frac{\partial x}{\partial a}v'.$$

Then taking the derivative of u and v yields

$$J^2\frac{\partial u}{\partial x} = \frac{\partial y}{\partial b}\left(\frac{\partial y}{\partial b}\frac{\partial u'}{\partial a} - \frac{\partial y}{\partial a}\frac{\partial u'}{\partial b}\right) - \frac{\partial y}{\partial a}\left(\frac{\partial y}{\partial b}\frac{\partial v'}{\partial a} - \frac{\partial y}{\partial a}\frac{\partial v'}{\partial b}\right),$$
$$J^2\frac{\partial v}{\partial y} = \frac{\partial x}{\partial b}\left(\frac{\partial x}{\partial b}\frac{\partial u'}{\partial a} - \frac{\partial x}{\partial a}\frac{\partial u'}{\partial b}\right) - \frac{\partial x}{\partial a}\left(\frac{\partial x}{\partial b}\frac{\partial v'}{\partial a} - \frac{\partial x}{\partial a}\frac{\partial v'}{\partial b}\right).$$

Summing the previous equations gives

$$J^2\frac{\partial u}{\partial x} + J^2\frac{\partial v}{\partial y} = \left(\left(\frac{\partial x}{\partial b}\right)^2 + \left(\frac{\partial y}{\partial b}\right)^2\right)\frac{\partial u'}{\partial a} - \left(\frac{\partial x}{\partial a}\frac{\partial x}{\partial b} + \frac{\partial y}{\partial a}\frac{\partial y}{\partial b}\right)\frac{\partial u'}{\partial b}$$
$$- \left(\frac{\partial x}{\partial a}\frac{\partial x}{\partial b} + \frac{\partial y}{\partial a}\frac{\partial y}{\partial b}\right)\frac{\partial v'}{\partial a} + \left(\left(\frac{\partial x}{\partial a}\right)^2 + \left(\frac{\partial y}{\partial a}\right)^2\right)\frac{\partial v'}{\partial b},$$

and in order to cancel the cross terms, we are forced to assume here that

$$\frac{\partial x}{\partial a}\frac{\partial x}{\partial b} + \frac{\partial y}{\partial a}\frac{\partial y}{\partial b} = 0, \tag{2.3}$$

so that

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = d_1\frac{\partial u'}{\partial a} + d_2\frac{\partial v'}{\partial b},$$
$$d_1 = \frac{1}{J^2}\left(\left(\frac{\partial x}{\partial b}\right)^2 + \left(\frac{\partial y}{\partial b}\right)^2\right),$$
$$d_2 = \frac{1}{J^2}\left(\left(\frac{\partial x}{\partial a}\right)^2 + \left(\frac{\partial y}{\partial a}\right)^2\right).$$

So

$$\rho\frac{du'}{dt} = -\frac{\partial p}{\partial a},$$
$$\rho\frac{dv'}{dt} = -\frac{\partial p}{\partial b},$$
$$\rho\frac{d\tau}{dt} = d_1\frac{\partial u'}{\partial a} + d_2\frac{\partial v'}{\partial b},$$
$$\rho\frac{de}{dt} = -p\left(d_1\frac{\partial u'}{\partial a} + d_2\frac{\partial v'}{\partial b}\right).$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Now ρ can be scaled into a, b, and as before, a simple substitution yields

$$\frac{du'}{dt} = -\frac{\partial p}{\partial a},$$
$$\frac{dv'}{dt} = -\frac{\partial p}{\partial b},$$
$$\frac{dp}{dt} = -c^2\left(d_1\frac{\partial u'}{\partial a} + d_2\frac{\partial v'}{\partial b}\right).$$

Finally, returning to (u, v, a, b), set

$$u = -c\sqrt{d_1}u', \qquad v = -c\sqrt{d_2}v',$$
$$a = c\sqrt{d_1}\,\widetilde{a}, \qquad b = c\sqrt{d_2}\,\widetilde{b},$$

to obtain the 2D wave equation as a model:

$$\frac{du}{dt} = \frac{\partial p}{\partial \widetilde{a}}, \qquad \frac{dv}{dt} = \frac{\partial p}{\partial \widetilde{b}}, \qquad \frac{dp}{dt} = \frac{\partial u}{\partial \widetilde{a}} + \frac{\partial v}{\partial \widetilde{b}}. \tag{2.4}$$

However, this formulation critically depends on (2.3), so ultimately we present this as only a potential model.[1]

3. Notation. This section defines the notation that will be used throughout the rest of this paper.

The standard inner product on a complex vector space of dimension N is $\langle \mathbf{f}, \mathbf{g}\rangle = \sum_{i=1}^{N} \bar{f}_i\, \bar{g}_i$ for two complex vectors $\mathbf{f} = (f_i)_{i=1,\ldots,N}$ and $\mathbf{g} = (g_i)_{i=1,\ldots,N}$.

Time. We will assume equal time steps; that is, the temporal interval $[0, T > 0]$ is discretized into equal intervals $[t_n, t_{n+1}]$, with $t_{n+1} = t_n + \Delta t$.

Space. In 2D we define uniform rectangles with vertices $x_{i,j}$, $y_{i,j}$, where $x_{i+1,j} = x_{i,j} + \Delta x$ and $y_{i,j+1} = y_{i,j} + \Delta y$. Nodal quantities are indexed by (i, j), while cell quantities are indexed by $(i + \frac{1}{2}, j + \frac{1}{2})$.

Discretization. The ratio of time step to space step in each dimension is given by $\lambda_x = \frac{\Delta t}{\Delta x}$, $\lambda_y = \frac{\Delta t}{\Delta y}$.

Hence, any quantity P at some point (i, j) at time t_n is represented as $P^n_{i,j}$. Similarly, any cell-based quantity C at time t_{n+1} is represented as $C^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}$.

For any variable w defined at two time levels $t_{n+1} > t_n$ on a point or in a cell, we define its interpolated value at an intermediate time $n + \kappa$ as:

$$w^{n+\kappa} = \kappa w^{n+1} + (1 - \kappa)\, w^n, \quad 0 \le \kappa \le 1.$$

We further define a vector $\mathbf{w} = (\mathbf{u}, \mathbf{v}, \mathbf{p})^t$ and $\mathbf{w}^{\alpha,\beta} = (\mathbf{u}^{n+\alpha}, \mathbf{v}^{n+\alpha}, \mathbf{p}^{n+\beta})^t$, where

$$\mathbf{u} = \{u_{i,j} : \ \forall\,(i,j)\}, \quad \mathbf{v} = \{v_{i,j} : \ \forall\,(i,j)\}, \quad \text{and} \quad \mathbf{p} = \left\{p_{i+\frac{1}{2},j+\frac{1}{2}} : \ \forall\,(i,j)\right\}.$$

[1] The Jacobian matrix $\mathbf{J}$ contains information relating to the volume, shape, and orientation of an element after the transformation $(a, b) \to (x, y)$. $\mathbf{\Lambda} = \mathbf{J}^{\mathbf{T}}\mathbf{J}$ is the associated metric tensor:

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_{11} & \lambda_{21} \\ \lambda_{12} & \lambda_{22} \end{pmatrix} = \begin{pmatrix} \left(\frac{\partial x}{\partial a}\right)^2 + \left(\frac{\partial x}{\partial b}\right)^2 & \frac{\partial x}{\partial a}\frac{\partial x}{\partial b} + \frac{\partial y}{\partial a}\frac{\partial y}{\partial b} \\ \frac{\partial x}{\partial a}\frac{\partial x}{\partial b} + \frac{\partial y}{\partial a}\frac{\partial y}{\partial b} & \left(\frac{\partial y}{\partial a}\right)^2 + \left(\frac{\partial y}{\partial b}\right)^2 \end{pmatrix},$$

$\mathbf{\Lambda}$ is a symmetric matrix, and moreover the constraint (2.3) states that $\lambda_{12} = 0$: The metric tensor $\mathbf{\Lambda}$ is diagonal, meaning that the implied system of coordinates is orthogonal.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Boundary conditions. We assume throughout that all sums exist and that Fourier transforms can be taken. Also, in the usual way, it is supposed that effectively there are no boundaries and therefore that indices can be shifted in a sum without changing its value. For example, for all integers k

$$\sum_i u_{i,j}\; p_{i+\frac{1}{2},j} = \sum_i u_{i+k,j}\; p_{i+\frac{1}{2}+k,j}.$$

Operators. There are two sets of data: nodal quantities and cell quantities. Consider the operator Q_x that maps cell data to nodal data given by

$$(Q_x p)_{i,j} = \frac{1}{2}\left(p_{i+\frac{1}{2},j+\frac{1}{2}} + p_{i+\frac{1}{2},j-\frac{1}{2}} - p_{i-\frac{1}{2},j+\frac{1}{2}} - p_{i-\frac{1}{2},j-\frac{1}{2}}\right).$$

By using the usual scalar product in both nodal and cell spaces, the adjoint Q^*_x maps nodal data to cell data and is defined by

$$\langle Q^*_x u, p\rangle = \langle u, Q_x p\rangle = \frac{1}{2}\sum u_{i,j}\left(p_{i+\frac{1}{2},j+\frac{1}{2}} + p_{i+\frac{1}{2},j-\frac{1}{2}} - p_{i-\frac{1}{2},j+\frac{1}{2}} - p_{i-\frac{1}{2},j-\frac{1}{2}}\right)$$
$$= \frac{1}{2}\sum p_{i+\frac{1}{2},j+\frac{1}{2}}\Big(u_{i,j} + u_{i,j+1} - u_{i+1,j} - u_{i+1,j+1}\Big);$$

that is,

$$(Q^*_x u)_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\Big(u_{i,j} + u_{i,j+1} - u_{i+1,j} - u_{i+1,j+1}\Big).$$

Similar definitions exist for Q_y and Q^*_y.

The notation for a staggered scheme on a rectangular grid for the 2D wave equation is depicted in Figure 3.1, where p is defined at the center of each cell and u and v are defined at nodes. For this scheme p has fractional indices, whereas u and v have integer indices. Moreover, one defines interpolated values at each midedge point; for example, for a midedge point defined with indices $i + \frac{1}{2}, j + 1$ one has $p_{i+\frac{1}{2},j+1} = \frac{1}{2}(p_{i+\frac{1}{2},j+\frac{3}{2}} + p_{i+\frac{1}{2},j-\frac{1}{2}})$ and $u_{i+\frac{1}{2},j+1} = \frac{1}{2}(u_{i,j+1} + u_{i+1,j+1})$, whereas for a midedge point $i, j + \frac{1}{2}$ one has $p_{i,j+\frac{1}{2}} = \frac{1}{2}(p_{i+\frac{1}{2},j+\frac{1}{2}} + p_{i-\frac{1}{2},j+\frac{1}{2}})$ and $u_{i,j+\frac{1}{2}} = \frac{1}{2}(u_{i,j+1} + u_{i+1,j})$.

By using this notation, a fully implicit staggered scheme applied to system (2.4) can be written as

$$u^{n+1}_{i,j} = u^n_{i,j} + \lambda_x\left(p^{n+\alpha}_{i+\frac{1}{2},j} - p^{n+\alpha}_{i-\frac{1}{2},j}\right),$$
$$v^{n+1}_{i,j} = v^n_{i,j} + \lambda_y\left(p^{n+\alpha}_{i,j+\frac{1}{2}} - p^{n+\alpha}_{i,j-\frac{1}{2}}\right), \tag{3.1}$$
$$p^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = p^n_{i+\frac{1}{2},j+\frac{1}{2}} + \lambda_x\left(u^{n+\beta}_{i+1,j+\frac{1}{2}} - u^{n+\beta}_{i,j+\frac{1}{2}}\right) + \lambda_y\left(v^{n+\beta}_{i+\frac{1}{2},j+1} - v^{n+\beta}_{i+\frac{1}{2},j}\right).$$

The difference operator is given by the matrix

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & Q_x \\ 0 & 0 & Q_y \\ -Q^*_x & -Q^*_y & 0 \end{pmatrix},$$

together with

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_x & 0 & 0 \\ 0 & \lambda_y & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

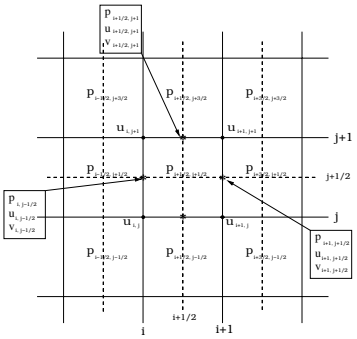


FIG. 3.1. *Rectangular staggered scheme. p is defined at cell centers with fractional indices: $p_{i+\frac{1}{2},j+\frac{1}{2}}$, whereas u, v are defined at nodes with integer indices: $u_{i,j}$. Moreover u, v, p are interpolated at each midedge point $*$. For example, $p_{i+\frac{1}{2},j+1} = \frac{1}{2}(p_{i+\frac{1}{2},j+\frac{3}{2}} + p_{i+\frac{1}{2},j-\frac{1}{2}})$ and $u_{i+\frac{1}{2},j+1} = \frac{1}{2}(u_{i,j+1} + u_{i+1,j+1})$, and similarly for v.*

Thus, the implicit difference scheme (3.1) has the form

$$\mathbf{w}^{n+1} = \mathbf{w}^n + \mathbf{\Lambda}\mathbf{M}\mathbf{\Lambda}\mathbf{w}^{\alpha,\beta}.$$

THEOREM 3.1. *The staggered implicit scheme is stable for any λ_x, λ_y if $\alpha \ge \frac{1}{2}$ and $\beta \ge \frac{1}{2}$.*

Proof. Throughout the proofs, all sums are taken over both indices i and j. Applying the energy method yields

$$H(\alpha,\beta) = \sum\left(u^{n+1}_{i,j} - u^n_{i,j}\right)u^{n+\beta}_{i,j} + \sum\left(v^{n+1}_{i,j} - v^n_{i,j}\right)v^{n+\beta}_{i,j}$$
$$+ \sum\left(p^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - p^n_{i+\frac{1}{2},j+\frac{1}{2}}\right)p^{n+\alpha}_{i+\frac{1}{2},j+\frac{1}{2}} = 0.$$

This follows from the fact that

$$\mathbf{M}^* = -\mathbf{M}$$

and therefore for real $\mathbf{w}$

$$\langle \mathbf{M}\mathbf{w}, \mathbf{w}\rangle = 0\,.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Now let us define $U(\beta), V(\beta), P(\alpha)$ such that $H(\alpha,\beta) = U(\beta) + V(\beta) + P(\alpha)$ and

$$\begin{aligned}
U(\beta) &= \sum \left(u_{i,j}^{n+1} - u_{i,j}^{n}\right) u_{i,j}^{n+\beta} \\
&= \beta \sum \left(u_{i,j}^{n+1}\right)^2 - (1-\beta) \sum \left(u_{i,j}^{n}\right)^2 - (2\beta - 1) \sum \left(u_{i,j}^{n+1} u_{i,j}^{n}\right), \\
V(\beta) &= \sum \left(v_{i,j}^{n+1} - v_{i,j}^{n}\right) v_{i,j}^{n+\beta} \\
&= \beta \sum \left(v_{i,j}^{n+1}\right)^2 - (1-\beta) \sum \left(v_{i,j}^{n}\right)^2 - (2\beta - 1) \sum \left(v_{i,j}^{n+1} v_{i,j}^{n}\right), \\
P(\alpha) &= \sum \left(p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} - p_{i+\frac{1}{2},j+\frac{1}{2}}^{n}\right) p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\alpha} \\
&= \alpha \sum \left(p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1}\right)^2 - (1-\alpha) \sum \left(p_{i+\frac{1}{2},j+\frac{1}{2}}^{n}\right)^2 \\
&\quad - (2\alpha - 1) \sum \left(p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} p_{i+\frac{1}{2},j+\frac{1}{2}}^{n}\right).
\end{aligned}$$

We detail the steps in the proof by using U and comment that the steps in the proof for V and P are similarly obtained.

Differentiating U with respect to β gives

$$U'(\beta) = \sum \left(u_{i,j}^{n+1}\right)^2 + \sum \left(u_{i,j}^{n}\right)^2 - 2 \sum \left(u_{i,j}^{n+1} u_{i,j}^{n}\right).$$

Then by the Schwarz inequality

$$U'(\beta) > 0$$

($U' = 0$ only in the trivial case that the initial data are constant). Since

$$U\left(\frac{1}{2}\right) = \frac{1}{2}\left(\sum \left(u_{i,j}^{n+1}\right)^2 - \sum \left(u_{i,j}^{n}\right)^2\right),$$

it follows that for $\beta \geq \frac{1}{2}$

$$U(\beta) \geq \frac{1}{2}\left(\sum \left(u_{i,j}^{n+1}\right)^2 - \sum \left(u_{i,j}^{n}\right)^2\right),$$

while for $\beta < \frac{1}{2}$

$$U(\beta) < \frac{1}{2}\left(\sum \left(u_{i,j}^{n+1}\right)^2 - \sum \left(u_{i,j}^{n}\right)^2\right).$$

By applying the same reasoning to V and P we see that for $\beta \geq \frac{1}{2}$ and $\alpha \geq \frac{1}{2}$

$$\begin{aligned}
\sum \left(\left(u_{i,j}^{n+1}\right)^2 - \left(u_{i,j}^{n}\right)^2\right) + \sum \left(\left(v_{i,j}^{n+1}\right)^2 - \left(v_{i,j}^{n}\right)^2\right) \\
+ \sum \left(\left(p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1}\right)^2 - \left(p_{i+\frac{1}{2},j+\frac{1}{2}}^{n}\right)^2\right) \leq 0;
\end{aligned}$$

that is, we have stability because

$$\begin{aligned}
\sum \left(u_{i,j}^{n+1}\right)^2 + \sum \left(v_{i,j}^{n+1}\right)^2 + \sum \left(p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1}\right)^2 \\
\leq \sum \left(u_{i,j}^{n}\right)^2 + \sum \left(v_{i,j}^{n}\right)^2 + \sum \left(p_{i+\frac{1}{2},j+\frac{1}{2}}^{n}\right)^2. \qquad \square
\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

4. Rectangular predictor-corrector scheme. The predictor-corrector scheme in 2D [4] for the wave equation can now be described.

Predictor step.

$$\begin{aligned}
\widetilde{u}_{i,j}^{n+1} &= u_{i,j}^{n} + \lambda_x \left(Q_x p^n\right)_{i,j}, \\
\widetilde{v}_{i,j}^{n+1} &= v_{i,j}^{n} + \lambda_y \left(Q_y p^n\right)_{i,j}, \\
\widetilde{p}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} &= p_{i+\frac{1}{2},j+\frac{1}{2}}^{n} - \lambda_x \left(Q_x^* u^{n+\beta}\right)_{i+\frac{1}{2},j+\frac{1}{2}} - \lambda_y \left(Q_y^* v^{n+\beta}\right)_{i+\frac{1}{2},j+\frac{1}{2}};
\end{aligned}$$

recall that $w^{n+\beta} = \beta \widetilde{w}^{\,n+1} + (1-\beta) w^n$ for $w = u, v$.

Corrector step.

$$\begin{aligned}
u_{i,j}^{n+1} &= u_{i,j}^{n} + \lambda_x \left(Q_x p^{n+\alpha}\right)_{i,j}, \\
v_{i,j}^{n+1} &= v_{i,j}^{n} + \lambda_y \left(Q_y p^{n+\alpha}\right)_{i,j}, \\
p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} &= p_{i+\frac{1}{2},j+\frac{1}{2}}^{n} - \lambda_x \left(Q_x^* u^{n+\beta}\right)_{i+\frac{1}{2},j+\frac{1}{2}} - \lambda_y \left(Q_y^* v^{n+\beta}\right)_{i+\frac{1}{2},j+\frac{1}{2}};
\end{aligned}$$

recall that $p^{n+\alpha} = \alpha \widetilde{p}^{\,n+1} + (1-\alpha) p^n$ and now $w^{n+\beta} = \beta w^{n+1} + (1-\beta) w^n$ for $w = u, v$.

By substituting the equation resulting from the predictor step into the corrector step, we obtain the following difference scheme:

$$\mathbf{w}^{n+1} = \mathbf{S}\mathbf{w}^n.$$

Rather than write this out in terms of the difference operators, we immediately move to the Fourier transforms of the variables and operators.

The Fourier transform employs a substitution of variables; for instance, for p we have

$$p_{i+\frac{1}{2},j+\frac{1}{2}}^{n} \longmapsto p_0 e^{\theta(n\Delta t) + \mathrm{i}\left(2\delta\left((i+\frac{1}{2})\Delta x\right) + 2\gamma\left((j+\frac{1}{2})\Delta y\right)\right)},$$

where θ is complex and δ, γ are real. After factoring we obtain:

$$\begin{aligned}
\left(\widehat{Q}_x p\right)_{i,j} &= p_0 e^{\theta(n\Delta t) + \mathrm{i}(2\delta(i\Delta x) + 2\gamma(j\Delta y))} \frac{1}{2}\left(2\mathrm{i}\sin(\delta\Delta x)\right)\left(e^{\mathrm{i}\gamma\Delta y} + e^{-\mathrm{i}\gamma\Delta y}\right) \\
&= p_0 e^{\theta(n\Delta t) + \mathrm{i}(2\delta(i\Delta x) + 2\gamma(j\Delta y))} \left(2\mathrm{i}\sin(\delta\Delta x)\cos(\gamma\Delta y)\right),
\end{aligned}$$

with $\mathrm{i} = \sqrt{-1}$. Then if one denotes $\xi = \delta\Delta x$, and $\eta = \gamma\Delta y$, the dimensionless wave numbers in the x and y directions, respectively, on an uniform mesh, the Fourier transforms of the operator Q_x, Q_y are

$$\left(\widehat{Q}_x p\right)_{i,j} = p_0 e^{\theta(n\Delta t) + 2\mathrm{i}(i\xi + j\eta)} \left(2\mathrm{i}\cos\xi\sin\eta\right), \tag{4.1}$$

$$\left(\widehat{Q}_y p\right)_{i,j} = p_0 e^{\theta(n\Delta t) + 2\mathrm{i}(i\xi + j\eta)} \left(2\mathrm{i}\cos\xi\sin\eta\right). \tag{4.2}$$

The same formulas can be similarly obtained for Q^*.

By setting

$$\Phi_x = 2\lambda_x \sin\xi\cos\eta \quad \text{and} \quad \Phi_y = 2\lambda_y \sin\eta\cos\xi,$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

and dropping the hat notation, we obtain

$$\mathbf{S} = \begin{pmatrix}
1 - \alpha\Phi_x^2 & -\alpha\Phi_x\Phi_y & \mathrm{i}\Phi_x\left(1 - \alpha\beta(\Phi_x^2 + \Phi_y^2)\right) \\
-\alpha\Phi_x\Phi_y & 1 - \alpha\Phi_y^2 & \mathrm{i}\Phi_y\left(1 - \alpha\beta\left(\Phi_x^2 + \Phi_y^2\right)\right) \\
\mathrm{i}\Phi_x\left(1 - \alpha\beta(\Phi_x^2 + \Phi_y^2)\right) & \mathrm{i}\Phi_y\left(1 - \alpha\beta(\Phi_x^2 + \Phi_y^2)\right) & 1 + \alpha\beta^2(\Phi_x^2 + \Phi_y^2)^2 - \beta\left(\Phi_x^2 + \Phi_y^2\right)
\end{pmatrix}.$$

The Lax–Richtmyer stability theory [6] tells us that we must show that the amplification factor $\mathbf{S}$ has uniformly bounded powers. First, observe that $\mathbf{S}\mathbf{S}^* \neq \mathbf{S}^*\mathbf{S}$, so it is not sufficient to show that the eigenvalues of $\mathbf{S}$ are less than 1. We shall show instead that the numerical radius of $\mathbf{S}$ is bounded by 1, since this implies that the powers of $\mathbf{S}$ are uniformly bounded by 2. A survey of results about the numerical radius is given in [5] along with a direct proof that, if the numerical radius is bounded by one, the powers are bounded by 2. We remark that the Kreiss matrix theorem implies a weaker result, namely, that the powers are bounded by a constant that depends on the size of the matrix [6]. The numerical radius was apparently first used as a stability analysis tool in [2], and here we use the basic ideas of that work.

We henceforth use the same symbols for grid functions and their Fourier transforms, realizing that this is an abuse of the notation, but hoping that no confusion will occur. The numerical radius of $\mathbf{S}$, denoted $R(\mathbf{S})$, is

$$R(\mathbf{S}) = \sup_{\mathbf{w}} |\langle \mathbf{S}\mathbf{w}, \mathbf{w}\rangle|, \quad \text{with} \quad \langle \mathbf{w}, \mathbf{w}\rangle = 1.$$

The matrix $\mathbf{S}$ can be split into real and imaginary parts as

$$\mathbf{S} = \mathbf{A} + \mathrm{i}\mathbf{B},$$

where

$$\mathbf{A} = \begin{pmatrix}
1 - \alpha\Phi_x^2 & -\alpha\Phi_x\Phi_y & 0 \\
-\alpha\Phi_x\Phi_y & 1 - \alpha\Phi_y^2 & 0 \\
0 & 0 & 1 + \alpha\beta^2(\Phi_x^2 + \Phi_y^2)^2 - \beta\left(\Phi_x^2 + \Phi_y^2\right)
\end{pmatrix},$$

$$\mathbf{B} = \left(1 - \alpha\beta(\Phi_x^2 + \Phi_y^2)\right) \begin{pmatrix}
0 & 0 & \Phi_x \\
0 & 0 & \Phi_y \\
\Phi_x & \Phi_y & 0
\end{pmatrix}.$$

Now let

$$r = \langle \mathbf{A}\mathbf{w}, \mathbf{w}\rangle, \quad j = \langle \mathbf{B}\mathbf{w}, \mathbf{w}\rangle.$$

Since $\mathbf{A}$ and $\mathbf{B}$ are real and symmetric, r and j are real. Then

$$|\langle \mathbf{S}\mathbf{w}, \mathbf{w}\rangle|^2 = |\langle \mathbf{A}\mathbf{w}, \mathbf{w}\rangle + \mathrm{i}\langle \mathbf{B}\mathbf{w}, \mathbf{w}\rangle|^2 = |r + \mathrm{i}j|^2 = r^2 + j^2.$$

THEOREM 4.1. *The* 2D *staggered rectangular scheme is stable if* $\alpha \geq \frac{1}{2}$, $\beta \geq \frac{1}{2}$, *and* $4\alpha\beta\max(\lambda_x^2, \lambda_y^2) \leq 1$.

Proof. Suppose that $\alpha \geq \frac{1}{2}$, $\beta \geq \frac{1}{2}$, and $4\alpha\beta\max(\lambda_x^2, \lambda_y^2) \leq 1$. Let

$$\overline{\mathrm{x}} = \sqrt{\alpha\beta}\,\Phi_x, \quad \overline{\mathrm{y}} = \sqrt{\alpha\beta}\,\Phi_y.$$

First note that

$$z^2 = \overline{\mathrm{x}}^2 + \overline{\mathrm{y}}^2 \leq 1,$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

since

$$\begin{aligned}
z^2 = \overline{\mathrm{x}}^2 + \overline{\mathrm{y}}^2 &= \alpha\beta\left(\Phi_x^2 + \Phi_y^2\right) \\
&= \alpha\beta\left(4\lambda_x^2 \sin^2\xi\cos^2\eta + 4\lambda_y^2 \sin^2\eta\cos^2\xi\right) \\
&\leq \underbrace{4\alpha\beta\max\left(\lambda_x^2, \lambda_y^2\right)}_{\leq 1}\left[(\cos\eta\sin\xi)^2 + (\cos\xi\sin\eta)^2\right] \\
&\leq (\cos\eta\sin\xi)^2 + (\cos\xi\sin\eta)^2 \leq 1.
\end{aligned}$$

Recall that $|\mathbf{u}|^2 + |\mathbf{v}|^2 + |\mathbf{p}|^2 = 1$, and note that

$$|\overline{\mathrm{x}}u + \overline{\mathrm{y}}v|^2 \leq \left(1 - |p|^2\right)\left(\overline{\mathrm{x}}^2 + \overline{\mathrm{y}}^2\right), \tag{4.3}$$

since

$$\begin{aligned}
|\overline{\mathrm{x}}u + \overline{\mathrm{y}}v|^2 = \left(\overline{\mathrm{x}}u + \overline{\mathrm{y}}v\right)\left(\overline{\mathrm{x}}\overline{u} + \overline{\mathrm{y}}\overline{v}\right) &= \overline{\mathrm{x}}^2 u\overline{u} + \overline{\mathrm{y}}^2 v\overline{v} + \overline{\mathrm{x}}\overline{\mathrm{y}}\left(\overline{u}v + \overline{v}u\right) \\
&\leq \overline{\mathrm{x}}^2 u\overline{u} + \overline{\mathrm{y}}^2 v\overline{v} + \overline{\mathrm{x}}^2|v|^2 + \overline{\mathrm{y}}^2|u|^2 \\
&= \left(|u|^2 + |v|^2\right)\left(\overline{\mathrm{x}}^2 + \overline{\mathrm{y}}^2\right) \\
&\leq \left(1 - |p|^2\right)\left(\overline{\mathrm{x}}^2 + \overline{\mathrm{y}}^2\right).
\end{aligned} \tag{4.4}$$

Also, by setting

$$\overline{\mathrm{a}} = 1/\alpha, \quad \overline{\mathrm{b}} = 1/\beta,$$

we have that $\overline{\mathrm{a}}, \overline{\mathrm{b}}$ are positive and bounded by 2. Then

$$\begin{aligned}
r &= 1 - \overline{\mathrm{b}}|\overline{\mathrm{x}}u + \overline{\mathrm{y}}v|^2 - \overline{\mathrm{a}}\left(\overline{\mathrm{x}}^2 + \overline{\mathrm{y}}^2\right)|p|^2 + \overline{\mathrm{a}}\left(\overline{\mathrm{x}}^2 + \overline{\mathrm{y}}^2\right)^2|p|^2, \\
j &= 2\sqrt{\overline{\mathrm{a}}\overline{\mathrm{b}}}\,\mathcal{R}\left(\overline{p}\left(\overline{\mathrm{x}}u + \overline{\mathrm{y}}v\right)\right)\left(1 - \left(\overline{\mathrm{x}}^2 + \overline{\mathrm{y}}^2\right)\right),
\end{aligned}$$

where $\mathcal{R}(q)$ is the real part of q, and

$$\begin{aligned}
r^2 &= \left(1 - \overline{\mathrm{b}}|\overline{\mathrm{x}}u + \overline{\mathrm{y}}v|^2 - \overline{\mathrm{a}}z^2\left(1 - z^2\right)|p|^2\right)^2, \\
j^2 &\leq 4\overline{\mathrm{a}}\overline{\mathrm{b}}\left(1 - z^2\right)^2|p|^2|\overline{\mathrm{x}}u + \overline{\mathrm{y}}v|^2.
\end{aligned}$$

Now set

$$|\overline{\mathrm{x}}u + \overline{\mathrm{y}}v|^2 = z^2\left(1 - |p|^2\right)\gamma.$$

By (4.3), $0 \leq \gamma \leq 1$. We thus have that

$$\begin{aligned}
r^2 &= \left(1 - \gamma\overline{\mathrm{b}}z^2\left(1 - |p|^2\right) - \overline{\mathrm{a}}z^2\left(1 - z^2\right)|p|^2\right)^2, \\
j^2 &\leq 4\overline{\mathrm{a}}\left(\gamma\overline{\mathrm{b}}\right)z^2\left(1 - z^2\right)^2|p|^2\left(1 - |p|^2\right).
\end{aligned}$$

Also note that, for all $0 \leq p \leq 1$,

$$|p|^2\left(1 - |p|^2\right) \leq \frac{1}{4}.$$

Since $\gamma\overline{\mathrm{b}}$ has the same range as $\overline{\mathrm{b}}$, we can replace $\gamma\overline{\mathrm{b}}$ by $\overline{\mathrm{b}}$. What we need to show then is that

$$g\left(z, p, \overline{\mathrm{a}}, \overline{\mathrm{b}}\right) = \left[1 - \overline{\mathrm{b}}z^2\left(1 - |p|^2\right) - \overline{\mathrm{a}}z^2\left(1 - z^2\right)|p|^2\right]^2 + \overline{\mathrm{a}}\overline{\mathrm{b}}z^2\left(1 - z^2\right)^2 \leq 1$$

for $0 \leq \overline{\mathrm{a}}, \overline{\mathrm{b}} \leq 2$, and $0 \leq z, p \leq 1$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

LEMMA 4.2. *The function* $g(z,p,\overline{\mathrm{a}},\overline{\mathrm{b}}) \le 1$ *for* $0 \le \overline{\mathrm{a}},\overline{\mathrm{b}} \le 2$, *and* $0 \le z,p \le 1$.
Proof: Let

$$q = |p|^2, \quad t = z^2.$$

Then

$$g = \left[1 - \overline{\mathrm{b}}t\,(1-q) - \overline{\mathrm{a}}t\,(1-t)\,q\right]^2 + \overline{\mathrm{a}}\overline{\mathrm{b}}t\,(1-t)^2.$$

We first show that $g \le 1$ on the boundary of the (t,q) domain. Consider the t boundary values:

$$g\,(t=0) = 1,$$

and since $\overline{\mathrm{b}}(1-q) \le 2$,

$$g\,(t=1) \le 1.$$

For the q boundary values:

$$g\,(q=0) - 1 = \overline{\mathrm{b}}t\left[-2 + \overline{\mathrm{b}}t + \overline{\mathrm{a}}\,(1-t)^2\right] \le \overline{\mathrm{b}}t\Big[-2 + \overline{\mathrm{b}}t + \overline{\mathrm{a}}\,(1-t)\Big] \le \overline{\mathrm{b}}t\Big[-2 + 2\,(t+1-t)\Big] = 0,$$

while

$$g\,(q=1) - 1 = \overline{\mathrm{a}}t\,(1-t)\Big[-2 + \overline{\mathrm{a}}t\,(1-t) + \overline{\mathrm{b}}\,(1-t)\Big] \le \Big[-2 + \overline{\mathrm{a}}t + \overline{\mathrm{b}}\,(1-t)\Big] \le 0.$$

To finish the proof, we need only to show that, for each fixed t, g (a quadratic function in q) does not have a maximum for $0 < q < 1$. This is achieved by showing that either $\frac{\partial g}{\partial q}(q=1) \ge 0$, in which case g is either monotone nondecreasing or has a minimum in $0 < q < 1$, or that $\frac{\partial g}{\partial q}(q=0) \le 0$ and $\frac{\partial g}{\partial q}(q=1) \le 0$, in which case g is monotone nonincreasing in $0 < q < 1$. We proceed as follows:

$$\frac{\partial g}{\partial q} = 2\kappa\big[1 - \overline{\mathrm{b}}t + \kappa q\big],$$

where

$$\kappa = \overline{\mathrm{b}}t - \overline{\mathrm{a}}t + \overline{\mathrm{a}}t^2.$$

Note that

$$1 - \overline{\mathrm{b}}t + \kappa = 1 - \overline{\mathrm{a}}t(1-t) \ge 1 - \overline{\mathrm{a}}\frac{1}{4} \ge \frac{1}{2}. \tag{4.5}$$

- Case $\overline{\mathrm{a}} \le \overline{\mathrm{b}}$ $(\alpha \ge \beta)$:[2] In this case $\kappa \ge 0$, so by (4.5)

$$\frac{\partial g}{\partial q}\,(q=1) = 2\kappa\left(1 - \overline{\mathrm{b}}t + \kappa\right) \ge 2\kappa\,\frac{1}{2} \ge 0.$$

- Case $\overline{\mathrm{a}} \ge \overline{\mathrm{b}}$ $(\alpha \le \beta)$: Then $\kappa \le 0$ if $t \le t_0 = 1 - \frac{\overline{\mathrm{b}}}{\overline{\mathrm{a}}}$, and $\kappa \ge 0$ if $t \ge t_0$.
  - If $t \ge t_0$ $(k \ge 0)$. Then as in the case above, $\frac{\partial g}{\partial q}(q=1) > 0$.
  - If $t \le t_0$ $(k \le 0)$. Then by (4.5), both $\frac{\partial g}{\partial q}(q=1) < 0$ and $\frac{\partial g}{\partial q}(q=0) < 0$.

Thus ends the proof of the lemma and completes the proof of the theorem. □

[2] This case is the most important, since for the compatible Lagrangian hydrodynamics scheme applied to the Euler equations one must necessarily have $\beta = \frac{1}{2}$ for the scheme to conserve total energy.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

5. Instability for $\alpha < 1/2$ or $\beta < 1/2$. Both the fully implicit and the predictor-corrector schemes are unstable if either $\alpha < 1/2$ or $\beta < 1/2$, for any mesh ratios. We just need to show this in 1D for the special case of $p = u$, for then the schemes are being applied to $\frac{du}{dt} = \frac{\partial u}{\partial x}$, with one set of difference equations used to advance u at half-integer indexes $i + 1/2$ and a different set at integer indexes i. Instability occurs if some Fourier component is amplified. So by setting $\Phi_y = 0$ in the amplification factor $\mathbf{S}$ for the predictor-corrector factor, we need only to show first that $|1 - \alpha\Phi_x^2 + \mathrm{i}\Phi_x(1 - \alpha\beta(\Phi_x^2))| > 1$ if $\alpha < 1/2$. This is clearly the case if $|\Phi_x|$ is positive and sufficiently small since

$$\left|1 - \alpha\Phi_x^2 + \mathrm{i}\Phi_x\left(1 - \alpha\beta(\Phi_x^2)\right)\right|^2 = 1 + \Phi_x^2\underbrace{(1-2\alpha)}_{>0} + \underbrace{\Phi_x^4\,\alpha^2\beta^2\,\left(\frac{1}{\beta^2} + \Phi_x^2 - \frac{2}{\alpha\beta}\right)}_{\sim o(\Phi_x^2)}.$$

A similar argument works for the second row of $\mathbf{S}$ if $\beta < 1/2$. The fully implicit scheme yields to the same kind of analysis; we omit the details.

6. Numerical results. We solve the simple problem already described in [3] where, on a unit domain using a quadrangular mesh of 100×100, velocity is set to zero, density and sound speed are set to unity by using a perfect equation of state with γ, the ratio of specific heats, equal to 5/3 and reflective boundary conditions.

Since for this problem no velocity should develop, simulations are run for a very large number of time cycles, typically 10^5, and with varying values for the CFL number $\lambda = \lambda_x = \lambda_y$ and parameters α, β. A sensitive gauge of instability is to keep track of the total kinetic energy $K^\lambda(t^n) = \frac{1}{2}\sum[(u_{i,j}^n)^2 + (v_{i,j}^n)^2]$ for a given CFL number λ at a given time t_n. Since the density and domain size are scaled to unity, this number should remain at the square of machine precision, about 10^{-28}–10^{-30} in our case. For an unstable scheme, it is observed that $K^\lambda(t^n)$ grows by several orders of magnitude long before the 10^5 cycle limit is reached. In this way, one can accurately identify the stability boundary in the CFL number and α space for a given β.

2D wave equations. We solve the 2D wave equations with the rectangular predictor-corrector scheme. From Theorem 4.1 we have that

$$4\alpha\beta\max\left(\lambda_x^2, \lambda_y^2\right) \le 1 \quad \Longrightarrow \quad \lambda \le \frac{1}{2\sqrt{\alpha\beta}},$$

provided that $\lambda = \lambda_x = \lambda_y$. Thus, the maximum CFL number achievable, denoted $\mathrm{CFL}_{\max}$, therefore depends on α and β and is described by the following function:

$$\mathrm{CFL}_{\max} = \mathrm{CFL}_{\max}(\alpha,\beta) = \frac{1}{2\sqrt{\alpha\beta}}. \tag{6.1}$$

In Figure 6.1, we plot the maximal CFL as a function of α for a sample of five β values: $\beta = \{0.5, 0.6, 0.75, 0.9, 1.0\}$. The maximum CFL numbers obtained numerically with the code for a pair of α, β are superimposed; $+$ for $\beta = 0.5$, $*$ for $\beta = 0.6$, □ for $\beta = 0.75$, ■ for $\beta = 0.9$, and ○ for $\beta = 1.0$. As expected they perfectly match the theoretical CFL limit (6.1) given by Theorem 4.1.

2D Euler equations. We present the results obtained with our 2D Lagrangian predictor-corrector code for the Euler equations by using Lagrangian coordinates. In this context (see [3], for example) the conservation of total energy is ensured provided that $\beta = \frac{1}{2}$. In Figure 6.2, we plot seven data points (symbol +) obtained by the code which represent the maximum CFL number that can be used for a given α for which

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

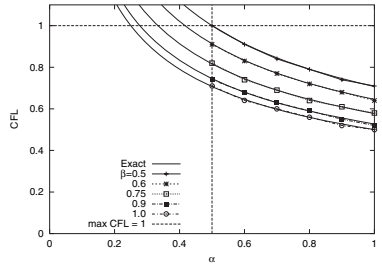


FIG. 6.1. 2D *wave equations—maximum CFL number* λ *as a function of* α *(*β *fixed) for the kinetic energy to remain on the order of machine precision for the test case. The vertical dashed line is the* $\alpha = 1/2$ *limit, and the horizontal one is the CFL* $= 1$ *limit. The theorem predicts the continuous thick lines, and the code produces the data:* $+$ *for* $\beta = 0.5$, $*$ *for* $\beta = 0.6$, □ *for* $\beta = 0.75$, ■ *for* $\beta = 0.9$, *and* ○ *for* $\beta = 1.0$. *Any scheme defined by a value* $\alpha \ge 1/2$ *and* $\beta \ge 1/2$ *is stable with the following CFL number:* $\lambda \le 1/(2\sqrt{\alpha\beta})$.

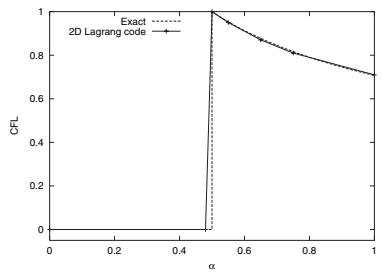


FIG. 6.2. *Hydrodynamics equations—maximum CFL number* λ *as a function of* α *(*$\beta = 1/2$*) for the kinetic energy to remain on the order of machine precision for the test case. The theorem predicts the continuous line; the 2D code produces the dashed line. Any scheme defined by a value* $\alpha \ge 1/2$ *is stable with the CFL number* $\lambda \le 1/\sqrt{2\alpha}$ *and is unstable otherwise.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

numerical stability is preserved. The exact function CFL (dashed line) predicted by (6.1) is superimposed.[3] We can easily see that any scheme defined by a value $\alpha \ge 1/2$ is stable with the CFL number $\lambda \le 1/\sqrt{2\alpha}$ and is unstable otherwise. We comment that for $\alpha = 1/2$ one exactly reaches the maximum CFL number $\lambda = 1$. The scheme with $\alpha = \beta = 1/2$ is the only staggered predictor-corrector scheme that reaches the maximum CFL number. Therefore the choice of using this scheme in several Lagrangian simulation codes, at least for this "optimal stability reason," seems justified. The results of our simulations perfectly match the predicted curve; this is also true of our 1D and 3D codes.

7. Conclusion. We have proposed the 2D wave equation as a linear constant coefficient model for 2D Lagrangian hydrodynamics, and we have established a sufficient stability condition for the standard staggered-grid numerical scheme. The 1D case is concurrently obtained as a special case of the 2D analysis.

This stability boundary has been tested with the 2D compatible Lagrangian hydrodynamics scheme for the wave equations and the Euler equations in Lagrangian coordinates. The numerical results fit the theoretical curve, showing that

- the general belief that for such numerical schemes "the more implicit, the more stable" is not always true,
- the classical choice made by several generations of Lagrangian code developers to use the $\alpha = \beta = 1/2$ scheme is reasonable as it leads to optimal stability results (at least for the artificial test case presented).

For the Euler equations these numerical results have been obtained in 1D, 2D, and 3D.

Acknowledgment. The authors thank Misha Shashkov for many fruitful discussions.

REFERENCES

[1] R. COURANT, K. O. FRIEDRICHS, AND H. LEWY, *Uber die partiellen Differenzengleichungen der mathematischen Physik*, Math. Ann., 100 (1928), pp. 32–74.
[2] P. D. LAX AND B. WENDROFF, *Difference schemes for hyperbolic equations with high order of accuracy*, Comm. Pure Appl. Math., XVII (1964), pp. 381–398.
[3] A. L. BAUER, D. E. BURTON, E. J. CARAMANA, R. LOUBÈRE, M. J. SHASHKOV, AND P. P. WHALEN, *The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics*, J. Comput. Phys., 218 (2006), pp. 572–593.
[4] E. J. CARAMANA, D. E. BURTON, M. J. SHASHKOV, AND P. P. WHALEN, *The construction of compatible hydrodynamics algorithms utilizing conservation of total energy*, J. Comput. Phys., 146 (1998), pp. 227–262.
[5] M. GOLDBERG AND E. TADMOR, *On the numerical radius and its applications*, Linear Algebra Appl., 42 (1982), pp. 263–284.
[6] R. D. RICHTMYER AND K. W. MORTON, *Difference Methods for Initial Value Problems*, 2nd ed., Interscience, 1967.
[7] W. H. HUI, P. Y. LI, AND Z. W. LI, *A unified coordinate system for solving the two-dimensional Euler equations*, J. Comput. Phys., 153 (1999), pp. 596–637.
[8] W. H. HUI, *The unified coordinate system in computational fluid dynamics*, Commun. Comput. Phys., 2 (2007), pp. 577–610.

[3] In this case $\mathrm{CFL} = 1/2\sqrt{\alpha\beta} = 1/\sqrt{2\alpha}$, because $\beta = 1/2$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

1.3.3 Volume consistency

One annoying feature of the compatible staggered Lagrangian scheme is the existence of two different cell volumes as pointed out in [12]. Although this difference was involved in a measure of consistency for numerical simulations, the conclusion of this work left a sour taste and a feeling of incompleteness. With M. Shashkov and B. Wendroff we further investigated this point and this work has led to paper [16] entitled *Volume consistency in a staggered grid Lagrangian hydrodynamics scheme*.
Let us remind that the classic compatible staggered Lagrangian compressible hydrodynamics scheme involves a choice of how internal energy is advanced in time. The options depend on two ways of defining cell volumes : an indirect one, that guarantees total energy conservation, and a direct one that computes the volume from its definition as a function of the cell vertices. It is shown that the motion of the vertices can be defined so that the two volume definitions are identical. In this note we construct a modification of the scheme such that we remove the ambiguity in the definition of cell volume that results from requiring both total energy conservation and the modeling of the internal energy advance from the differential equation $\frac{d\varepsilon}{dt} + p\frac{d(1/\rho)}{dt} = 0$. This is brought about by appropriately relating the motion of cell vertices to the cell volume change. This approach is purely algebraic.

More precisely we showed that the two volume definitions are equivalent if and only if certain matrices are equal. We explicitly gave the form of these matrices. The classical explicit discretizations of the scheme are such that these matrices are not equal. We have therefore developed a modification of the scheme involving an "inner consistency iterative procedure" [8] for the matrices to match at convergence. This procedure uniquely implies the discretizations of momentum equation and internal energy equation in order to get volume consistency and total energy conservation. Then we have tested this modification in 2D axisymetric geometry $(r - z)$ on the Coggeshall adiabatic compression problem [57]. These results compare the inconsistent Control Volume (iCV) scheme, say the classical scheme and the proposed consistent Control Volume (CV) scheme. They are reproduced in Fig 1.7. We observed that in addition to energy conservation the cell entropies are almost exactly conserved.

This paper is reproduced in the following pages.

8. From an implementation point of view this consists in an inner loop within the usual predictor-corrector loop.

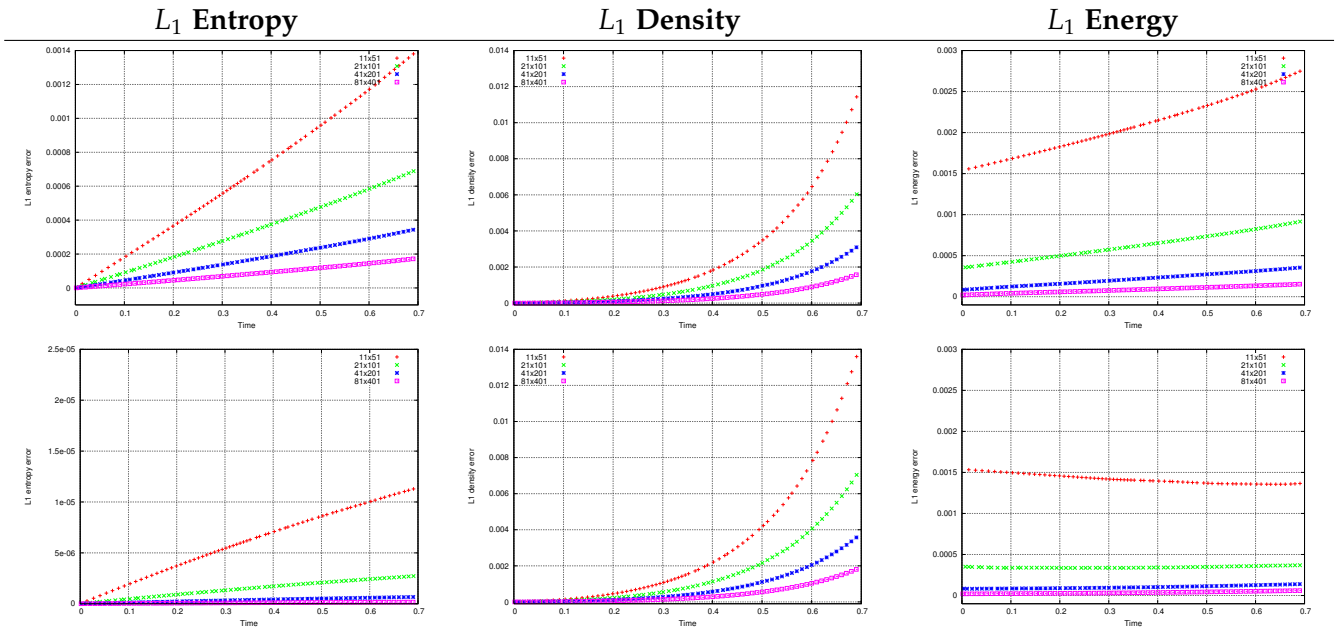


Figure 1.7 – *Numerical results from paper [16]. Coggeshall problem on a quarter of a disk in $r-z$ geometry — Entropy (left panels), Density (middle panels), Energy (right panels) — L_1 errors as functions of time for successively refined meshes 11×51 up to 81×401 for a CFL condition $1/4$. — Top line : Inconsistent control Volume (iCV) scheme — Bottom line : Consistent control Volume (CV) scheme. The scales for the entropy error plots are different as the consistent control volume scheme exhibits a quasi-exact entropy conservation.*

Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 3731–3737
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Short Note

Volume consistency in a staggered grid Lagrangian hydrodynamics scheme

R. Loubère [a,*], M. Shashkov [b], B. Wendroff [b]

[a] Université de Toulouse, UPS, IMT (Institut de Math. de Toulouse) et CNRS, F-31062 Toulouse, France
[b] Theoretical Division, T-7, Los Alamos National Laboratory, MS-B284, Los Alamos, NM 87545, USA

Received 8 August 2007; received in revised form 28 November 2007; accepted 7 January 2008
Available online 17 January 2008

Abstract

Staggered grid Lagrangian schemes for compressible hydrodynamics involve a choice of how internal energy is advanced in time. The options depend on two ways of defining cell volumes: an indirect one, that guarantees total energy conservation, and a direct one that computes the volume from its definition as a function of the cell vertices. It is shown that the motion of the vertices can be defined so that the two volume definitions are identical. A so modified total energy conserving staggered scheme is applied to the Coggeshall adiabatic compression problem, and now also entropy is basically exactly conserved for each Lagrangian cell, and there is increased accuracy for internal energy. The overall improvement as the grid is refined is less than what might be expected.
© 2008 Elsevier Inc. All rights reserved.

Keywords: Staggered Lagrangian schemes; Volume consistency; Entropy; Hydrodynamics

1. Introduction

In this note we construct a modification of the classic staggered grid Lagrangian compressible hydrodynamics scheme as described, for example, in [2]. With this modification we remove the ambiguity in the definition of cell volume that results from requiring both total energy conservation and the modeling of the internal energy advance from the differential equation $\frac{de}{dt} + p\frac{d(1/\rho)}{dt} = 0$. This is brought about by appropriately relating the motion of cell vertices to the cell volume change. Our approach is algebraic and simply stated. We then test this modification on the Coggeshall adiabatic compression problem [5]. We observe that now in addition to energy conservation the cell entropies are almost exactly conserved.

In the staggered scheme there are two sets of variables. First, for definiteness specifically in two dimensions, there is a set of indexed nodes or vertices at which the variables are coordinates (x_i, y_i), velocities (u_i, v_i), and nodal masses m^i. Next, there is a set of indexed cells at which the variables are cell volumes V_j, masses m_j, densities ρ_j, specific internal energies e_j, and pressures p_j. The set of cell (resp. node) indexes is $\mathscr{J}$ (resp. $\mathscr{I}$). Both sets of data are given at the start of a time step. Nodal and cell masses are Lagrangian, that is, independent of time, and $\rho_j = m_j/V_j$. Pressure is given by an equation of state, $p_j = p(\rho_j, e_j)$. The set of nodes i belonging to the same cell j is $\mathscr{I}_j$. Likewise the set of cells j sharing the same node i is $\mathscr{J}_i$

Two critical but standard assumptions follow, namely:

(i) the volume of any cell is a computable function of the nodal coordinates; typically, the volume of a cell will only depend on those nodes that are the vertices of the cell;
(ii) the velocities are constant during the time step. If those velocities are $(\bar{u}_i, \bar{v}_i)$, we can define the nodal coordinates as functions of time in the interval $(t^n,\ t^{n+1} = t^n + \Delta t)$ for $t \in [t^n, t^{n+1}]$ as

$$x_i(t) = x_i^n + \bar{u}_i(t - t^n), \quad y_i(t) = y_i^n + \bar{v}_i(t - t^n),$$

so that

$$x_i^{n+1} = x_i^n + \bar{u}_i\Delta t, \quad y_i^{n+1} = y_i^n + \bar{v}_i\Delta t.$$

This defines cell volume V_j as a function of time, and we have the identity

$$V_j^{n+1} - V_j^n = \int_{t^n}^{t^{n+1}} \frac{dV_j}{dt}\,dt = \sum_{i\in\mathscr{I}_j} \bar{u}_i \int_{t^n}^{t^{n+1}} \frac{\partial V_j}{\partial x_i}\,dt + \sum_{i\in\mathscr{I}_j} \bar{v}_i \int_{t^n}^{t^{n+1}} \frac{\partial V_j}{\partial y_i}\,dt. \tag{1}$$

Noting that $\bar{u}_i = (x_i^{n+1} - x_i^n)/\Delta t, \bar{v}_i = (y_i^{n+1} - y_i^n)/\Delta t$, (1) is just another way of writing the cell volume at time t^{n+1}, V_j^{n+1}, as a function of the coordinates at time t^{n+1}. Specific instances of this are given in Section 2.

We define matrices $\mathbf{A}$ and $\mathbf{B}$ by their entries

$$\mathbf{A}_{ji} = \int_{t^n}^{t^{n+1}} \frac{\partial V_j}{\partial x_i}\,dt, \quad \mathbf{B}_{ji} = \int_{t^n}^{t^{n+1}} \frac{\partial V_j}{\partial y_i}\,dt, \tag{2}$$

with $j \in \mathscr{J}$ and $i \in \mathscr{I}$, so that $\mathbf{A}$ and $\mathbf{B}$ are rectangular sparse sized $|\mathscr{J}| \times |\mathscr{I}|$ matrices, where $|\mathscr{I}|$ is the size of $\mathscr{I}$. They will play a role in the evolution of the hydrodynamic variables since (1) becomes

$$V_j^{n+1} - V_j^n = \sum_{i\in\mathscr{I}_j} (\mathbf{A}_{ji}\bar{u}_i + \mathbf{B}_{ji}\bar{v}_i). \tag{3}$$

An important point to emphasize here is that $\mathbf{A}$ and $\mathbf{B}$ are not in general simple time averages of the integrands, except in the case of Cartesian coordinates.

2. Momentum, energy, entropy

2.1. Momentum

The differential equations for momentum are

$$\rho\frac{du}{dt} = -(\mathrm{grad}\,p)_x, \quad \rho\frac{dv}{dt} = -(\mathrm{grad}\,p)_y.$$

Staggered grid momentum difference equations have the form

$$m^i(u_i^{n+1} - u_i^n) = \sum_{j\in\mathscr{J}_i} p_j\mathbf{a}_{ij}, \quad m^i(v_i^{n+1} - v_i^n) = \sum_{j\in\mathscr{J}_i} p_j\mathbf{b}_{ij}, \tag{4}$$

where the matrix $\mathbf{a}$ involves geometrical grid vectors so that $\sum_{j\in\mathscr{J}} p_j\mathbf{a}_{ij}$ is an approximation of the integral of the pressure gradient in x direction over cell indexed j, likewise for $\mathbf{b}$. Matrices $\mathbf{a}$ and $\mathbf{b}$ are rectangular sparse $|\mathscr{I}| \times |\mathscr{J}|$.[1] We now set $\bar{u}_i = \frac{1}{2}(u_i^{n+1} + u_i^n)$, $\bar{v}_i = \frac{1}{2}(v_i^{n+1} + v_i^n)$. To each pressure p_j there will be added an artificial viscosity q_j to deal with shock waves. However, we take q to be zero in our analysis and in the example presented later.

2.2. Energy

Kinetic energy is a nodal quantity for any time t^n, $K_i^n = \frac{1}{2}m^i((u_i^n)^2 + (v_i^n)^2)$, and the total kinetic energy is $K^n = \sum_{i\in\mathscr{I}} K_i^n$. Then since

$$\frac{1}{2}m^i((u_i^{n+1})^2 - (u_i^n)^2) + \frac{1}{2}m^i((v_i^{n+1})^2 - (v_i^n)^2) = \bar{u}_i\sum_{j\in\mathscr{J}_i} p_j\mathbf{a}_{ij} + \bar{v}_i\sum_{j\in\mathscr{J}_i} p_j\mathbf{b}_{ij},$$

that is to say $K_i^{n+1} - K_i^n = \bar{u}_i\sum_{j\in\mathscr{J}_i} p_j\mathbf{a}_{ij} + \bar{v}_i\sum_{j\in\mathscr{J}_i} p_j\mathbf{b}_{ij}$, the change in total kinetic energy is

$$K^{n+1} - K^n = \sum_{i\in\mathscr{I}} \bar{u}_i \sum_{j\in\mathscr{J}_i} p_j\mathbf{a}_{ij} + \sum_{i\in\mathscr{I}} \bar{v}_i \sum_{j\in\mathscr{J}_i} p_j\mathbf{b}_{ij}.$$

The total energy is taken to be the sum of the total nodal kinetic energy and total cell internal energy, that is, $E = \left(\sum_{j\in\mathscr{J}} m_j e_j\right) + K$. Then energy conservation requires that $(E^{n+1} - E^n) = 0$, that is to say $\left(\sum_{j\in\mathscr{J}} m_j(e_j^{n+1} - e_j^n)\right) + (K^{n+1} - K^n) = 0$, or

$$\sum_{j\in\mathscr{J}} \left(m_j(e_j^{n+1} - e_j^n) + p_j \sum_{i\in\mathscr{I}_j} \bar{u}_i\mathbf{a}_{ij} + p_j \sum_{i\in\mathscr{I}_j} \bar{v}_i\mathbf{b}_{ij} \right) = 0.$$

Thus a *sufficient condition* for energy conservation, *no matter how the* $\mathbf{a}$ *and* $\mathbf{b}$ *matrices have been defined*, is that for any cell, the internal energy evolution be[2]

$$m_j(e_j^{n+1} - e_j^n) + p_j \sum_{i\in\mathscr{I}_j} (\bar{u}_i\mathbf{a}_{ij} + \bar{v}_i\mathbf{b}_{ij}) = 0. \tag{5}$$

2.3. Entropy

For adiabatic flows the entropy S satisfies $T\frac{dS}{dt} = \frac{de}{dt} + p\frac{d(1/\rho)}{dt} = 0$. The Lagrangian difference expression of this, according to (3), is

$$m_j(e_j^{n+1} - e_j^n) + p_j(V_j^{n+1} - V_j^n) \equiv m_j(e_j^{n+1} - e_j^n) + p_j \sum_{i\in\mathscr{I}_j} (\bar{u}_i\mathbf{A}_{ji} + \bar{v}_i\mathbf{B}_{ji}) = 0. \tag{6}$$

It can now be seen that there are *two implied volume definitions*,[3] following from (5) and (6). They will be identical if for all $i \in \mathscr{I}$, $j \in \mathscr{J}$

$$\mathbf{a}_{ij} = \mathbf{A}_{ji} \quad \text{and} \quad \mathbf{b}_{ij} = \mathbf{B}_{ji}. \tag{7}$$

and then we will have both total energy conservation and (6).

This is different from the approach in, e.g. [1], where the $\mathbf{a}$ and $\mathbf{b}$ matrices are chosen in order to satisfy some symmetry conditions and then the $\mathbf{A}$ and $\mathbf{B}$ matrices are defined by (7), in which case (3) cannot be expected to hold. Indeed, the discrepancy between $\sum_{i\in\mathscr{I}_j}(\bar{u}_i\mathbf{a}_{ij} + \bar{v}_i\mathbf{b}_{ij})$ and $(V_j^{n+1} - V_j^n)$ for the area-weighted scheme of [1] and its effect on stability is the subject of [4].

3. Two geometries

For specific examples we need to indicate the relation between nodes and cells and we have to compute the volume change matrices $\mathbf{A}$ and $\mathbf{B}$.

[1] In [4] the momentum equations corresponding to (4) can be seen on page 575 Eq. (2.1); it involves the "corner force" entity: f_z^p for a zone/cell z and a point/node p of z. Indeed we urge the reader to consult Section 2 of [4, pp. 575–577] to get a detailed description of the original staggered Lagrangian scheme viewed from a different perspective. The corner force in x direction from this work corresponds in our notation to $\frac{1}{2}p_j\mathbf{a}_{ij}$.

[2] Eq. (5) can be seen in [4, Eq. (2.2), p. 575], or in [2, Eqs. (12) and (13), p. 234–235].

[3] See the discussion in [2, Section 2.3, pp. 244–245], where this volume inconsistency is referred as to an entropy error.

[4] For example see Fig. 1 in [4, p. 575].

and this defines the matrix elements of (3). **A** and **B** being defined, it uniquely implies the discretizations of (4) and (5) in order to get volume consistency and total energy conservation.

The above expressions were easily incorporated into the ALE-INCUBATOR [3] code in order to obtain the computations in Section 4.

4. The full predictor corrector scheme and the Coggeshall cylindrical adiabatic compression problem

This is a modification of the predictor corrector scheme of [2]. The scheme solves the implicit system

$$m^i(u_i^{n+1} - u_i^n) = \sum_{j\in\mathscr{J}_i} p_j \mathbf{a}_{ij}, \quad m^i(v_i^{n+1} - v_i^n) = \sum_{j\in\mathscr{J}_i} p_j \mathbf{b}_{ij}, \tag{12}$$

$$x_i^{n+1} = x_i^n + \bar{u}_i \Delta t, \quad y_j^{n+1} = y_j^n + \bar{v}_i \Delta t. \tag{13}$$

This is solved by simple substitution, keeping the pressures fixed. That is, predict the nodal coordinates in the right sides of (12) to get predicted $\bar{u}$ and $\bar{v}$, and then use (13) to obtain corrected coordinates. Let us call this the *inner consistency iteration*. This produces the new cell volumes V_j^{n+1} which can then be entered in the internal energy Eq. (6) to get a new internal energy and then a new pressure. But then we can iterate on the pressure (*outer iteration*), putting $p_j = \frac{1}{2}(p_j^{n+1} + p_j^n)$. The currently used method does the consistency iteration to convergence,[5] and then just one outer corrector iteration.[6]

4.1. The Coggeshall problem

The ALE-INCUBATOR [3] code is used to obtain the following numerical tests. The code is run without artificial viscosity and without anti-hourglass forces (see [3] and the references therein), so that only pressure forces enter the calculation as described in this note.

The choice of numerical tests is limited to tests free of shock waves and hourglass spurious modes; the Coggeshall adiabatic compression is described in [6].

The geometry is 2D r–z cylindrical. A sphere of initial radius $R = 1.0$ is filled with a perfect gas ($\gamma = 5/3$) in motion leading to the following exact solution $u^{ex}(t) = -\frac{r(t)}{1-t}$, $v^{ex}(t) = -\frac{z(t)}{4(1-t)}$, $\rho^{ex}(t) = (1-t)^{-9/4}$, $e^{ex}(t) = \left(\frac{3\varepsilon(t)}{8(1-t)}\right)^2$. At each boundary, the exact velocity is imposed up to the final time $t^n = 0.7$. Initial and final meshes can be seen [6].[7] We then look at various errors, comparing Consistent control Volume (CV) method, as described in this note, to the original discrete compatible formulation of Lagrangian hydrodynamics scheme, referred as in Consistent control Volume method and labeled (iCV) see [1,2]. The grid is rectangular polar made of $n_r \times n_z$ nodes, and refined several times in r and z directions by a factor 2.

4.2. Entropy, density and specific internal energy errors

For any mesh we compute the error in density ρ, entropy S and energy e, (the number of cells being n_c, $\mathbf{x}_j = (r_j, z_j)^t$ and Q stands for ρ, S, or e) as

$$\varepsilon_Q^n = \frac{1}{n_c} \sum_{j\in\mathscr{J}} |Q^{ex}(\mathbf{x}_j, t^n) - Q_j^n| / \max_{j\in\mathscr{J}} |Q^{ex}(\mathbf{x}_j, t^n)|.$$

Fig. 1 compares the errors ε_Q^n as functions of time ($t_n \leqslant 0.7$) for different mesh sizes. This figure shows that: (i) errors decrease as the mesh is refined for iCV and CV, (ii) asymptotically, a ratio 2 (first order convergence) is

[5] That is to say if ν is the iteration indices and $\mathbf{x}_i = (x_i, y_i)$, convergence of the inner consistency iteration is attained if $\epsilon = \frac{\sum_{j\in\mathscr{J}} \|\mathbf{x}_j^{\nu+1} - \mathbf{x}_j^{\nu}\|^2}{\sum_{j\in\mathscr{J}} \|\mathbf{x}_j^{\nu+1}\|^2} \leqslant 10^{-10}$.

[6] The predictor corrector scheme from [2] simply does one inner iteration and one outer iteration. In the case of cylindrical geometry, in [2] the Cartesian geometrical vector $\vec{a}_1$ (see Fig. 4, p. 249) is modified into $\vec{a}_1(3r_1 + r_2)/4$ on p. 261: This can not fulfill consistency of volumes and total energy conservation. In this new method $\vec{a}_1$ is replaced by $\frac{1}{6}\left(\left(\frac{2}{3}r_1^* + \frac{1}{3}r_2^*\right)(2\vec{a}_1^* + \vec{a}_1^n) + \left(\frac{2}{3}r_1^n + \frac{1}{3}r_2^n\right)(\vec{a}_1^* + 2\vec{a}_1^n)\right)$ with * referring to the most updated value from the inner consistency iteration.

[7] See Fig. 2, p. 5 for a 20 × 20 mesh of [6].

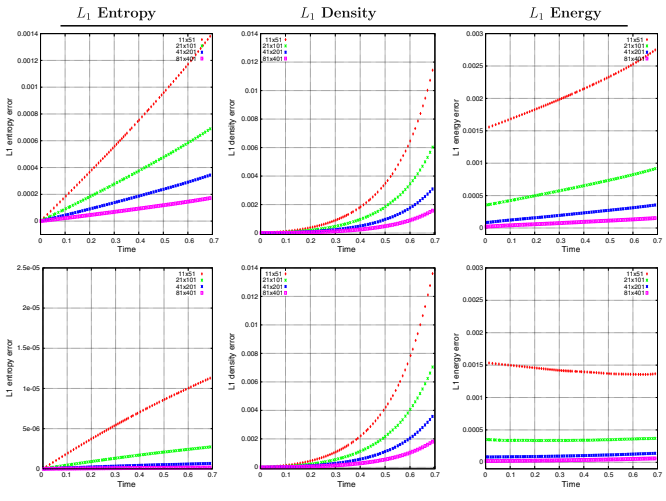


Fig. 1. Coggeshall problem on a quarter of a disk in r–z geometry – Entropy (left panels), density (middle panels), energy (right panels) – L_1 errors as functions of time for successively refined meshes 11 × 51 up to 81 × 401 for a CFL condition 1/4. *Top*: Inconsistent control Volume (iCV) scheme. *Bottom*: Consistent control Volume (CV) scheme. The scales for the entropy error plots are different as the consistent control Volume scheme exhibits a quasi-exact entropy conservation.

obtained for any variable, (iii) CV is nearly exact for entropy and more accurate for internal energy, but density accuracy is not increased.

Finally then, we have proposed a staggered Lagrangian numerical scheme with the following properties:

- It is volume consistent: there is no ambiguity in the cell volume definition.
- Total energy is conserved.
- For the adiabatic compression Coggeshall problem, with the artificial viscosity set to zero, cell entropies are almost exactly conserved.

Acknowledgments

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory, under Contract W-7405-ENG-36 and Contract DE-AC52-06NA25396. The authors acknowledge the partial support of the DOE Advance Simulation and Computing (ASC) Program and the DOE Office of Science ASCR Program, and the Laboratory Directed Research and Development program (LDRD) at the Los Alamos National Laboratory.

References

[1] E.J. Caramana, P.P. Whalen, Numerical preservation of symmetry properties of continuum problems, J. Comput. Phys. 141 (1998) 174.
[2] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1998) 227.
[3] R. Loubère, First steps into ALE INC(ubator) – Version 2.0.0, Los Alamos National Laboratory Report LA-UR-04-8840, 2004.
[4] A.L. Bauer, D.E. Burton, E.J. Caramana, R. Loubère, M.J. Shashkov, P.P. Whalen, The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics, J. Comput. Phys. 218 (2006) 572.
[5] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (1998) 70.
[6] L.G. Margolin, M.J. Shashkov, M.A. Taylor, Symmetry-preserving discretizations for Lagrangian gas dynamics, in: P. Neittaanmäki, T. Tiihonen, P. Tarvainen, (Eds.), Proceedings of the 3rd European Conference, Numerical Mathematics and Advanced Applications, World Scientific publisher, 2000, pp. 725–732.

1.4 Special additions

In this section I present some other topics related to the compatible staggered Lagrangian scheme to which my colleagues and myself have studied. More precisely in this section we treat the following subjects :

In the endless story on human being struggling with artificial viscosity, E.J. Caramana and I made a contribution with article [11] entitled *"Curl-q" : A vorticity damping artificial viscosity for essentially irrotational Lagrangian hydrodynamics calculations*. The goal of this work is to supplement the so-called edge artificial viscosity, see section 1.2.3, with a viscous term that is designed to eliminate spurious vorticity.

The purpose of the work made by E.J. Caramana and I, published in [10] under the title *The Force/Work Differencing of Exceptional Points in the Discrete, Compatible Formulation of Lagrangian Hydrodynamics* is to complete the compatible formulation of Lagrangian hydrodynamics by addressing the remaining finite-volume discretization questions that arise when treating grid points that must be internally enslaved within the grid to prevent timestep collapse. In other words what we have called "exceptional points". The work in article [10] focuses on developing a treatment of exceptional points such that collateral damages brought by the existence of exceptional points are literaly (or at least virtualy) absent.

With M. Kucharik, R. Liska and L. Bednarik at CTU in Prague (Czech Republic) we have faced a situation where a slide-line treatment for the compatible staggered Lagrangian scheme was needed. Starting with the paper of E.J. Caramana [80] we have published in [25] a work based on two enhancements — interpolated interaction instead of a simple one-to-one point interaction described in the previous article, and a numerical surface tension model improving the stability of the interface. Both improvements stabilize the slide line and lead to more realistic results, as shown on selected numerical examples such as pure sliding, some sanity checks such as the Saltzman piston, two sliding rings, and some more realistic simulations like the explosion with sliding and the bullet in a channel.

Trying to unite cell-centered and staggered Lagrangian schemes into a common framework in order to extrude the similarity of these supposedly different approaches is the next considered topic. This subject has been brought to light by P.-H. Maire all alone. At the very end of his thoughte he has been further joined by P. Váchal and me for several publications in 2D [23] *Staggered Lagrangian Discretization Based on Cell-Centered Riemann Solver and Associated Hydrodynamics Scheme*, and its counterpart in 3D [24] : *3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver based artificial viscosity*. The bridge between the cell-centered and staggered approaches has led us to re-derive the staggered compatible scheme, and most of all re-define the concept of subcell force invoking Galilean invariance and thermodynamical consistency. More precisely we have drawn the basics to design a new form of artificial viscosity subcell force driven by a subcell-based positive definite tensor, M_{cp}, which is the true essence of the numerical scheme.

1.4.1 Vorticity damping artificial viscosity

The bane of Lagrangian hydrodynamics calculations in multi-dimensions is the appearance of vorticity that causes tangling of the mesh and consequent run termination. This vorticity may be numerical or physical in origin, and is in addition to the spurious "hourglass" modes associated with quadrilateral or hexahedral zones that in pure form have both zero curl and divergence associated with their velocity field. The purpose of this paper is to introduce a form of vorticity damping,

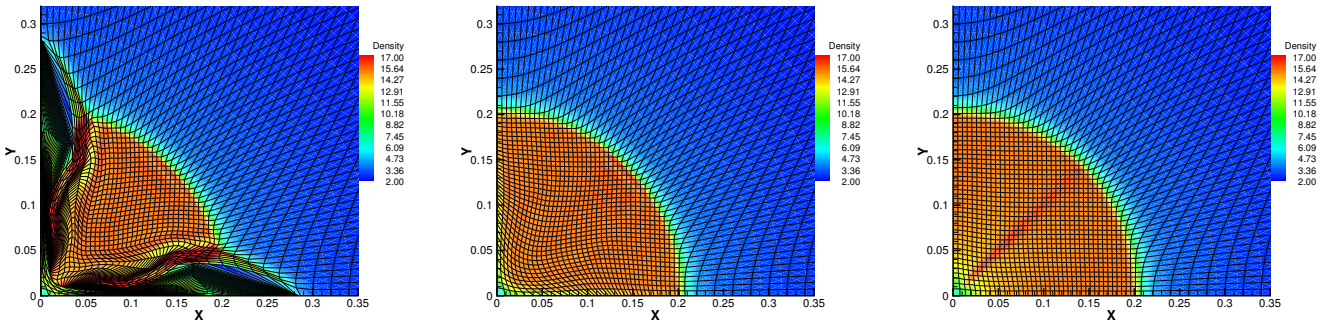


Figure 1.8 – *Numerical results from paper [11]. Noh problem, 2D Cartesian geometry — Mesh and density at time* $t = 0.6$*, the exact radius of the shock wave is* $r = 0.2$ *— Left : Edge viscosity without curl-q,* 1422 *time steps, — Middle : Edge viscosity with curl damping,* 606 *time steps, — Right : Tensor viscosity,* 423 *time steps.*

based on a edge-centered artificial viscosity [57], see also section 1.2.3, that extends the runtime and range of calculations over which a pure Lagrangian code can compute.

The origin of this work was the superiority of the tensor artificial viscosity [53] to damp spurious vorticity especially for the Noh problem on quadrangular grid for which the edge-centered artificial viscosity [57] is creating jets along axes leading to a serious lack of robustness and symmetry. The analysis of the tensor artificial viscosity was not trivial. The hope of recasting it into an edge-centered artificial viscosity that would ease the comparison and as such enlight the extra-terms was not a success. Consequently E.J. Caramana and I adopt a different strategy and developed an extra-term to the edge-centered artificial viscosity denoted as the "curl-q", because it is a function of the curl of the velocity field in a zone. Notice that this new "curl-q" does not resolve shock waves and is always to be utilized with an artificial viscosity that performs this task. This curl-q force is formulated as an analogy to the edge-centered artificial viscosity. The development and justification of this curl-q force is provided in this work. Moreover numerical results are given both in 2D and 3D showing the effectiveness of the approach. In particular, results are contrasted between this new term and the tensor artificial viscosity [53]. It is shown that these two forms give quite similar results in 2D. As instance we reproduce in Fig. 1.8 the results for the Noh problem on 2D Cartesian geometry with initially square zones (50×50 cells). Moreover in Fig. 1.9 we reproduce the 3D Noh results obtained with and without the curl-q vorticity damping term. The main advantage of this "curl-q" is its ability to damp curl like motion and this is also its main drawback when fluid instabilities, that is to say rotational flows, are expected to occur.

This paper is reproduced in the following pages.

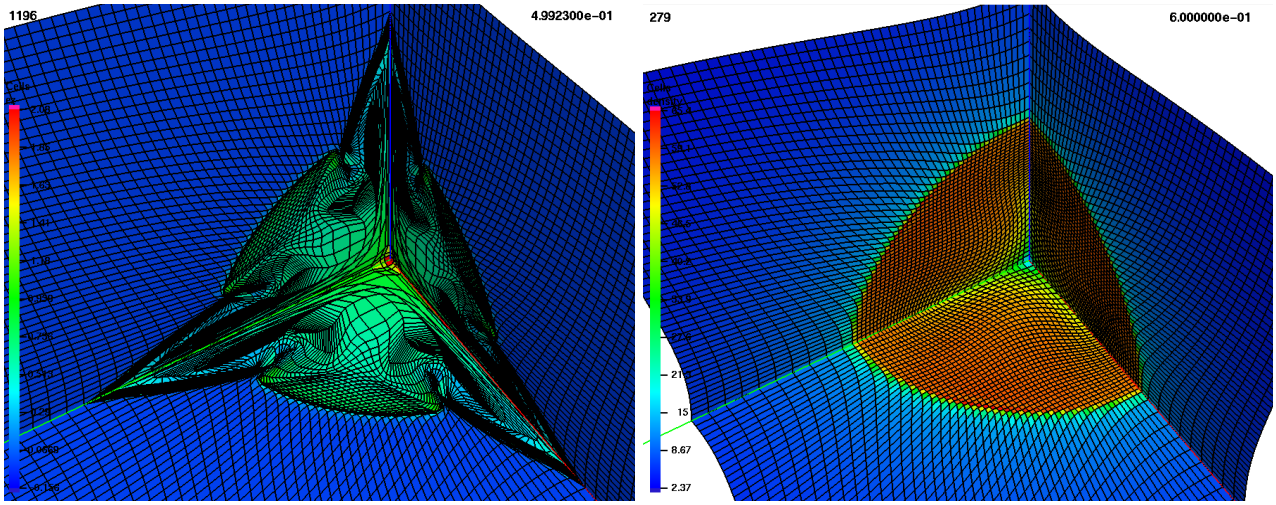


FIGURE 1.9 – *Numerical results from paper [11]. Spherical Noh problem, 3D Cartesian geometry, one octant,* 50^3 *cubic initial grid — Edge viscosity — Corner is the origin. Left : without curl-q damping, fails at time* $t = 0.5$ *due to mesh tangling at* 1196 *time steps. Specific internal energy is shown for contrast, density is unphysical due to the severe mesh distortion. — Right : with curl-q damping at time* $t = 0.6$, 279 *time steps. Variable plotted is density, which matches the exact solution (shock wave at radius* $r = 0.2$*, density of* 64*, red color).*

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 215 (2006) 385–391
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Short note

"Curl-q": A vorticity damping artificial viscosity for essentially irrotational Lagrangian hydrodynamics calculations

E.J. Caramana *, R. Loubère

CCS-2 and T-7, Los Alamos National Laboratory, MS-D413, Los Alamos, NM 87545, United States

Received 1 September 2005; received in revised form 8 November 2005; accepted 10 November 2005
Available online 27 December 2005

1. Introduction

The bane of Lagrangian hydrodynamics calculations in multi-dimensions is the appearance of vorticity that causes tangling of the mesh and consequent run termination. This vorticity may be numerical or physical in origin, and is in addition to the spurious "hourglass" modes associated with quadrilateral or hexahedral zones that in pure form have both zero curl and divergence associated with their velocity field.

The purpose of this note is to introduce a form of vorticity damping, based on a previously published edge-centered artificial viscosity [1], that extends the runtime and range of calculations over which a pure Lagrangian code can compute. Since the explicit inclusion of an artificial viscosity into the fluid equations is often referred to as the "q term", we denote this new term as the "curl-q", because it is a function of the curl of the velocity field in a zone. It is formulated in the context of the "discrete, compatible formulation of Lagrangian hydrodynamics" [2,3]. This employs a staggered placement of variables in space (velocity and position at nodes, with density and stresses in zones), but a predictor/corrector time integration scheme so that all variables are known at the same time level, allowing total energy to be exactly conserved [2]. This new "curl-q" does not resolve shock waves and is always to be utilized with an artificial viscosity that performs this task. In order to set the stage for the introduction of the new curl-q force, the edge-centered artificial viscosity given in [1] is briefly reviewed in a slightly simplified form; after this the curl-q force is formulated as an analogy to this edge-centered artificial viscosity. Numerical results are given in both 2D and 3D that display its effectiveness. In particular, results are contrasted between this new term and a recently published tensor artificial viscosity [4]. It is shown that these two forms give quite similar results in 2D. We end with a brief discussion concerning the validity of the use of this type of numerical device.

In Fig. 1 is shown a quadrilateral zone with its defining points, $i = 1$–4, and associated median mesh vectors $\vec{S}_i$. These vectors point in the indicated direction and have a magnitude of the surface area that lies between their defining points in 2D or in 3D [2,5]. In terms of the median mesh vector $\vec{S}_1$, the force exerted by the edge-centered artificial viscosity between points "1" and "2" from zone "z" is given by

$$\vec{f}_{21}^{\,\text{visc}} = c_1(1-\psi_{21})\rho_z(c_{s,z}+\Delta v_{21})[\Delta\vec{v}_{21}\cdot\vec{S}_1]\widehat{\Delta v_{21}}, \tag{1.1}$$

* Corresponding author. Tel.: +505 667 7779; fax: +505 667 3726.
E-mail addresses: ejc@lanl.gov (E.J. Caramana), loubere@lanl.gov (R. Loubère).

0021-9991/$ - see front matter. Published by Elsevier Inc.
doi:10.1016/j.jcp.2005.11.018

Note that this force is applied to grid points "1" and "2" with the same signs as that of the artificial viscosity force of Eq. (1.1). Also, if the term in square brackets is negative, $\vec{f}_{21}^{\,\text{curl-}q} = 0$, as required for it to act in a dissipative manner. Thus, all of the properties of the artificial viscosity that apply to Eq. (1.1), as enumerated in Ref. [1], are automatically transferred to the above definition of the curl-q force. In particular, this force turns off for rigid rotation due to the limiter function, and for grid-aligned flow because the lever arm $l_{21,\perp}$ is then zero. Because of this latter fact the limiter function has been weakened somewhat in Eq. (2.1) by using ψ_{21}^2 in place of ψ_{21} itself. This is a rather minor modification that allows us to run all test problems with the coefficient $c_1 = 1.0$ instead of somewhat larger values in certain cases. Also, to prevent numerical noise $l_{21,\perp}$ is set to zero if $l_{21,\perp}/|\vec{l}_{21}| < 10^{-3}$. Just as for the artificial viscosity force, the curl-q force contributes an effective sound speed along the given edge that is added to the sum of the squares of the zone sound speed and the artificial viscosity $c_{q,21}^2$ for timestep determination, as discussed previously. The square of the effective curl-q force sound speed is given simply by inspection of Eq. (2.1) as $c_1(1-\psi_{21}^2)(c_{s,z}+\Delta v_{21})l_{21,\perp}|\vec{\omega}_z|$.

The above form for the force is stated directly without reference to any term in a set of PDE's. That this is possible is a consequence of the discrete compatible form of Lagrangian hydrodynamics in that the discrete finite-volume description that it utilizes is more fundamental than the underlying continuum equations. For edge forces such as those given in Eqs. (1.1), (2.1), there exists no direct continuum limit, but only a loose analogy to terms in a differential equation. For the edge-centered artificial viscosity of Eq. (1.1), this analogy yields the term $q \equiv \rho(c_s + \Delta v)\Delta v$ in a zone "z", which in 1D allows one to insert $P \to P + q$ into all terms that contain the pressure "P". For the curl-q force of Eq. (2.2) such an analogy is more difficult to justify since $\vec{\omega}_z = 0$ in 1D, and this force vanishes. However, keeping this fact in mind one can still loosely make the analogy that the simple scalar form of the artificial viscosity ascribed above to Eq. (1.1) becomes

$$\rho_z(c_{s,z}+(\Delta v)_z)(\Delta v)_z \to \rho_z(c_{s,z}+|(\nabla\times\vec{v})_z|l_\perp|)l_\perp|(\nabla\times\vec{v})_z|. \tag{2.2}$$

In this equation "$l_\perp$" is to be interpreted as some average length that is normal to Δv taken over all zone edges. Then this term augments the pressure "P", subject to the limiter and force on/off switch, as does the usual artificial viscosity.

3. Numerical results and discussion

Two well-known test problems are utilized to demonstrate the effectiveness of the new curl-q vorticity damping mechanism. These are Saltzman's piston [6], and Noh's problem [7], computed in both 2D and 3D Cartesian geometry. In the 2D case results of the curl-q force, in conjunction with the edge-centered artificial viscosity, are compared to a recently published "tensor" form of the artificial viscosity [4]. Unlike the edge-centered form, this tensor artificial viscosity contains zone information, and thus reduces the dependence of the solution on the relation of the grid to the flow direction. These test problems have a $\gamma = 5/3$ ideal gas law equation of state; they are run with all artificial viscosity coefficients, as well as "c_1" in Eq. (2.1), set to unity. Anti-hourglass subzone pressure forces as described in [8], with a "merit factor" of unity, are also utilized for all simulations. Modifications to the standard setups referenced above are indicated; comparisons in 2D are made within the same computer code [9].

We begin with the standard setup for the Saltzman piston problem in 2D Cartesian geometry that has been elongated by an amount of 3:1 in the shock wave (or driving piston, from left to right with unit velocity)

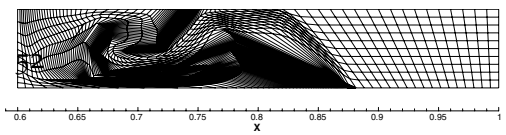


Fig. 2. Saltzmann piston in 2D – aspect ratio 3:1 – edge viscosity without curl damping, mesh at $t = 0.6$ before failure due to severe grid distorsion.

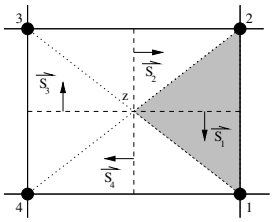


Fig. 1. Quadrilateral zone z: dotted interior lines delineate triangular subzones (gray), dashed lines form the median mesh with normal vectors $\vec{S}_1, \vec{S}_2, \vec{S}_3, \vec{S}_4$.

where the various terms are defined as follows: ρ_z and $c_{s,z}$ are the density and speed of sound in zone "z"; $\Delta\vec{v}_{21} \equiv \vec{v}_2 - \vec{v}_1$ is the difference in velocity along the edge "21" defined by points "1" and "2", from which are defined its magnitude $\Delta v_{21} \equiv |\Delta\vec{v}_{21}|$, and its direction $\widehat{\Delta v_{21}} \equiv \Delta\vec{v}_{21}/\Delta v_{21}$; the limiter function ψ_{21} is given in [1] and turns this force off for situations in which the velocity difference with respect to the edge direction is a linear function of the local coordinates; c_1 is a simple coefficient that is generally set to unity. This edge force is applied to points "1" and "2", with plus and minus signs, according to the compression switch $[\Delta\vec{v}_{21}\cdot\vec{S}_{21}]$; for the given definition of $\Delta\vec{v}_{21}$ and the orientation of the $\vec{S}_1$ vector in Fig. 1, compression is defined for $[\Delta\vec{v}_{21}\cdot\vec{S}_1] > 0$. Considering a frame of reference where $\vec{v}_1 = 0$, $\vec{f}_{21}^{\,\text{visc}}$ is then applied to point "1" with a "+" sign, and to point "2" with a "−" sign in order to ensure that this term acts in a dissipative manner [1]. If $[\Delta\vec{v}_{21}\cdot\vec{S}_1] < 0$, then $\vec{f}_{21}^{\,\text{visc}} = 0$. If the quantity in square brackets is summed over all edges of a zone and this sum is then divided by the zone volume, one obtains the negative divergence of the velocity field as defined by finite volume differencing: namely, $(\nabla\cdot\vec{v})_z = -\sum_e \Delta\vec{v}_e\cdot\vec{S}_e/V_z$, where "e" ranges over all edges of the given zone "z", and V_z is its volume. The work performed by Eq. (1.1), and all other forces in the discrete, compatible formulation of Lagrangian hydrodynamics, is given by the negative dot product of the particular force with the displacement in a timestep of the point that this force acts upon [2,3].

To correctly specify an effective sound speed with respect to the edge "21" for use in determining the CFL number for timestep determination, a sound speed "$c_{q,21}$" is defined by $c_{q,21}^2 \equiv c_1(1-\psi_{21})(c_{s,z}\Delta v_{21} + \Delta v_{21}^2)$; the effective edge sound speed is then defined as $\sqrt{c_{s,z}^2 + c_{q,21}^2}$. What has been done is to define an effective edge sound speed in the presence of dissipation that resembles the magnitude of the largest characteristic speed along an edge in 1D in the Lagrangian frame, which is $(c_{s,z} + \Delta v_{21})$. The CFL condition sets the timestep by requiring that this characteristic speed be resolved for all edges of all zones in the computational domain.

2. Curl-q force specification

The first attempt at turning Eq. (1.1) into a vorticity damping form of an artificial viscosity is to change the dot product in the compression switch into a cross product, obtaining $[\Delta\vec{v}_{21}\times\vec{S}_1]$. However, this quantity is a vector, and what is needed is an appropriate scalar. In order to construct a scalar the magnitude of $\Delta\vec{v}_{21}$ is separately factored out of this term and rewritten as a vector $l_{21,\perp}\vec{\omega}_z$, where $\vec{\omega}_z \equiv \sum_e \Delta\vec{v}_e\times\vec{S}_e/V_z$ is the finite volume difference form of the curl of the velocity field in zone "z", and $l_{21,\perp}$ is a length scale along edge "21" that is defined as $l_{21,\perp} \equiv |\Delta\vec{v}_{21}\times\vec{l}_{21}|$; $\vec{l}_{21}$ is the vector distance between points "1" and "2" of edge "21". Now a dot product can be formed between $\vec{\omega}_z$ and the remaining factor, $(\widehat{\Delta v_{21}}\times\vec{S}_1)$, to obtain the desired scalar. Assembling these "postulates" results in our form for the "curl-q" force as

$$\vec{f}_{21}^{\,\text{curl-}q} = c_1(1-\psi_{21}^2)\rho_z(c_{s,z}+\Delta v_{21})[l_{21,\perp}\vec{\omega}_z\cdot(\widehat{\Delta v_{21}}\times\vec{S}_1)]\widehat{\Delta v_{21}}. \tag{2.1}$$

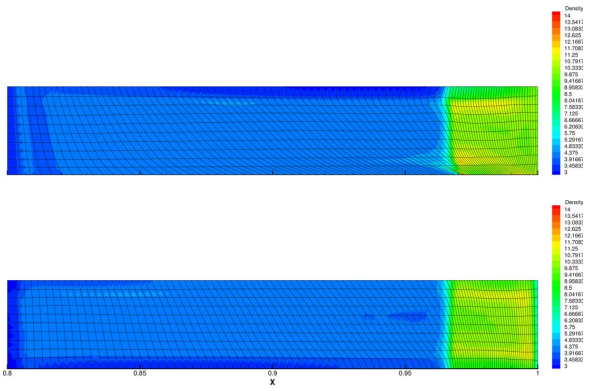


Fig. 3. Saltzmann piston in 2D – aspect ratio 3:1 – mesh and density. (Top) Edge viscosity with curl-q damping at $t = 0.8$. (Bottom) Tensor viscosity at $t = 0.8$.

direction. The result shown in Fig. 2 is the grid at time $t = 0.6$ without curl-q forces using the edge-centered artificial viscosity. The code fails due to severe grid distortion shortly after this time.

Results with the same initial and boundary conditions are given in the top and bottom parts of Fig. 3 at the final time of $t = 0.8$ for the edge-centered artificial viscosity with curl-q forces, and with the tensor artificial viscosity, respectively. At this time the shock wave has reflected from the fixed wall on the right side of the figure and is heading back towards the piston. Both of these cases show little grid distortion due to unphysical vorticity generation. These results show that the tensor artificial viscosity performs substantial damping of the unphysical vorticity that would otherwise be generated, and thus acts much like the curl-q damping force. If the grid is further elongated in the shock direction both the edge-centered artificial viscosity with curl-q force, and the tensor artificial viscosity, will fail; the former at about 4:1 aspect ratio, the latter at about 5:1. This is expected since as one further elongates the grid, pressure gradients in the ignorable vertical direction become larger, eventually leading to grid distortion and failure of the calculation. One can make modifications to increase the aspect ratio at which this failure occurs, but these can lead to non-positive heating of the artificial viscosity and curl-q forces that is unacceptable. Finally, if this problem is run with the grid highly elongated (10^4:1) in the ignorable, vertical direction as in [8], the net vorticity is small and the curl-q force has no discernable effect; in this case subzone, anti-hourglass forces are needed to obtain quality results.

Next, results are displayed for the Noh problem in 2D Cartesian geometry with initially square zones, 50×50, on a unit domain in both "x" and "y" directions. The initial velocity is minus one in the radial direction for all points, but zero at the (0, 0) origin; a shock propagates outward from this point. Fig. 4 shows the grid and density at time $t = 0.6$ for three cases: part (a) gives the result for the edge-centered artificial viscosity without curl-q forces; part (b) is with curl-q forces added, and part (c) is with the tensor artificial viscosity alone. The latter two results are similiar, again showing that the tensor artificial viscosity damps unphysical vorticity much as the curl-q force does. The large jets seen in part (a) without curl-q damping can be thought of as arising from "false" vorticity present in the initial conditions using a square grid. The Noh problem is irrotational in its initial conditions, but on a square grid, and using our finite-volume formulas to compute the curl of the velocity field in the zone, one finds zero curl only on the 45° line; the curl of the velocity field

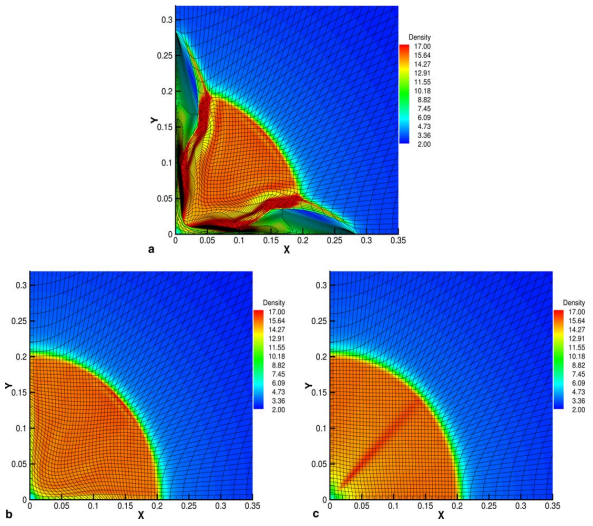


Fig. 4. Noh problem, 2D Cartesian geometry – mesh and density at time $t = 0.6$, the exact radius of the shock wave is $r = 0.2$. (a) Edge viscosity without curl damping, 1422 time steps. (b) Edge viscosity with curl damping, 606 time steps. (c) Tensor viscosity, 423 time steps.

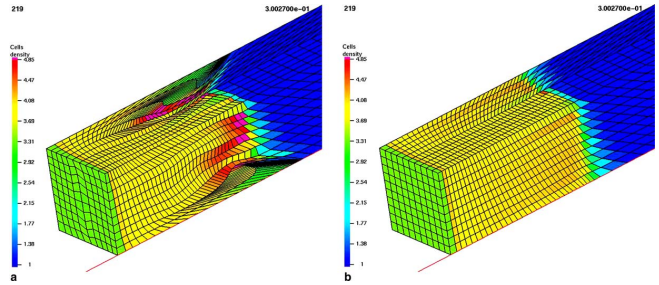


Fig. 5. Extended Saltzmann problem, 3D Cartesian geometry, aspect ratio 2:1 – edge viscosity – density and mesh at time $t = 0.3$, density behind shock should be 4.0. (a) Without curl-q damping; fails due to mesh tangling soon after this time. (b) With curl-q damping; code runs to about $t = 0.7$ and then fails due to mesh tangling.

increases with magnitude and with opposite signs as one approaches the horizontal and vertical axes. The limiter functions keep the artificial viscosity and the curl-q force off until the shock wave propagates past any given point. Then, without the curl-q force or the tensor artificial viscosity, strong unphysical vorticity generation is seen to occur along the axes.

In 3D we show results with the extended version of the Saltzman piston problem as previously defined in [5]. This extended problem is completely 3D in its setup; the Saltzman skewing of the grid is made to change parity uniformly in the third dimension. This grid is compressed by a 2:1 aspect ratio in the non-shock directions so that zones are elongated by this amount in the shock direction. Fig. 5 shows the grid and density profile at time $t = 0.3$ for two cases: part (a) is without the curl-q force; this calculation fails shortly after this time due to unphysical grid distortion; part (b) is with the curl-q force at the same time and shows a quite satisfactory result. This latter simulation will fail later (at about $t = 0.7$) due to grid distortion about the vertical plane with the high (red) density spot seen in the top-middle of part (b) of this figure. Lagrangian codes can always be broken, and no amount of curl smoothing or other devices guarantees run to completion, much less correct answers.

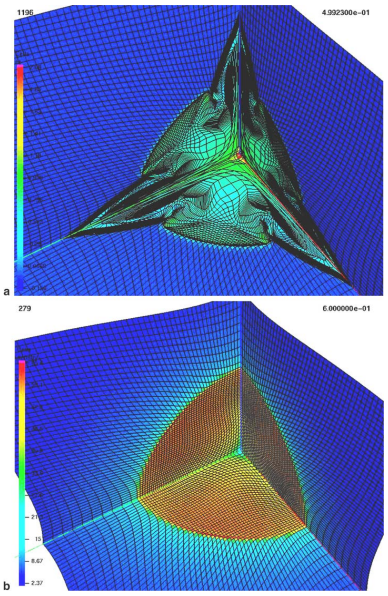


Fig. 6. Spherical Noh problem, 3D Cartesian geometry, one octant, 50^3 cubic initial grid – edge viscosity – corner is the origin. (a) Without curl-q damping, fails at time $t = 0.5$ due to mesh tangling at 1196 time steps. Specific internal energy is shown for contrast, density is unphysical due to the severe mesh distortion. (b) With curl-q damping at time $t = 0.6$, 279 time steps. Variable plotted is density, which matches the exact solution (shock wave at radius $r = 0.2$, density of 64, red color). Wall heating error is clearly visible.

Last, the Noh problem is calculated using initially cubic zones in 3D Cartesian geometry, $(50 \times 50 \times 50)$ for an octant. Fig. 6 has two parts: part (a) gives the grid at time $t = 0.5$ when the calculation fails due to excessive grid distortion without using any curl-q damping force. Jets along the three coordinate axes dominate the solution, which is colored using specific internal energy to obtain contrast; density has become unphysical due to zone collapse. Part (b) of this figure gives the result showing density at the final time $t = 0.6$ with curl-q forces. This latter result is of high quality for this problem; the ubiquitous wall-heating error is clearly visible, otherwise this is the correct solution for the 3D spherical Noh problem – a density of 64 behind the shock at a major radius of 0.2 at time $t = 0.6$.

In summary, the Lagrangian formulation of hydrodynamics produces reliable results only for irrotational, compressible, deterministic fluid flow. These conditions are strictly valid only in one-dimension. However, in multi-dimensions this may be the case to a substantial degree, particularly when the initial conditions of the simulation are homomorphic to one-dimensional geometry. Use of a vorticity damping mechanism as introduced herein is justified not only to remove unphysical vorticity, but also for calculations where physical vorticity is present but not dominant. In the latter case the solution may be incorrect in certain regions of the problem, but global, spatially integrated quantities may still remain substantially correct. If it is these quantities that are of principal interest, then Lagrangian frame calculations are appropriate. However, it is necessary to keep such calculations running to completion in order to assess these results. The "curl-q" term introduced herein affords a very simple and effective way to sustain Lagrangian calculations under such circumstances.

Acknowledgments

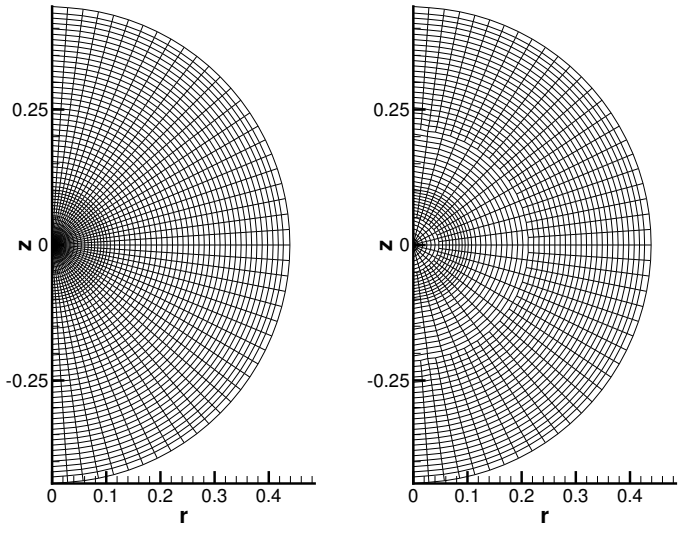
The authors acknowledge M. Shashkov for useful discussions. This work was supported by the Advanced Strategic Computing Initiative (ASCI) of the US Department of Energy.

References

[1] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comp. Phys. 144 (1998) 70–97.
[2] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comp. Phys. 146 (1998) 227–262.
[3] R. Loubère, E.J. Caramana, The force/work differencing of exceptional points in the discrete, compatible formulation of Lagrangian hydrodynamics, LAUR-04-8906, J. Comp. Phys., accepted, doi:10.1016/j.jcp.2005.11.022.
[4] J.C. Campbell, M.J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comp. Phys. 172 (2001) 739–765.
[5] E.J. Caramana, C.L. Rousculp, D.E. Burton, A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional cartesian geometry, J. Comp. Phys. 157 (2000) 89–119.
[6] J.K. Dukowicz, J.A. Meltz, Vorticity errors in multidimensional Lagrangian codes, J. Comp. Phys 99 (1992) 115–134.
[7] W.F. Noh, Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux, J. Comp. Phys. 72 (1987) 78.
[8] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comp. Phys. 142 (1998) 521–561.
[9] R. Loubère, First steps into ALE. INC(ubator) A 2D ALE code on general code on general polygonal mesh for compressible flows – version 1.0.0, Los Alamos report, LAUR-04-8840.

1.4.2 Dealing with exceptional points

The purpose of the work made by E.J. Caramana and I and published in [10] under the title *The Force/Work Differencing of Exceptional Points in the Discrete, Compatible Formulation of Lagrangian Hydrodynamics* is to complete the compatible formulation of Lagrangian hydrodynamics by addressing the questions that arise when treating grid points that must be internally enslaved within the grid to prevent timestep collapse, in other words "exceptional points".



Left : Example of a polar mesh with 51×49 *cells. Right : Radial mesh with three layers of exceptional points.*

The figure to the left presents an example of a three layer of exceptional point polar mesh, one also calls termination lines the mesh lines stopping at an exceptional point. A zone with an exceptional point is also sometimes called a dendritic zone. The reason to use such a mesh is the potential gain in CPU time compared to a full radial mesh : The CFL number is often ruled by the smallest cell length which is located on the triangles at the origin. Roughly speaking with three layers of exceptional points one expects three times bigger CFL number, therefore a three times less expensive computation. It is this decrease in spatial grid stiffness without degradation in solution quality that the procedure proposed in this work is intended to produce.

Indeed dendritic zones could be treated as pentagons by the compatible staggered Lagrangian scheme but the fact that any exceptional point is dynamical often leads to lack of robustness and tendency to unpleasant fatal mesh tangling situations. The work in article [10] focuses on developing a treatment of exceptional points such that collateral damages brought by the existence of exceptional points are literaly (or at least virtualy) absent.

First we have shown that the types of grids that must be used to reduce numerical perturbations about exceptional points involve a restriction of the type of zone to be uniform across the grid. A pentagon must become a quadrilateral. It was also shown that the basic discretization about exceptional, or nondynamical, points could be largely handled by three basic rules that involve already computed subcell masses and forces so that the number of additional operations is small. These involved impedance matching of subcell volumes and masses by donation to nearby dynamical points to which the velocity of the nondynamical exceptional points is enslaved. An appropriate addition of the remaining subcell forces, after median mesh grid adjustment, and their subsequent division and donation to neighboring dynamical points is performed such that force equilibrium is achieved for uniform stress (a necessary sanity check). It was shown that for the internal energy equation to obey momentum conservation about nondynamical points, it is required that the definition of what region in space constitutes a "zone" be generalized. A zone becomes the smallest region in space for which its associated subcell forces sum to zero. Thus, primitive zones that contain common nondynamical points must be glued together in calculating the work from the internal energy equation. In this work the edge-centered artificial viscosity [57] or the tensor artificial viscosity [53]

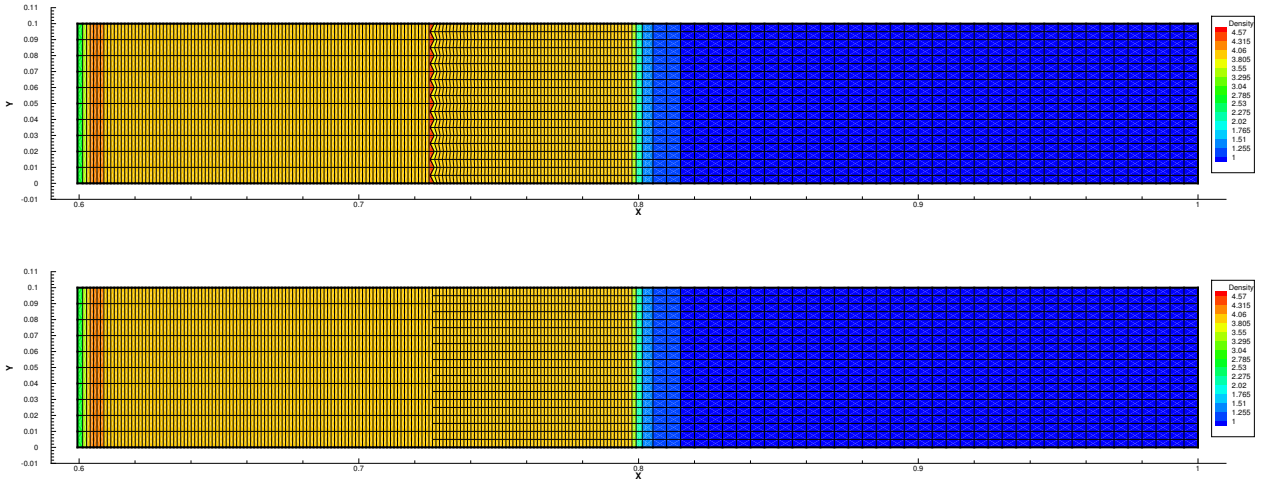


FIGURE 1.10 – *Numerical results from paper [10]. Straight piston with termination lines — (a) Mesh and density at* $t = 0.6$ *original scheme — (b) Mesh and density at* $t = 0.6$ *with the technique proposed.*

are considered and adapted to the presence of exceptional points if needed. Numerical results were shown to validate the procedures given, and to quantify the magnitude of the errors that necessarily occur with the introduction of terminated lines (and their associated nondynamical points). In our work a straight piston is used as a sanity check then the Saltzman problem is simulated with or without exceptional points. Then the spherical Noh problem in axisymmetric geometry is treated as to measure the symmetry error introduced by the presence of exceptional points. Finally the Guderley problem in axisymmetric geometry is run to assess the efficiency of the treatment. Moreover the Guderley problem with false center of convergence is further simulated to show the robustness of our approach when the grid is no more aligned with the flow.
As an illustration we reproduce in Fig. 1.10 the results on a straight piston for the original scheme and the scheme coupled with our donation technique. This sanity check shows how the technique is able to treat termination lines without spurious side effects whereas the original scheme presents an oscillation. In Fig. 1.11 the axisymmetric Guderley problem is simulated (compression of an homogeneous gas initialy at rest due to external velocity boundary condition) on a polar $\Delta r \times \Delta\theta = 51 \times 49$ mesh and on the same effective resolution mesh but with three termination lines. This problem is run to a time $t = 0.8$ at which point the shock wave has reached the true center of convergence, is propagating outward, and has recrossed the inner layer of terminated points. The results are very comparable in quality but the number of timesteps needed to run without and with terminations is about 3000 and 1000.

This paper is reproduced in the following pages.

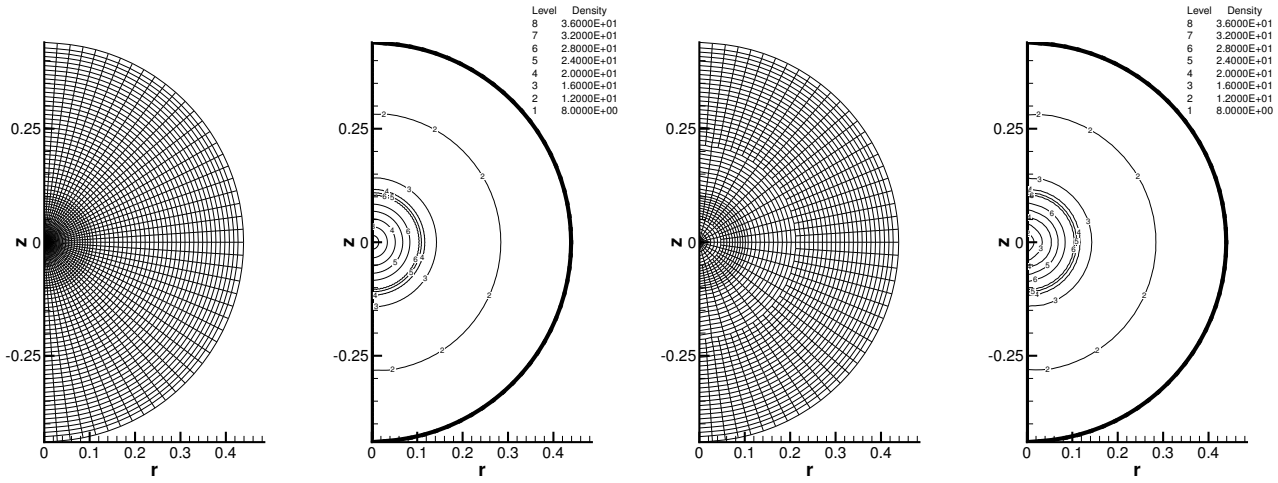


Figure 1.11 – *Numerical results from paper [10]. Guderley problem at $t = 0.8$ — Left : Final mesh and density isolines with no termination layers ($\sim$ 3000 time steps). — Right : Two termination layer final mesh and density isolines ($\sim$ 1000 time steps).*

Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Computational Physics 216 (2006) 1–18
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

The force/work differencing of exceptional points in the discrete, compatible formulation of Lagrangian hydrodynamics

R. Loubère *, E.J. Caramana

T-7 and CCS-2, Los Alamos National Laboratory, Theoretical Division, T7, MS B284, Los Alamos, NM 87545, United States

Received 21 December 2004; received in revised form 8 June 2005; accepted 15 November 2005
Available online 27 December 2005

Abstract

This study presents the force and mass discretization of exceptional points in the compatible formulation of Lagrangian hydrodynamics. It concludes a series of papers that develop various aspects of the theoretical exposition and the operational implementation of this numerical algorithm. Exceptional points are grid points at the termination of lines internal to the computational domain, and where boundary conditions are therefore not applied. These points occur naturally in most applications in order to ameliorate spatial grid anisotropy, and the consequent timestep reduction, that will otherwise arise for grids with highly tapered regions or a center of convergence. They have their velocity enslaved to that of neighboring points in order to prevent large excursions of the numerical solution about them. How this problem is treated is given herein for the aforementioned numerical algorithm such that its salient conservation properties are retained. In doing so the subtle aspects of this algorithm that are due to the interleaving of spatial contours that occur with the use of a spatially-staggered-grid mesh are illuminated. These contours are utilized to define both forces and the work done by them, and are the central construct of this type of finite-volume differencing. Additionally, difficulties that occur due to uncertainties in the specification of the artificial viscosity are explored, and point to the need for further research in this area.
© 2005 Elsevier Inc. All rights reserved.

Keywords: Lagrangian; Hydrodynamics; Energy conserving; High speed flow; Artificial viscosity

1. Introduction

Most physical applications that involve Lagrangian or ALE hydrodynamics calculations employ grids that must be unstructured to some degree to avoid the spatial grid stiffness that would otherwise occur, resulting in an unacceptable decrease in timestep. While the problem of differencing the hydro equations about exceptional, or irregular, points of such grids has been addressed in a previous paper for the case where the exact one-dimensional symmetry limit is desired in curvilinear coordinates [1], the solution presented requires rotations of the force that may not be desirable for general cases, particularly those where low-dimensional symmetry is not important and does not attain to any approximate degree in the solution. However, the general discussion given therein of the timestep vicissitudes of such grids remains valid and is not repeated here. An example of the type of grid previously considered is shown in Fig. 1. Here the terminated lines, and associated exceptional points, are displayed as hollow circles that are placed at the midpoints of straight lines connecting neighboring regular points. This is opposed to being placed on a common radius as previously depicted in [1]. It is in general necessary to enslave the terminated points shown in Fig. 1 to prevent large unphysical perturbations in the solution from occurring about them when disturbances such as shock waves propagate in any direction across them. Thus the force and mass discretizations that are suitable for regular zones that do not contain these points must be appropriately modified where they are present in the grid. This requires a careful investigation of the discretization properties of the underlying hydro algorithm that elucidates all assumptions, transparent or hidden. To this end, Section 2 gives a brief review of the hydro algorithm that forms the title of this paper. This is, however, a new and concise presentation that displays essential features that complement previous expositions [2,3].

In Section 3 the types of grids encountered are briefly detailed as well as the basic constraints that are enforced when both mass and force from the exceptional points are "donated" to neighbors to which their velocity is enslaved, making the former "nondynamical". Section 4 gives numerical results that validate the discretization rules introduced in Section 3, and quantifies the magnitude of errors that necessarily occur when strong shock waves encounter these points. In particular, special attention is paid to the artificial viscosity forces as they are velocity dependent; and since velocity interpolation is utilized, these forces can result in sensitivities that are difficult to counter by any general prescriptions. Last, a discussion of this work and its principal conclusions is given.

* Corresponding author. Tel.: +1 505 667 1407.
E-mail addresses: loubere@lanl.gov (R. Loubère), ejc@lanl.gov (E.J. Caramana).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.11.022

2. Discrete, compatible Lagrangian hydrodynamics

The discrete, compatible formulation of Lagrangian hydrodynamics [3] essentially modernizes older forms of Lagrangian hydro [4]. It places this type of numerical algorithm into a simple and consistent framework where conservation of total energy plays the central role, but where the principal dependent variables remain density, velocity, and specific internal energy. Like all the older versions of Lagrangian hydrodynamics it employs a staggered grid in space with velocity and position carried on points "p", and density, specific internal energy, and stress centered in zones "z". However, both zones and points are considered to be surrounded by interleaved volumes circumscribed by lines in 2D (or surfaces in 3D) that are termed the "coordinate-line" and "median" meshes, respectively, as shown in Fig. 2. This interleaved topology allows for a simple finite-volume calculation of forces. These act from zones that carry a mass M_z, and onto points that carry a mass M_p, where "z" and "p" are integer indices that range over all zones and points, respectively. Auxiliary quantities denoted as "corner" masses and forces are introduced; these are unique and common to both a zone and a given point of that zone, and thus carry both the zone and the point indices. The zone and point masses and the total force acting on a point are then constructed from these more primitive entities as simple sums. The corner mass is denoted as m_z^p or m_p^z, and the corner force as $\vec{f}_z^{\,p}$ or $\vec{f}_p^{\,z}$, where the lower index denotes that which

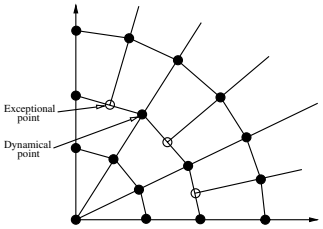


Fig. 1. Typical mesh with exceptional points. Exceptional points are enslaved to adjacent dynamical points on an angular grid line.

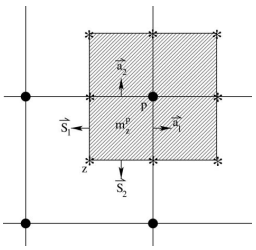


Fig. 2. Coordinate-line mesh thru solid dots; median mesh thru stared points: subcell geometrical vectors $\vec{a}_1, \vec{a}_2, \vec{S}_1, \vec{S}_2$ associated with zone "z"; m_z^p is corner mass.

is summed, but they are otherwise equal. The corner forces mediate the exchange of kinetic and internal energy from points to zones, or vice-versa.

The essentials of this algorithm can be presented in a very succinct form that displays both its power and simplicity in the following manner: consider the change in the kinetic energy of a point "p" between two time levels "n" and "$n+1$" as given by $\Delta K_p \equiv M_p[(\vec{v}_p^{\,n+1})^2 - (\vec{v}_p^{\,n})^2]/2$ in a timestep Δt. Then note that the difference in the squares of the point velocity, $\vec{v}_p$, between the two time levels can be uniquely factored to yield

$$\Delta K_p = M_p \Delta\vec{v}_p \cdot \Delta\vec{r}_p/\Delta t, \tag{2.1}$$

where $\Delta\vec{v}_p \equiv (\vec{v}_p^{\,n+1} - \vec{v}_p^{\,n})$, and more importantly the change in the position vector, $\vec{r}_p^{\,n}$, is $\Delta\vec{r}_p \equiv (\vec{v}_p^{\,n+1} + \vec{v}_p^{\,n})\Delta t/2$. That is, the reason the coordinates are advanced as $\vec{r}_p^{\,n+1} = \vec{r}_p^{\,n} + \Delta\vec{r}_p$ with the just specified form for $\Delta\vec{r}_p$ is not to obtain second order accuracy in time (other forms may give higher order), but because of the manner in which the kinetic energy difference between two time levels factors! Defining the change in internal energy in a zone in a time Δt as $M_z\Delta e_z$, where $\Delta e_z \equiv (e_z^{n+1} - e_z^n)$, and summing the change of kinetic energy over all points "p", and the change in internal energy over all zones "z", the change in total energy on a timestep, ΔE, can be written as

$$\Delta E \equiv \sum_p M_p \Delta\vec{v}_p \cdot \Delta\vec{r}_p/\Delta t + \sum_z M_z \Delta e_z = \Delta W_{\rm bd}, \tag{2.2}$$

where $\Delta W_{\rm bd}$ is the boundary work performed on the particular timestep. Next the momentum and specific internal energy equations are "postulated" to have the discrete forms

$$M_p \Delta\vec{v}_p = \sum_z \vec{f}_z^{\,p} \Delta t, \tag{2.3}$$

where the sum of the arbitrary corner force is over all zones that contain point "p", and

$$M_z \Delta e_z = -\sum_p \vec{f}_p^{\,z} \cdot \Delta\vec{r}_p, \tag{2.4}$$

where this sum is over all points that circumscribe zone "z". Note that what has been done is to define the total force that acts on a point "p" as the sum of all corner forces that act from zones adjacent to that point; and also, to define the work done into a zone "z" as the negative sum of the corner forces of that zone dotted into their respective point displacements on the given timestep. For example, for the case of pressure forces this sum is a discrete form of $-P_z\Delta V_z$ work, where P_z is the zone pressure and ΔV_z is the change in zone volume in a timestep.

Writing the momentum equation in the form given by Eq. (2.3) may appear to be simply a useful division of the force acting on a point into single contributions from surrounding zones. However, that the work performed by a zone can be written as Eq. (2.4) utilizing the same corner forces such that this work becomes a simple sum of corner force times point displacement is more significant, as this guarantees exact conservation of total energy for forces of completely arbitrary origin. Note that the force of gravity exchanges kinetic energy with gravitational potential energy and thus does not enter into the corner force; but rather, enters directly as an additional term on the RHS of the momentum equation, and with the kinetic energy of a point augmented by its gravitational potential in the total energy tally. Likewise, Eq. (2.4) can contain an additional source term on the RHS that originates from the direct deposition of internal energy from chemical or other sources.

To demonstrate energy conservation suppose that the momentum equation, Eq. (2.3), is dotted into the point displacement on a timestep, $\Delta\vec{r}_p$, and its RHS inserted into the first sum in Eq. (2.2). Then the RHS of Eq. (2.4) is likewise inserted into the second sum of Eq. (2.2). This results in

$$\Delta W_{\rm bd} = \sum_p \sum_z \vec{f}_z^{\,p} \cdot \Delta\vec{r}_p - \sum_z \sum_p \vec{f}_p^{\,z} \cdot \Delta\vec{r}_p, = \sum_{p={\rm bd}} \sum_{z={\rm ext}} \vec{f}_z^{\,p} \cdot \Delta\vec{r}_p. \tag{2.5}$$

Since the corner force is unique to a given zone and point of that zone (recall that $\vec{f}_z^{\,p} = \vec{f}_p^{\,z}$), the second sum from the internal energy equation completely cancels with terms from the first sum due to the momentum equation. There remains only terms that sum over the boundary points "$p = {\rm bd}$" from the external zones "$z = {\rm ext}$" that lie adjacent to these points that form the prescribed boundary; these terms define the boundary work performed on the given timestep. Thus total energy is conserved exactly in that when integrated in time Eq. (2.2) yields $E^n = E^{n=0} + W_{\rm bd}^n$, where $E^{n=0}$ is the initial total energy and $W_{\rm bd}^n$ is the sum of the boundary work as given by Eq. (2.5) over all timesteps to the final time "t"'. Total energy is also conserved locally in that the internal domain of interest can consist of any collection of zones, or just a single zone. This latter case is illustrated in Fig. 3 where a single 2D quadrilateral zone "z" is shown with its associated corner forces "$\vec{f}_i^{\,z}$", and where "$i = 1\ldots4$" ranges over its defining points. For this case $E^n = M_z e_z^n + \sum_{i=1}^4 M_i(\vec{v}_i^{\,n})^2/2$. Defining "$\vec{F}_i$" as the sum of all corner forces that act on point "$p = i$" in a timestep, it follows from Eq. (2.5) that the boundary work performed on a timestep is $\Delta W_{\rm bd} = \sum_{i=1}^4 (\vec{F}_i - \vec{f}_i^{\,z}) \cdot \Delta\vec{r}_i$. This is the total exterior work performed with respect to any single zone. If there is no mass exterior to the boundary, the sum of all zone masses and the sum of all point masses are equal (a consistency requirement), but otherwise this is not true and is not necessary for total energy to be conserved.

As previously noted [3], the discrete form of the equations for total energy, momentum, and internal energy when written as Eqs. (2.2)–(2.4) constitute an algebraic identity; given any two, the third uniquely follows. This identity employs three totally arbitrary entities referred to as the zone mass, M_z, the point mass, M_p, and the corner force ($\vec{f}_z^{\,p}$ or $\vec{f}_p^{\,z}$). It is the proper specification of these otherwise abstract objects that gives this system physical meaning; however, whatever their specification the quantity defined above as "total energy" is always exactly conserved. The program of the rest of the development of the discrete, compatible formulation

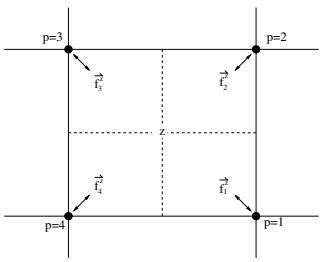


Fig. 3. Single quadrilateral zone "z" with its defining points p_1, p_2, p_3, p_4, and associated corner forces $\vec{f}_p^{\,z}$, $1 \leqslant i \leqslant 4$.

of Lagrangian hydrodynamics is to specify these three objects. The word "discrete" is to be emphasized in that these equations are considered to describe the time evolution of small fluid volumes, as opposed to a differencing of the equations of continuum fluid dynamics.

The nodal and zonal masses, M_p and M_z, are straightforwardly constructed by introducing the corner mass m_z^p or m_p^z, where the zone mass is simply the sum of all m_p^z's over points about the specified zone "z"; and likewise, the point mass M_p is the sum of all m_z^p's over all zones adjacent to point "p" [3]. The median mesh is hinged by auxiliary points, shown as asterisks in Fig. 2, that are determined by time independent interpolation coefficients (usually 1/2, 1/4, ...) applied to neighboring dynamical points. That these coefficients be time independent is necessary to prevent destructive nonlinear feedback between the median and coordinate-line meshes. The specification of the corner force is much more involved, and is given for regular zones in both 2D and 3D for all force contributions: mean zone stress, artificial viscosity, and anti-hourglass forces in a series of previously published papers [3,5–9]. Although this corner force specification is given as a form of finite volume differencing, other discretizations can be utilized to define the corner forces. It is the purpose of this work to extend the finite volume definitions to the aforementioned case of exceptional points that preserve not only conservation of total energy, but also the conservation properties of Eqs. (2.2)–(2.4) that the original corner force specifications enforce. The nature of these are briefly detailed.

Principal amongst the additional conservation laws is the conservation of linear momentum, which can be stated most succinctly as the homogeneity of space (referred to as Noether's theorems [10]). This adheres for our discrete equations as follows: suppose that a constant but arbitrary displacement vector, "$\vec{r}_0$", is added to the coordinates of all points in space by performing a Galilean boost, $\vec{v}_0$, such that $\vec{r}_0 = \vec{v}_0 \Delta t$ during any given timestep. Then for the discrete form of the internal energy equation, Eq. (2.4), to remain unchanged it is required that

$$\sum_z \vec{f}_z^{\,p} \cdot \vec{r}_0 = \vec{r}_0 \cdot \sum_z \vec{f}_z^{\,p} = 0. \tag{2.6}$$

Thus the requirement of linear momentum conservation is just the statement that all corner forces in a zone must sum to zero. For zones with a constant stress throughout the entire zone, this means that the zone volume is closed and simply connected so that the sum of its surface area vectors equal zero. (Note that for 1D cylindrical or spherical geometry this requirement does not hold since these zone volumes, as formed by two concentric circles or spheres, are not simply connected.) Conservation of angular momentum, as the isotropy of space, follows similarly as requiring $\sum_p \vec{r}_p \times \vec{f}_p^{\,z} = 0$ in each zone, where $\vec{r}_p$ is the position vector to a point "p" from any arbitrary origin of the coordinate system. All corner forces are required to obey conservation of linear momentum on a single zone basis when cast into Cartesian geometry. This property is central to our treatment of exceptional points; however, not all force contributions obey angular momentum conservation on a single zone basis if they arise from stresses that are not constant throughout a given zone. This is true for both the subzone-pressure anti-hourglass forces [6], and the edge-centered artificial viscosity forces [5]. However, studies have shown that global breaking of angular momentum conservation remains at small truncation error levels for these force contributions. Finally, for the stated system of equations to give useful results they must be time advanced in a numerically stable manner. This is achieved by a predictor-corrector step where the predictor and corrector stages are identical except that the corner forces are centered at the "n" time level on the predictor, and at the "$n+1/2$" time level on the corrector, resulting in second order accuracy in time and numerical stability subject to the usual CFL constraint [1].

3. Finite-volume mass/force/work computation

The basic program of this work is to give a general prescription of how to perform the force and mass discretizations about exceptional points of terminated lines as shown in Fig. 1. However, this may not be for the simple terminations shown; instead these may consist of a line of such points in two or in three dimensions. It is not our purpose to detail all possible cases, but instead, to give some general rules that enable one to adapt the basic underlying differencing to any situation, and to illustrate the general method and show its effectiveness for certain relevant cases. The central idea is to treat all points as though they are regular, then the exceptional points have their associated corner forces and masses appropriately "donated" to points to which their

3. Conservation of linear momentum as defined by Eq. (2.6) must be maintained for all zones. This generalizes the definition of the term "zone".

The consequences of these three requirements are now examined.

The basic idea behind the first of our three statements is readily seen from Fig. 4(b). To make the corner volume and corner mass on the right side of the two dynamical points "p_1" and "p_2" match that from the large zone on the left of these points, one must let $m_{p_1}^{z_1} \to m_{p_1}^{z_1} + m_{p_e}^{z_1}$ and $m_{p_2}^{z_2} \to m_{p_2}^{z_2} + m_{p_e}^{z_2}$. Next one sets both $m_{p_e}^{z_1}$ and $m_{p_e}^{z_2}$ equal to zero, so that the nondynamical point "p_e" carries no mass; thus it does not contribute kinetic energy to the total energy tally.

The requirement of equilibrium for uniform stress demands that the two corner forces that act on point "p_e" from zones "z_1" and "z_2" first be summed, then one-half of this sum is donated to points "p_1" and "p_2". That these corner forces must be summed before being donated can be seen from the fact that for a uniform pressure "P" in zones "z_1", "z_2" and "z_0", $\vec{f}_{p_e}^{z_1} = P(\vec{a}_1 + \vec{a}_3)$ and $\vec{f}_{p_e}^{z_2} = P(\vec{a}_2 - \vec{a}_3)$. (Here the grid vectors $\vec{a}_i$ are oriented as shown in Fig. 4(c) and have magnitudes of the length of the half-edges to which they are normal.) By adding these two corner forces the term associated with the grid vector $\vec{a}_3$ cancels. Thus when this sum is donated to the dynamical points "p_1" and "p_2" with simple factors of one-half to each, force equilibrium is achieved when taking into account the corner force contribution from the large zone, "z_0", that lies to the left side of these points. That is, we define $\vec{F}_{p_e} \equiv \vec{f}_{p_e}^{z_1} + \vec{f}_{p_e}^{z_2}$ and then let $\vec{f}_{p_1}^{z_1} \to \vec{f}_{p_1}^{z_1} + \vec{F}_{p_e}/2$, $\vec{f}_{p_2}^{z_2} \to \vec{f}_{p_2}^{z_2} + \vec{F}_{p_e}/2$. Last, we set $\vec{f}_{p_e}^{z_1}$ and $\vec{f}_{p_e}^{z_2}$ to zero and can proceed with zone sweeps as before to assemble the total force on all points.

The sum of the original corner forces add to zero individually in the two rectangular zones of Fig. 4(b) that contain the nondynamical point "p_e". However, because of the sum and then division performed to satisfy the requirement of force equilibrium for uniform pressure, zones "z_1" and "z_2" must now be considered as a single composite zone when the internal energy equation, Eq. (2.4), is solved. That is, the sum of the corner forces of both of these zones still sum to zero, but no longer on a single zone basis. Thus the change of specific internal energy, Δe_z, due to the corner forces acting in these two zones must be considered as a single sum. This is because by donating the force from the nondynamical point, the grid vectors that make up the force contours sum to zero only over the composite region "$z_1 + z_2$". If there are no other energy sources in Eq. (2.4), and if both regions "z_1 and "z_2" have equal specific internal energies at time zero, then this equality is maintained for later time. However, even in this case the pressure in these two zones need not be equal since their zone densities can still have different values.

The velocity enslavement of nondynamical points always utilizes a simple linear interpolation to neighboring dynamical points. This serves to keep the position of these points closely synchronized to dynamical ones preventing large spatial excursions that would otherwise occur. Thus for the case shown in Fig. 4(b), $\vec{v}_{p_e} = (\vec{v}_{p_1} + \vec{v}_{p_2})/2$, keeping the enslaved point "p_e" at the midpoint of the line connecting points "p_1" and "p_2".

The procedure just outlined can be extended to two or more terminated lines with associated nondynamical points between two dynamical points, to which their velocities are enslaved by linear interpolation, in a completely analogous manner. The large zone adjacent to the terminated lines is again made into a quadrilateral, and all its corner forces are computed the same as for any other quad. Corner volumes are donated to the two dynamical points such that corner volume matching is achieved with respect to the large zone. Corner forces are summed on the terminated line side and then donated to the dynamical points with coefficients that are the same as used to donate the nodal volume/mass. (That is, a single nondynamical point may now contribute all of its nodal mass and force to a single dynamical point, instead of one-half to each.) Likewise in 3D, although the situation may be more complicated to visualize, the manner in which the nondynamical points of terminated lines are treated is not fundamentally different than in 2D; the three principles given above are applied to all situations using a linear interpolation of the velocity from the dynamical to the nondynamical points.

The artificial viscosity forces that appear in these equations must be given special consideration as these forces are velocity dependent, and thus do not compose about nondynamical points in the same manner as the total stress and anti-hourglass forces that are completely accounted for by the above set of rules. However, in the calculations shown in the next section they are treated the same as all other forces since the basic algorithm consists of assembling the corner force from all the various contributions and then manipulating the

velocity is enslaved. One must examine the character of the individual contributions to the corner force to ascertain that they still behave in a physically meaningful manner. First, the kind of zone topologies encountered with the exceptional points of terminated lines must be considered.

Two types of median mesh configurations that can occur in 2D geometry for the exceptional point "p_e" are shown in Fig. 4. If this point is considered to be a defining point of all zones, as is the case in Fig. 4(a), then the large zone on the left is a degenerate pentagon; whereas in Fig. 4(b), this point belongs to the smaller rectangles on the right but is not considered a defining point for the large zone on the left, which remains a quadrilateral. Since the lines that comprise the median mesh connect midpoints of the coordinate lines to the zone center point, the subzone corner volumes appear skewed to one side if the exceptional (termed "nondynamical" from hereon) point is considered part of the larger zone. Thus the light gray region in Fig. 4(a) denotes the three corner masses that are associated with point "p_e" if it were to remain dynamical. Our first alteration from the usual definitions used to construct the median mesh is to erase the nondynamical point from being associated with the large zone on the left in Fig. 4. All corner forces are thus constructed from the mesh shown in Fig. 4(b). The nondynamical point "p_e" now obtains mass only from the two small corner volumes of the rectangular zones of Fig. 4(b) shown as the light gray region, and no mass from the larger zone to its left that no longer identifies it as a part of that zone; the same statement holds for the corner forces. Next, three basic principles are enunciated that determine how to difference about any one or any set of nondynamical points:

1. Corner masses are donated from nondynamical points such that, aside from convergence effects, corner volumes are matched in size about dynamical points that neighbor nondynamical ones.
2. Corner forces are donated from nondynamical points to neighboring dynamical points such that force equilibrium is achieved for uniform stress in a region about these points.

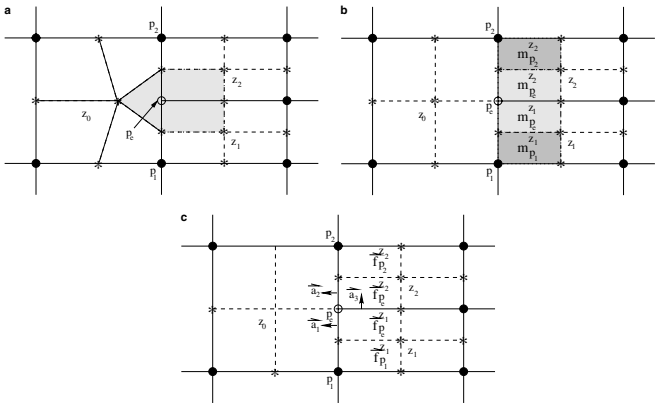


Fig. 4. (a) Exceptional point p_e, shown as "○", is considered as a dynamical point for the large zone z_0 and the smaller zones z_1, z_2; Subcells of point p_e are light gray regions. The median mesh is plotted with dashed lines, the coordinate-line mesh with solid lines. (b) Exceptional point p_e considered a dynamical point for only for the smaller zones z_1, z_2. The subcell associated with p_e in zone z_0 no longer exists. The dark gray subcells are the ones to which the remaining masses/forces of point p_e are donated, m_p^z is the mass of subcell associated with zone z and point p. (c) Coordinate-line mesh area vectors $\vec{a}_i$, $1 \leqslant i \leqslant 3$, associated with the subcell force donations and subcell forces $\vec{f}_{p_e}^{z}$.

corner force as previously described. Two different forms of artificial viscosity are utilized: an edge-centered form [5] and a tensor form [9]. The edge-centered form is simple and intuitively based, and works particularly well when the fluid flow is close to grid-aligned; lower-dimensional symmetry properties are automatically captured. However, it contains no information about the zone topology. In contrast, the tensor form connects all points of a zone and gives better results when the fluid flow is far from being aligned with the grid; however, for highly elongated zones it can connect disparate zone length scales in an unphysical manner. The tensor form of the artificial viscosity is unchanged in its calculation in zones that contain nondynamical points. The edge-centered artificial viscosity is modified from the form given in [5] as follows: first, the density and sound speed used to compute this viscosity is taken to be the zone values, rather than from point-centered averages. This is done for all zones to obtain consistency across the mesh. Any change in the functional form of the artificial viscosity within the computational domain will itself result in errors for a steady-state shock even if there are no terminated lines. Second, the limiter function used with the viscosity calculated on the edge that connects the nondynamical point to that with a larger major radius, as shown in Fig. 1, is calculated as an average of that of the two limiters adjacent to it and oriented in the same direction. Thus for grid aligned flow in the radial direction in Fig. 1 these edges all have the same value for the limiter function. Other concerns about the artificial viscosity forces are discussed after presenting the numerical results that highlight difficulties that can occur with their use.

Finally, for completeness the trivial case of degenerate points is mentioned. Degenerate points occur at a center of convergence in 2D or 3D, or along the z-axis of a polar grid in 3D, where a set of points that are initially coincident are required to move together. In this case one simply adds the individually separate nodal masses and forces associated with these points for use in updating the common point velocity from the momentum equation. The change in specific internal energy of these zones need not be averaged as they are still separately Galilean invariant; however, this averaging can result in more robust results and is performed for the edge-centered artificial viscosity. An example of this is shown later with the Guderley test problem [11] utilizing a mesh with a center of convergence that is displaced along the z-axis, and moves when it encounters a converging shock wave.

4. Numerical results

The numerical results given next are of two types: sanity checks that gauge the effect of our donation procedure for two driven piston problems (a straight piston and the so-called Saltzman piston [12]), and more difficult tests utilizing the well-known Noh [13] and Guderley problems. The latter problems assess the size of errors in 1D symmetry, and the general robustness of the method in addition to its ability to keep the timestep from decreasing precipitiously. The results are computed with a 2D, unstructured grid code developed in the group T-7 at LANL [14,15]. A simple ideal gas law, $\gamma = 5/3$, equation of state is employed to compute pressures. An advection-type limiter is utilized to restrict the magnitude of both forms of artificial viscosity that are employed, edge-centered [5] and tensor [9], unless otherwise indicated.

We begin with a straight piston problem whose initial grid is shown in Fig. 5. The grid is on a domain of unit horizontal length and with a width of 0.1 in the ignorable, vertical direction with 101 equally spaced vertical lines and 21 equally spaced horizontal lines, one-half of which terminate at the middle of the domain. Thus the zones in the coarser region on the left half of the figure have one to one aspect ratio, and those in the finer region on the right half have 2:1 aspect ratio, elongated in the vertical direction. The initial density

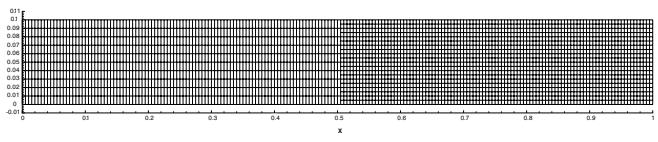


Fig. 5. Straight piston with termination lines – initial mesh.

is unity, with zero initial specific internal energy; the left piston boundary moves to the right with unit velocity driving a shock wave into cold media and across the terminated lines. Results showing the grid and density isolines at time $t = 0.6$ using the edge-centered viscosity are given in Fig. 6 parts (a) and (b): part (a) shows the result obtained with no modifications about the terminated lines, and part (b) presents results obtained if the mass and force donation procedure is performed. In the first instance the perturbations caused by the shock wave crossing the layer of terminated lines is clearly visible at the shock front that has moved far to the right of this layer. In part (b) these perturbations are totally absent; there are also no residual density changes at the termination layer. That is, the presence of terminated lines causes no perturbations to the solution, which is the same as though these lines were totally absent. With the use of the tensor artificial viscosity similar results are also seen; perturbations in the density are somewhat smaller than those in Fig. 6(a), but are still clearly visible; they are removed by the donation procedure. No subzone pressure forces are utilized, as these are unnecessary for this type of problem.

Next the initial grid shown in Fig. 5 is given the standard Saltzman perturbation to obtain the one displayed as Fig. 7. This problem is rerun with subzone pressure forces using a merit factor, $M_f = 1.0$. In Fig. 8 results are given at time $t = 0.6$ for four cases: parts (a) and (b) show the grid and density contours without, and with, modification for points at the termination layer, respectively, using the edge-centered artificial viscosity. These two figures are almost identical, and thus the donation procedure did not result in an improved solution but it also did not degrade it. In parts (c) and (d) are shown the results using the tensor artificial viscosity without, and with, modification for points at the termination layer. In this case the mass and force donations at the

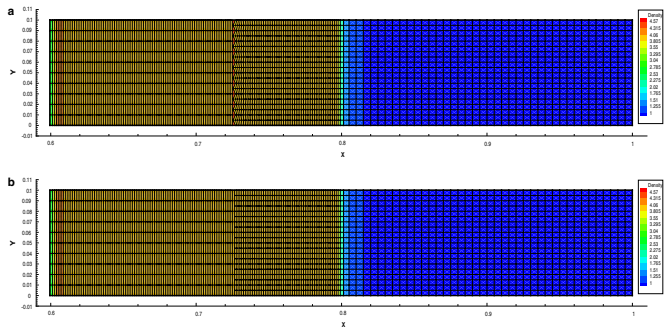


Fig. 6. Straight piston with termination lines for the edge viscosity – (a) mesh and density at $t = 0.6$ original code; (b) mesh and density at $t = 0.6$ with donation.

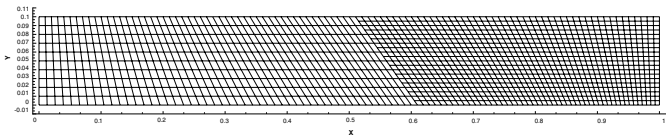


Fig. 7. Saltzman piston with termination lines – initial mesh.

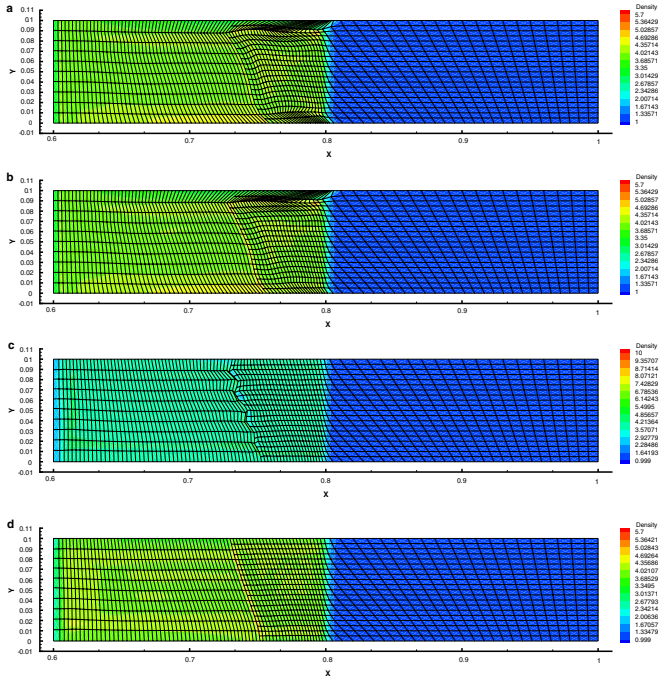


Fig. 8. Saltzman piston with terminated lines – mesh and density at $t = 0.6$: (a) edge viscosity original code, (b) edge viscosity with donation, (c) tensor viscosity original code, (d) tensor viscosity with donation – note different scales due to different behaviors.

termination layer result in a very substantial improvement. The shock front nearly breaks apart downstream of the termination layer and is close to failure without these modifications, but the results seen with them are better than those with the edge-centered viscosity. This problem also shows the necessity for averaging the work done on a timestep in the zones that share a common nondynamical point, which is a consequence of our third criterion that guarantees linear momentum conservation. If this averaging procedure is not carried out, negative specific internal energy is encountered in some of these zones and the calculation ends. It is only the sum of the work done into zones with a common nondynamical point that is positive.

The spherical Noh problem [13] is next considered in cylindrical geometry using area-weight differencing on an angular grid with one layer of terminated lines. The initial conditions are unit density, zero specific internal energy, and an inward radial velocity of magnitude 1.0. A spherical stagnation shock wave is generated and

travels outward from the center. Subzone pressure forces are turned off because they are not needed for this problem when run without terminated lines. The timestep is increased by a factor of two as expected with the single termination layer; for a high resolution simulation multiple termination layers are necessary. The simulation is performed to the point where the shock reaches the location of the first termination line at $t = 0.07866$. The grid at this time is shown in Fig. 9: parts (a) and (b) are without, and with, mass and force donation, respectively, using the edge-centered viscosity. A very large sawtooth-like oscillation occurs at the termination layer. This is largely, but not totally, removed by the enslavement procedure. Without subzone pressure forces the calculation in part (a) will not continue much longer; with them the grid distortion is somewhat reduced but not eliminated. In parts (c) and (d) are given the corresponding results for the tensor viscosity without, and with, the donation procedure, respectively. The grid without donation in part (c) is comparable in quality to that in part (b) for the edge-centered viscosity with donation. The symmetry errors in the density along the termination layer are detailed in Table 1. The donation procedure reduces these errors

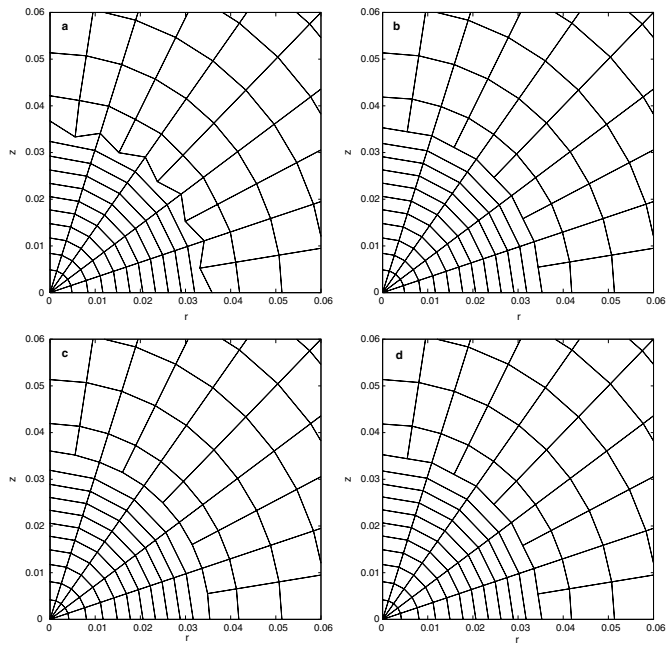


Fig. 9. Noh problem at $t = 0.07866$, the shock has just passed the termination line – final meshes (zoom): (a) edge viscosity, (b) edge viscosity with donation process, (c) tensor viscosity, (d) tensor viscosity with donation process.

Table 1
Noh problem: symmetry errors for edge and tensor viscosities along termination layer

$(\rho_{max} - \rho_{min})/\rho_{max}$	Original code	With donation
Edge viscosity	3.1×10^{-2}	4.0×10^{-4}
Tensor viscosity	4.5×10^{-2}	1.2×10^{-4}

to truncation levels, but recall that for no termination layer and area-weight differencing they are always at roundoff error level.

The last test problem considered is the driven spherical impulsion problem of Guderley [11]. This has an initial density of unity, zero specific internal energy, and a velocity boundary condition given in [6] using area-weight differencing in cylindrical geometry. Results are shown with two layers of terminated points, and for comparison without terminated lines, for two types of grid setups: first, with 51 equally-spaced radial and 51 equally-spaced angular divisions on a one-half circular domain of unit initial radius, with a center of convergence at ($r = 0$, $z = 0$); and second, the same total domain but with circular lines that have their centers moved along the z-axis such that the center of convergence is displaced to the point ($r = 0$, $z = 0.5$). In both cases the velocity boundary condition is the same, and the outer boundary remains a circle with center at ($r = 0$, $z = 0$). Subzone pressure forces with merit factor unity are utilized for both the centered and off-axis center of convergence cases. This problem is run to a time $t = 0.8$ at which point the shock wave has reached the true center of convergence, is propagating outward, and has recrossed the inner layer of terminated points. The results are seen in Fig. 10 where the top two parts show the grid and density isolines for no radially terminated layers. The bottom two parts give these same results with two terminated layers; the edge-centered artificial viscosity and the donation procedure at the termination layers are utilized. The results are very comparable in quality. The number of timesteps needed to run without and with terminations is about 3000 and 1000, respectively. It is this decrease in spatial grid stiffness without degradation in solution quality that the donation procedure is intended to produce. The tensor form of the artificial viscosity produces results of similar quality.

Finally, Fig. 11 shows the initial grid for the Guderley problem with the displaced center of convergence described previously, both with two layers of termination lines and without such points. Results are given in Fig. 12 utilizing the edge-centered viscosity for the case without, and with, termination layers; both the grid and the density isolines are shown at the final time of $t = 0.8$. Likewise, similar results are given for the tensor viscosity at the same final time in Fig. 13. The density bulge that results from the shock wave that is reflected from the true center of convergence of this problem at ($r = 0$, $z = 0$) is quite visible in both figures. With the edge-centered viscosity and no terminated lines it took about 9000 timesteps to run to completion, whereas with the two termination layers this was reduced to roughly 2000; for the tensor viscosity these values are approximately 6300 and 1700, respectively. Thus our donation procedure is effective in this regard. However, in order to make these displaced center of convergence cases run to completion the limiters were turned off on both forms of the artificial viscosity. Additionally, the velocity of points on the z-axis were enslaved to those points on the next radial line angularly outward (for area-weight differencing this causes no energy errors since these points carry zero mass [3].), rather than being moved dynamically as they are for the nondisplaced center of convergence case shown in Fig. 9. Usually simulations terminate due to very small timesteps just after the shock wave intersects the false center of convergence without this set of alterations. This can be changed somewhat depending on the value of the merit factor that multiplies the subzone pressure forces. Generally the tensor form of the artificial viscosity shows more robustness and better quality of results for the case of very nongrid-aligned flow than the edge-centered form, which agrees with the results presented in [9]. This rather unsatisfactory state of affairs is due to the lack of a consistent form of the artificial viscosity that is optimal for all simulations. This topic is addressed next.

59

4.1. Artificial viscosity forces

The artificial viscosity forces that are necessary to resolve shocks in the otherwise dissipationless Euler equations that are solved herein have been the subject of much discussion and controversy since they were introduced many years ago [16]. While there is still no universal agreement on the precise functional form for these forces, they can usually be viewed as variations on the basic form expressed in [17] as

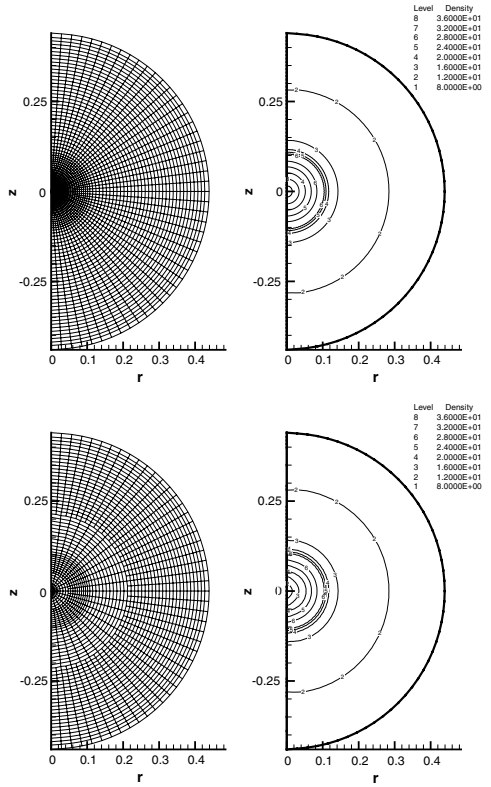


Fig. 10. Guderley problem at $t = 0.8$ with edge viscosity. Top: final mesh and density isolines with no termination layers; Bottom: two termination layer final mesh and density isolines.

$$q_z = \rho[c_s\Delta v + (\Delta v)^2], \tag{4.1}$$

where "q_z" acts as an addition to the scalar pressure of zone "z", with the additional requirement that the zone be under compression for the forces to be dissipative. The factor "ρ" is a measure of density, "c_s" is a sound

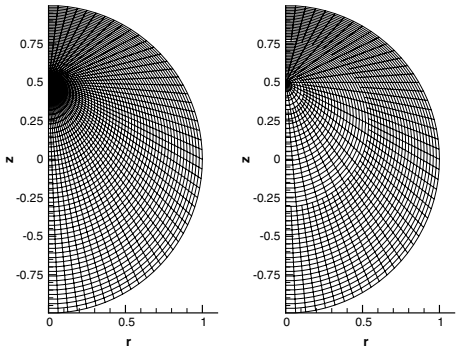


Fig. 11. Initial meshes for Guderley problem with displaced grid center of convergence. Left: without terminations (51×51) circular and radial lines; Right: two termination layers.

speed, and "Δv" is a jump in the magnitude of the velocity about a zone; all of these can be calculated in more than one manner. The two terms on the RHS of Eq. (4.1) are referred to as the linear and nonlinear artificial viscosity, respectively.

Ideally one wishes to solve the Euler equations by integrating along the characteristics of this hyperbolic system. However, because this is both too difficult, and too expensive to compute, one resorts to shock capturing methods by inserting an artificial viscosity term that regularizes the discontinuities that otherwise occur. This term may be inserted directly as done herein, or as the consequence of solving a local Riemann problem, with much the same effect. With the former approach, more recent forms of the artificial viscosity term that give much improved results over older versions, are developed by specifying conditions under which the characteristics of the Euler equations do not cross, and thus where dissipation is not needed. This results in the inclusion of a limiter function that augments the compression switch in the basic artificial viscosity, Eq. (4.1). The limiter has the form of that used for advection except that the "upwind" direction of material advection is unimportant, and also, it should not limit the magnitude of these forces so much that oscillations occur behind a strong shock. The purpose of the limiter in this context is to turn off the artificial viscosity for adiabatic compression, or along a front of constant phase where the edge-projected velocity field is a linear function of the coordinates, and thus where the characteristics of the hyperbolic system do not cross. However, for general flow that is nongrid-aligned there is no theory that indicates how this limiter should be constructed. It can simply result in a decreased value of these forces that degrades overall robustness of the calculations, as occurs with the above simulations.

For the terminated line situation considered in this paper, there is a more fundamental difficulty when the artificial viscosity is computed using the velocity of a nondynamical point that has been set by linear interpolation. As is seen from Eq. (4.1), a linear interpolation of the velocity from two dynamical points to one nondynamical point results in a decrease in the strength of the linear viscosity term by one-half, and the nonlinear term by one-quarter. This situation gets geometrically worse for the case of multiple terminations. Furthermore, the limiter function defined along these edges is invalid since a limiter calculated from a linearly interpolated velocity will always turn completely off. The zone-centered tensor artificial viscosity also exhibits this problem, although in some averaged sense that is more difficult to quantify. One can separately reconstruct the edge-centered viscosity without using the points that have interpolated velocities. This is the

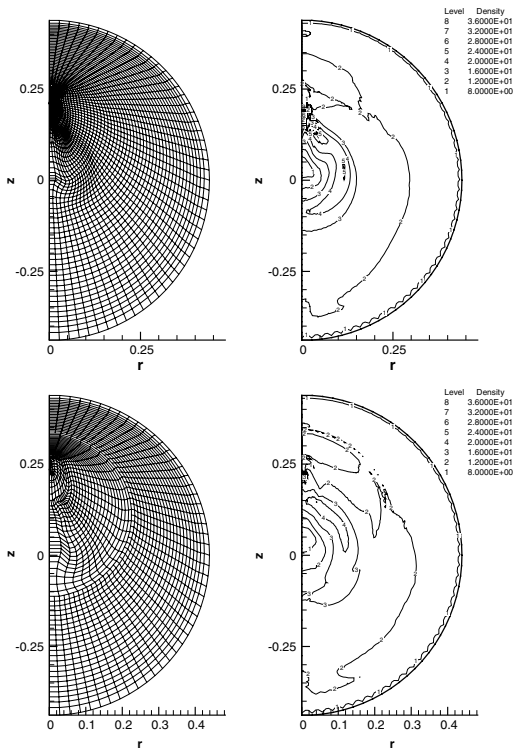


Fig. 12. Guderley displaced center problem at $t = 0.8$ with the edge viscosity. Top: final mesh and density isolines; Bottom: two termination layers, final mesh and density isolines – with donation.

ideal solution, but one that is expensive computationally. The treatment of the artificial viscosity is found to be problematical in that changes in zones that contain nondynamical points may be mirrored as errors (severe zone volume collapse) in regular neighboring zones that do not contain these modifications. There is no known "optimal" form of the artificial viscosity, and the inclusion of this term into the Euler equations is always fraught with difficulties.

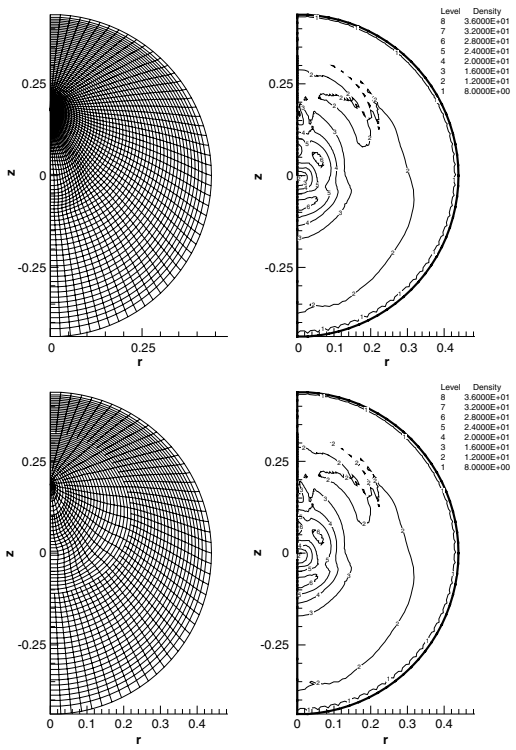


Fig. 13. Guderley displaced center problem at $t = 0.8$ with the tensor viscosity. Top: final mesh and density isolines; Bottom: two termination layers, final mesh and density isolines – with donation.

5. Discussion and conclusions

The purpose of this work is twofold: first, to complete the compatible formulation of Lagrangian hydrodynamics by addressing the remaining finite-volume discretization questions that arise when treating grid points that must be internally enslaved within the grid to prevent timestep collapse; and second, to address

certain more general questions concerning the basic structure and implementation of this numerical algorithm. That these issues are related is seen in Section 3. There it is shown that the types of grids that must be used to reduce numerical perturbations about exceptional points involve a restriction of the type of zone to be uniform across the grid. This is seen in going from Fig. 4(a) to (b) where a pentagon becomes a quadrilateral. A compelling reason for using a restricted set of zone types is that the central quantity that must be computed to implement this algorithm is the corner force. If one knows that the number of corners of all the zones is the same, then sweeps over all zones, combined with sweeps over the known number of internal zone edges and corners, allows the corner force to be assembled with computer code that is simple to construct, and which executes in an efficient manner.

It was shown that the basic discretization about exceptional, or nondynamical, points could be largely handled by three basic rules that involve already computed corner masses and forces so that the number of additional operations is small. These involved impedance matching of corner volumes and masses by donation to nearby dynamical points to which the velocity of the nondynamical points is enslaved. An appropriate addition of the remaining corner forces, after median mesh grid adjustment, and their subsequent division and donation to neighboring dynamical points is performed such that force equilibrium is achieved for uniform stress (a necessary sanity check). It was shown that for the internal energy equation to obey momentum conservation about nondynamical points, it is required that the definition of what region in space constitutes a "zone" be generalized. A zone becomes the smallest region in space for which its associated corner forces sum to zero. Thus, primitive zones that contain common nondynamical points must be lumped together in calculating the work term in the internal energy equation. Numerical results were shown to validate the procedures given, and to quantify the magnitude of the errors that necessarily occur with the introduction of terminated lines and their associated nondynamical points. Artificial viscosity forces are always the most crucial component of any shock-wave hydro algorithm. There is still no universally satisfactory form of the artificial viscosity suitable for all problems. The fact that different forms for this force are often utilized depending on the type of problem being studied is the major remaining deficiency of this class of hydrodynamics methods.

Finally, the discrete, compatible formulation of Lagrangian hydrodynamics was developed in a manner that explicitly displays the nature of this algorithm to be that of an algebraic identity. This identity consists of two arbitrary scalars, the zone and point masses, and one arbitrary vector, the corner force, such that the usual definition of total energy conservation is always exact. This algorithm thus reveals itself to be a true tautology in the sense described in [18]. As such, it describes a priori truth that cannot be confuted, since in primitive form it makes no assertion about any physical system. As remarked in [18], tautologies are neither trivial nor useless, but embody the kernel from which physical systems of truth may be constructed. Thus the quality with which the discrete, compatible formulation of Lagrangian hydrodynamics may describe certain physical situations is mostly, if not entirely, dependent on the quality of the specification of the three abstract quantities that compose it.

Acknowledgements

All numerical simulations were performed using ALE INC [standing for ALE INC(ubator)], a 2D Arbitrary-Lagrangian–Eulerian code that utilizes general polygonal grids, and which was developed in the Mathematical Modelling and Analysis Group (T-7) at LANL [14,15]. The authors wish to thank M. Shashkov for fruitful discussions, and the referees for their helpful comments. This work was supported by the Advanced Strategic Computing program of the US Department of Energy.

References

[1] E.J. Caramana, Timestep relaxation with symmetry preservation on high aspect-ratio angular or tapered grids, J. Comput. Phys. 166 (2001) 173–185.
[2] D.E. Burton, Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity, in: Advances in the Free Lagrange Method, Springer Verlag, New York, 1990.
[3] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1998) 227–262.
[4] B. Alder, S. Fernbach, M. Rotenberg (Eds.), Methods in Computational Physics, vol. 3, 1964.

[5] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (1998) 70–97.
[6] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of lagrangian subzonal masses and pressures, J. Comput. Phys. 142 (1998) 521–561.
[7] E.J. Caramana, P.P. Whalen, Numerical preservation of symmetry properties of continuum problems, J. Comput. Phys. 141 (1998) 174–198.
[8] E.J. Caramana, C.L. Rousculp, D.E. Burton, A compatible, energy and symmetry preserving lagrangian hydrodynamics algorithm in three-dimensional cartesian geometry, J. Comput. Phys. 157 (2000) 89–119.
[9] J.C. Campbell, M.J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys. 172 (2001) 739–765.
[10] C. Lanczos, The Variational Principles of Mechanics, fourth ed., Dover Publications, 1986, p. 384.
[11] R. Lazarus, Self-similar solutions for converging shocks and collapsing cavities, SIAM J. Numer. Anal. 18 (1981) 316.
[12] J.K. Dukowicz, B. Meltz, Vorticity errors in multidimensional Lagrangian codes, J. Comput. Phys. 99 (1992) 115.
[13] W.F. Noh, Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux, J. Comput. Phys. 72 (1987) 78.
[14] J. Campbell, M. Shashkov, A compatible lagrangian hydrodynamics algorithm for unstructured grids, Selcuk J. Appl. Math. 4 (2003) 53–70, Report version can be found at <http://cnls.lanl.gov/~shashkov>.
[15] R. Loubere, First steps into ALE INC(ubator). A 2D ARBITRARY-LAGRANGIAN-EULERIAN CODE ON GENERAL POLYGONAL MESH FOR COMPRESSIBLE FLOWS Version 1.0.0, Los Alamos National Laboratory Report, LA-UR-04-8840.
[16] J. VonNeumann, R.D. Richtmyer, A method for the calculation of hydrodynamic shocks, J. Appl. Phys. 21 (1950) 232–237.
[17] M.L. Wilkins, Use of artificial viscosity in multidimensional shock wave problems, J. Comput. Phys. 36 (1980) 281–303.
[18] A.J. Ayers, Language, Truth, and Logic, Dover Publications, 1952, pp. 84–87.

1.4.3 Slide-lines

When developing a simulation code based on a Lagrangian scheme at one point we can have to face situations for which slide-lines are required. Many hydrodynamical problems involve shear flows along material interfaces. If the materials move along each other but are tied to a single Lagrangian computational mesh without any sliding treatment, severe mesh distortions appear which can eventually cause the failure of the simulation.

With M. Kucharik, R. Liska and L. Bednarik at CTU in Prague (Czech Republic) we have implemented a slide-line treatment into the PALE code (Prague ALE) based on the compatible staggered Lagrangian scheme on quadrilateral meshes. Starting with the paper of E.J. Caramana [80] we have published paper [25] entitled *Enhancement of Lagrangian slide lines as a combined force and velocity boundary condition*.

In this work we first review the 2D approach described by E.J. Caramana in [80] and suggest two enhancements - interpolated interaction instead of a simple one-to-one point interaction described in the previous article, and a numerical surface tension model improving the stability of the interface. Both improvements stabilize the slide line and lead to more realistic results, as shown on selected numerical examples such as pure sliding, Saltzman piston, sliding rings, explosion with sliding and a bullet in a channel. In Fig. 1.12 we reproduce the sanity check of two columns of gas sliding on each other. The meshes are of different size. Nonetheless the sliding is perfectly reproduced. The same pure sliding test is further run with two annulii. In Fig. 1.13 one reproduces the results obtained by the original approach of E.J. Caramana *vs* our improved technique.

This paper is reproduced in the following pages.

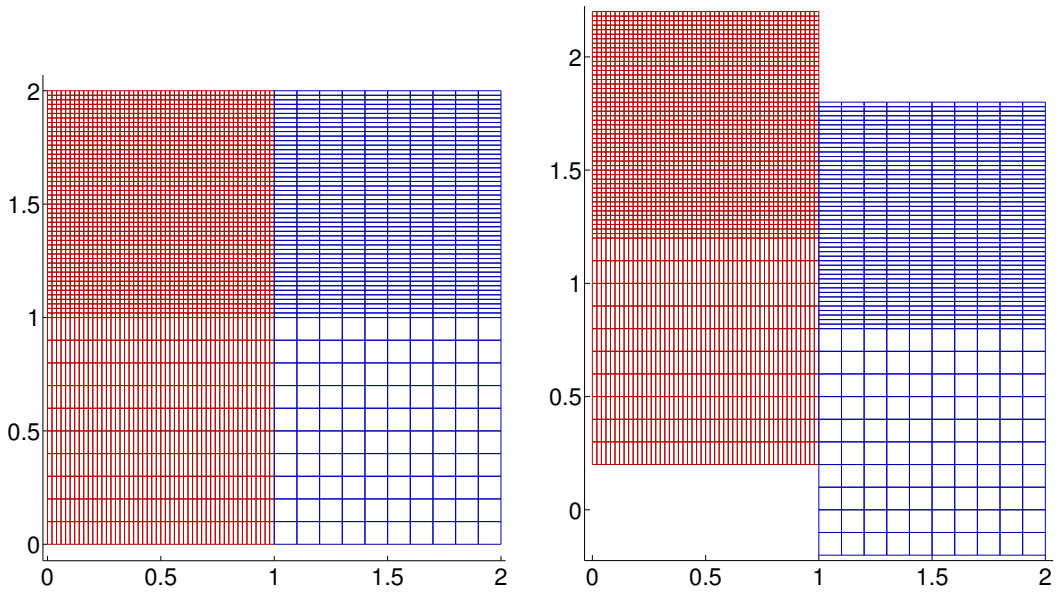


FIGURE 1.12 – *Numerical results from paper [8o]. Initial and final* 40×50 *and* 10×50 *meshes of the pure sliding sanity check. Two vertical blocks of fluid (Left and Right) meshed with non-uniform grids are sliding with velocity* $\boldsymbol{U}_L = (0, +v)$ *and* $\boldsymbol{U}_R = (0, -v)$, $v = 1$ *in our test. The vertical slide line is initiated at* $x = 1$ *and must remain vertical during the sliding.*

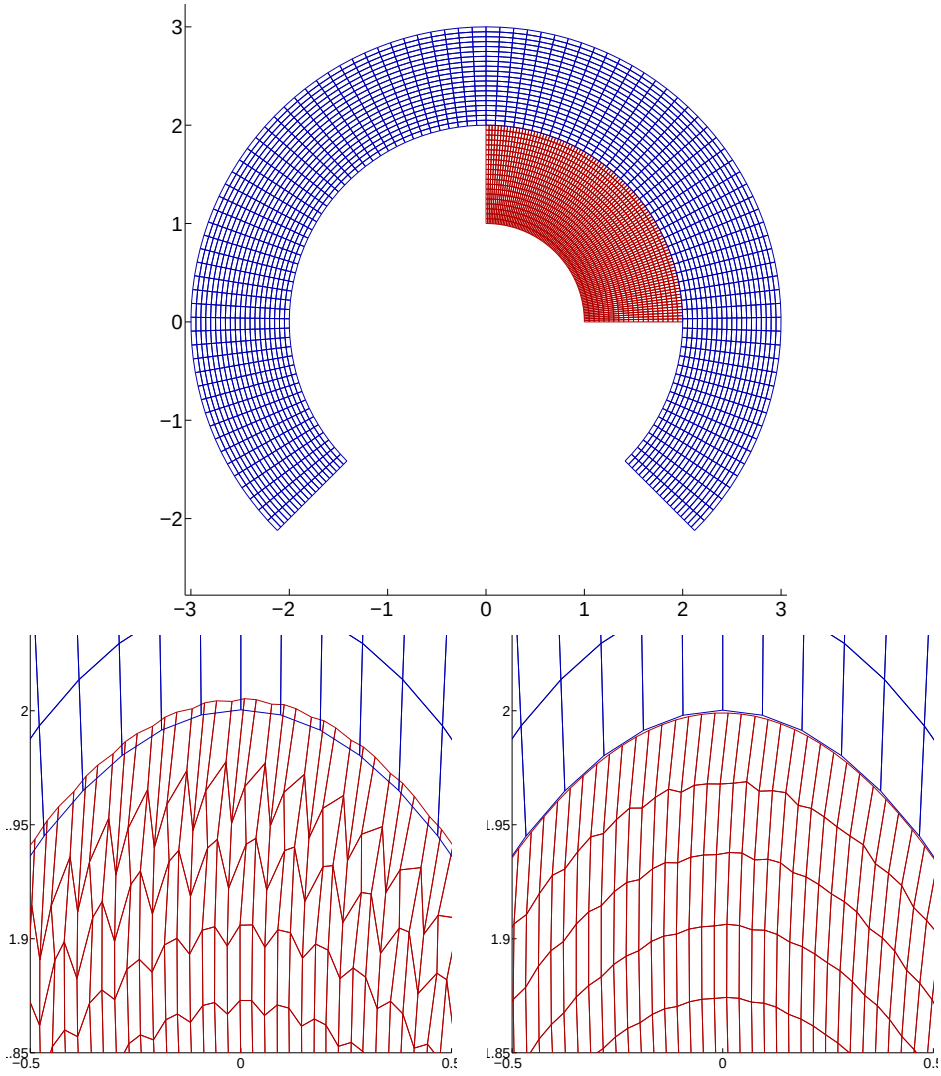


Figure 1.13 – *Numerical results from paper [80]. Top : Initial meshes for sliding rings problem separated by a circular slide line of radius $R = 2$. Bottom : Zoom to the interesting part along the slide line (aspect ratio not preserved). Comparison of computational meshes for the original approach (left) and our improved method (right) proposed in [80].*

Lagrangian Slide Lines as a Combined Force and Velocity Boundary Condition

M. Kucharík[a,*], R. Loubère[b], L. Bednárik[a], R. Liska[a]

[a] FNSPE, Czech Technical University in Prague, Břehová 7, Praha 1, 115 19, Czech Republic
[b] Institut de Mathématiques de Toulouse, CNRS, Université de Toulouse, France

Abstract

Many hydrodynamical problems involve shear flows along material interfaces. If the materials move along each other but are tied to a single Lagrangian computational mesh without any sliding treatment, severe mesh distortions appear which can eventually cause the failure of the simulation. This problem is usually treated by introducing the sliding line framework into the Lagrangian code. In this paper, we revise the 2D approach described in the article *The implementation of slide lines as a combined force and velocity boundary condition*, E. J. Caramana, Journal of Computational Physics, 228, (2009), and suggest two enhancements – interpolated interaction instead of a simple one-to-one interaction described in the previous article, and a numerical surface tension model improving the stability of the interface. Both improvements stabilize the slide line and lead to more realistic results, as shown on selected numerical examples.

Keywords: Lagrangian hydrodynamics, slide lines, staggered scheme

1. Introduction

In realistic physical simulations, people often face problems of shear flows at material interfaces. If the materials move along each other but are tied to a single computational mesh without any sliding treatment, severe distortions appear which can eventually cause the failure of the simulation. A typical example is the motion of a laser produced plasma in a deforming channel [?], or a shear flows in a high-velocity impact problem [?]. One option to solve this problem is the introduction of a slide line environment into the Lagrangian code.

The demand for a slide line treatment arose when first realistic Lagrangian simulations became attractive. Generally, all sliding algorithms can be classified into two groups [? ?]. The first group in which the overall forces between the sides of the slide line are computed, includes popular methods of Lagrangian multipliers [? ?] or the penalty method [?]. We focus here on the second type of methods, where quantities are mapped across the slide line and the nodes are treated in a similar manner as the internal nodes. The introduction of slide lines is an old but fruitful idea that dates back to Wilkins [?] as a chapter in a book (reproduced in chapter 5 of [?]). In this approach, the interaction of both sides of the slide line is explicitly calculated, and their inter-penetration is prevented by an explicit put-back-on step. This approach is very popular and is used for slide line treatment in many hydrodynamic codes. There exist many modifications of this approach, see for example [? ? ?]. For an overview of the slide line algorithms see the specialized report [?] or the classical more general paper [?].

Another possibility to treat sliding lines in a compatible staggered Lagrangian code [?] is a special type of boundary condition for nodal forces and velocities. Suppose that there exist two different meshes interacting with each other through a common sliding line, one of them is specified as the master side defining the slide line shape, while the other – slave – side follows the slide line. We keep the main idea of [?] and incorporate the contact forces for the nodes on the slide line, representing the forces due to the pressure gradient across the interface. The second main ingredient of the sliding line treatment is the correction preventing the inter-penetration of two sides. This is the main difference between the two approaches. In [?], the inter-penetration is prevented by an explicit put-back-on step in which the slave nodes are artificially moved back onto the master slide line edges. Instead of this artificial nodal motion, the velocity correction is used [?]. The correction is simpler and compatible with the rest of the Lagrangian solver.

In this paper, we describe the method from [?] with details, some of which are eluded in the original paper. In [?], the nodes on the slide line exclusively interact with one mesh node from the other side of the slide line, and this interaction can cause severe distortions of the interface. We suggest here the improvement in which the interaction is performed in an interpolated sense. Moreover, as the shear flow (sliding) is present here, the original method can result in interface disturbances due to evolving (real or numerical) Kelvin-Helmholtz instabilities. To stabilize the interface, we suggest here a type of numerical surface tension model preventing the instability to appear. We skip all issues related to void opening or closing, which may be very important for the practical computations, however this topic will be treated in future investigations.

The rest of the paper is organized as follows. In Section 2, we briefly describe the staggered Lagrangian hydrodynamics which we use to derive and test the slide line framework. In Section 3, the slide line treatment from [?] is reviewed and explained in detail. In Section 4, we suggest the interpolated interaction, changing the communication between the nodes on both sides of the slide line from one-to-one to interpolated. In Section 5, another improvement is suggested, the numerical surface tension, eliminating evolution of the Kelvin-Helmholtz instability due to the shear flow along the interface. The behavior of the improved method and its comparison with the original method on selected numerical tests is performed in Section 6. Finally, the whole paper is concluded in Section 7.

2. Staggered Lagrangian Hydrodynamics

In the Lagrangian framework, the gas dynamics equations can be written in the following form,

$$\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) - \nabla\cdot\vec{v} = 0\,, \tag{1}$$

$$\rho \frac{d}{dt}\vec{v} + \nabla P = \vec{0}\,, \tag{2}$$

$$\rho \frac{d}{dt}\varepsilon + P\nabla\cdot\vec{v} = 0\,, \tag{3}$$

where ρ is the fluid density, $\vec{v}$ is the velocity, ε the specific internal energy, and $\frac{d}{dt}$ denotes the total Lagrangian time derivative. The first equation expresses the volume conservation equation, whereas the second and third ones are the momentum and total energy conservation equations. Volume conservation equation is often referred to as the Geometric Conservation Law (GCL). The previous system is supplemented by a thermodynamics closure (equation of state – EOS), $P = P(\rho,\varepsilon)$. Often, the ideal gas equation of state is used, $p = (\gamma-1)\,\rho\,\varepsilon$, where γ is the ratio of specific heats. The last equations are the trajectory equations

$$\frac{d\vec{x}}{dt} = \vec{v}\,(\vec{x}(t),t)\;, \tag{4}$$

expressing the Lagrangian motion of any point located at position $\vec{x}$.

We use a staggered placement of variables in which position and velocity are defined at mesh nodes while thermodynamic variables (density, pressure, energy) are located at cell centers. We denote the computational zone (cell) by the symbol z, mesh points (nodes) are denoted by the symbol p. Following the compatible discretization from [?], the mass of zone z is connected with the mass of the adjacent point p by the notion of the subzonal masses, m_{zp},

$$m_z = \sum_{p\in P(z)} m_{zp}\,, \qquad m_p = \sum_{z\in Z(p)} m_{zp}\,, \tag{5}$$

*E-mail: kucharik@newton.fjfi.cvut.cz

where $P(z)$ stands for the set of points in zone z, and $Z(p)$ stands for the set of zones adjacent to point p. The subzones are assumed to be Lagrangian particles, which implies that the subzonal masses are initialized at the beginning of the simulation, $m_{zp} = \rho_{zp}\,V_{zp}$ and remain constant in time, implying that cell and nodal masses remain constant either.

The discrete trajectory equation for mesh nodes can be written as

$$\frac{d\vec{x}_p}{dt} = \vec{v}_p\,, \tag{6}$$

which together with the assumption of constant cell mass guarantees satisfaction of the GCL equation (1). The momentum equation (2) can be written in the semi-discrete form for a particular mesh point p,

$$m_p\frac{d\vec{v}_p}{dt} = \vec{F}_p\,, \tag{7}$$

where the Newton force is computed by evaluating the pressure gradient in the dual cell Ω_p corresponding to node p,

$$\vec{F}_p = -\int_{\Omega_p}\vec{\nabla}P\,dV = -\sum_{z\in Z(p)}\int_{\Omega_{zp}}\vec{\nabla}P\,dV = \sum_{z\in Z(p)}\vec{F}_{zp}\,, \tag{8}$$

where the subzonal forces $\vec{F}_{zp}$ are evaluated by transforming the volume integrals to boundary integrals using the Green divergence theorem. Finally, following [?], the semi-discrete form of the energy equation (3) can be written as

$$m_z\frac{d\varepsilon_z}{dt} = -\sum_{p\in P(z)}\vec{F}_{zp}\cdot\vec{v}_p\,, \tag{9}$$

which is derived from the total energy conservation on a cell-by-cell basis. This approach is supplemented by additional forces representing artificial viscosity [?] and anti-hourglass forces [?], stabilizing the solution and preventing the simulation from failure. The update of fluid specific internal energy and velocity come from the finite difference discretization of the equations (7) and (9), new nodal positions from equation (6), new fluid density is obtained by dividing the constant cell mass by the new cell volume, and new pressure from the EOS. For a full description of the compatible Lagrangian scheme including more details, such as timestep control of predictor-corrector time integration scheme, see [? ? ?].

3. Slide Line Treatment

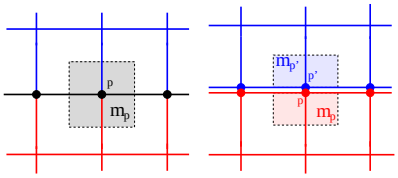
Quoting Caramana [?]: "Slide lines (2D) and surfaces (3D) are a way to treat interfaces in Lagrangian hydrocodes that allow different materials or regions to move relative to each other without the grid distortion that would otherwise terminate these calculations quickly." The presence of fluid instabilities is often, if not always, a cause of failure for Lagrangian codes. The idea of slide lines comes from [?], and was reproduced in Chapter 5 of [?]. By all means most of slide line treatments follow this original work of Wilkins as pointed by Caramana [?]. Slide line is an important feature although it is quite rarely described.

Any method treating slide lines in a Lagrangian hydrocode must ideally respect some general requirements. On the first hand some requirements are related to the properties of the physical underlying system and the Lagrangian numerical scheme used to solve it. As instance conservation of mass, momentum and total energy obeyed by the physical system and the numerical scheme must be also preserved by the specific slide line treatment. The normal acceleration between two materials being continuous, the slide line treatment is required to preserve such continuity. In addition the slide line treatment must not destabilize steady solutions (as instance homogeneous fluids with constant pressure and velocity field). Over and above ideal situations must be perfectly dealt with independently of the possibly different mesh sizes across the slide line. More precisely fake slide lines must have an ignorable impact, straight slide line must remain so in pure sliding situations. On the other hand some requirements are related to the physics involved at the slide line. As instance two materials sliding on each other never inter-penetrate, consequently the slide line treatment should also obey this principle. Furthermore the resulting contact force between two sliding materials must only act in the normal direction to the slide line. In addition, in the frictionless case, the tangential net force due to the contact must be zero.

Developing a slide line algorithm demands to, exactly or approximately, fulfill these requirements. The technique developed in [?] and described in section 3.1 first determines scaled contact forces normal to the slide line. Apart from momentum conservation (for non-ideal cases) and inter-penetration problem the technique fulfills the previous requirements. Contact forces are determined by trying to perfectly deal with ideal situations. Then, to ensure that no penetration occurs, the point velocity on one side of the slide line is further corrected in section 3.2. Unfortunately, doing so, the technique slightly losses total energy conservation. Nonetheless such energy discrepancy is used to assess the meaningfulness of the computation, see section 3.3.

3.1. Contact Forces

The first task is to determine the contact force for the ideal situation as shown in Figure 1 where a computational mesh is split in two parts separated by a single slide line splitting each point on the slide line into two half-points p and p'. Here the design principle consists of exactly retrieving the momentum



(a) single point (b) split in two points

Figure 1: (a) One point in a mesh separated in its lower (red) and upper (blue) part by imaginary slide line (black line). One particular point p is shown with its dual cell containing the corresponding nodal volume/mass m_p (gray rectangle). (b) Point separated to two half-points p and p', belonging to different sides of the slide line.

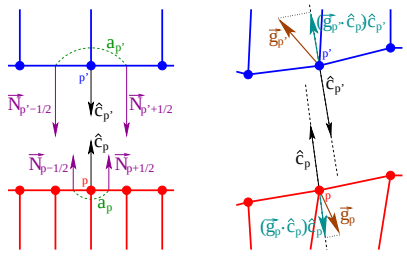
equation (7) for one point when the slide line splits (7) in two equations for two half-points,

$$m_p\frac{d\,\vec{v}_p}{d\,t} = \vec{F}_p + \vec{g}_p\,, \qquad m_{p'}\frac{d\,\vec{v}_{p'}}{d\,t} = \vec{F}_{p'} + \vec{g}_{p'}\,, \tag{10}$$

where $\vec{F}_p$ and $\vec{F}_{p'}$ represent the sum of subzonal forces from lower/upper subzones attached to the half-points and their sum is equal to the original total nodal force, and similarly with the masses of the half-points m_p and $m_{p'}$. Moreover, $\vec{g}_p/\vec{g}_{p'}$ represent the contact forces acting on the lower/upper half-point from the opposite side of the slide line. When we assume that p and p' coincide and have the same mass and geometry, the separate equations (10) have to sum to the original equation (7) which implies $\vec{g}_{p'} = -\vec{g}_p$. After substituting this back to (10) and assuming the same acceleration for both half-points, the contact forces become

$$\vec{g}_p = -\vec{g}_{p'} = \frac{m_p\,\vec{F}_{p'} - m_{p'}\,\vec{F}_p}{m_p + m_{p'}}\,. \tag{11}$$

Let us move to a less ideal situation shown in Figure 2 (a) where both sides of the slide line remain straight, however, the cell aspect ratios are different. In this case one defines the outer normals for each edge



(a) scaling (b) projection

Figure 2: (a) Definition of edge outer normals $\vec{N}_{p\pm1/2}$, nodal unit normals $\hat{c}_p$, and nodal characteristic lengths a_p. (b) Projection of contact forces $\vec{g}_p$ to the normal direction a_p.

on a slide line, $\vec{N}$, which have the normal directions, and their size is defined by the length of the edge. The nodal outer normal is defined (as suggested in [?]) by the average of the adjacent edge normals,

$$\vec{N}_p = (\vec{N}_{p-\frac{1}{2}} + \vec{N}_{p+\frac{1}{2}})/2\,, \tag{12}$$

where $p-1/2$ and $p+1/2$ denote the left and right edges in the slide line with respect to point p. The characteristic size a_p of node p is defined as $a_p = \|\vec{N}_p\|$ which corresponds to the length between the edge centers of the two adjacent edges. The nodal unit normal is defined as $\hat{c}_p = \vec{N}_p/a_p$.

Because two upper cells sharing one single node p' interact with possibly more than two lower cells we need to "impedance" match the force interactions across the slide line. (In other words an interaction area $a_{p'}$ of point p' must be matched by an approximately equal area of interaction from the opposite side no matter how many points p from this opposite side contribute.)
When p and p' coincide as in Figure 2 (a) it is somewhat enlightening to consider the "meta-cell" constituted of some part of the upper cells sharing point p' and the the lower cells associated to the interaction area a_p. In this meta-cell the mass of point $p \equiv p'$ is approximately given by $m_p + m_{p'}\frac{a_p}{a_{p'}}$ because, if one assumes that $m_{p'}/a_{p'}$ represents the density of the mass along lower boundary cells, then $m_{p'}\frac{a_p}{a_{p'}}$ does represent the approximate mass related to the interaction area a_p. In the same way the acceleration of $p \equiv p'$ in the meta-cell may be approximated by $\left(\vec{F}_p + \vec{F}_{p'}\frac{a_p}{a_{p'}}\right) / \left(m_p + m_{p'}\frac{a_p}{a_{p'}}\right)$.
The half-sided momentum equations are given by

$$m_p\,\frac{d\,\vec{v}_p}{d\,t} = \vec{F}_p + \vec{G}_p\,, \qquad m_{p'}\,\frac{d\,\vec{v}_{p'}}{d\,t} = \vec{F}_{p'} + \vec{G}_{p'}\,, \tag{13}$$

where $\vec{G}_p = (\vec{g}_p \cdot \hat{c}_p)\ \hat{c}_p$ and $\vec{G}_{p'} = (\vec{g}_{p'} \cdot \hat{c}_{p'})\ \hat{c}_{p'}$ are the normal component of the contact forces. The contact force is then defined as the normal component of the acceleration of the point in the meta-cell, that is to say $\vec{G}_p$ force is determined as the solution of

$$\frac{d\,\vec{v}_p}{d\,t}\cdot\hat{c}_p \;=\; \frac{\vec{F}_p + \vec{G}_p}{m_p}\cdot\hat{c}_p \equiv \frac{\vec{F}_p + \vec{F}_{p'}\frac{a_p}{a_{p'}}}{m_p + m_{p'}\frac{a_p}{a_{p'}}}\cdot\hat{c}_p, \tag{14}$$

leading to $\vec{g}_p \cdot \hat{c}_p = \frac{m_p(\vec{F}_{p'}\cdot\hat{c}_p) - m_{p'}(\vec{F}_p\cdot\hat{c}_p)}{a_{p'}m_p + a_p m_{p'}} a_p$. We further approximate the projected force by the projection to the normal from the opposite side, $\vec{F}_{p'}\cdot\hat{c}_p \approx -\vec{F}_{p'}\cdot\hat{c}_{p'}$ to get

$$\vec{g}_p \cdot \hat{c}_p \approx -\frac{m_p\left(\vec{F}_{p'}\cdot\hat{c}_{p'}\right) + m_{p'}\left(\vec{F}_p\cdot\hat{c}_p\right)}{a_{p'}m_p + a_p m_{p'}} a_p. \tag{15}$$

As a consequence $\vec{G}_p$ (and $\vec{G}_{p'}$ following the same logic) is determined by

$$\vec{G}_p = (\vec{g}_p\cdot\hat{c}_p)\ \hat{c}_p \approx -\frac{m_p\left(\vec{F}_{p'}\cdot\hat{c}_{p'}\right) + m_{p'}\left(\vec{F}_p\cdot\hat{c}_p\right)}{a_{p'}\,m_p + a_p\,m_{p'}}\,a_p\,\hat{c}_p\,, \tag{16}$$

$$\vec{G}_{p'} = (\vec{g}_{p'}\cdot\hat{c}_{p'})\ \hat{c}_{p'} \approx -\frac{m_{p'}\left(\vec{F}_p\cdot\hat{c}_p\right) + m_p\left(\vec{F}_{p'}\cdot\hat{c}_{p'}\right)}{a_p\,m_{p'} + a_{p'}\,m_p}\,a_{p'}\,\hat{c}_{p'}\,. \tag{17}$$

As we can see, both formulas are the same up to a prime sign again. The projection is demonstrated in Figure 2 (b). When the points p and p' do not coincide then the meta-cell construction is only an approximation, so are the point mass and its acceleration.

By construction two of the previously listed requirements, namely the contact force only acting in the normal direction and the frictionless requirement, are obeyed. Moreover the normal acceleration between the two materials is continuous because $\frac{1}{m_p}(\vec{F}_p + \vec{G}_p)\cdot\hat{c}_p = -\frac{1}{m_{p'}}(\vec{F}_{p'} + \vec{G}_{p'})\cdot\hat{c}_{p'}$. While the momentum conservation is not ensured to round-off error for non-ideal situations it is still conserved to truncation error when summed over all of the points of the slide line[1]. On the other hand total energy conservation is preserved because the slide line treatment is so far expressed as contact forces which are further participating in the internal energy update (9) relying on the compatible construction of the numerical scheme. Furthermore we have obtained equations (16-17) by requiring the slide line algorithm to retrieve ideal situations; Areas a are used to scale the G quantities so that we obtain the "exact" result given by (11) in the limit of exactly the same number of aligned grid points on both sides of the slide line. Lastly we can verify that the slide line treatment maintains a constant pressure even when non-uniform meshes are used, see also the numerical experiments on pure sliding situations in 6.1.
Unfortunately the slide line treatment does not prevent the two sides to inter-penetrate. The contact force is constructed to avoid such an unlikely situation to occur, however, there is no intrinsic mechanism that can prevent it. The next section presents the velocity correction technique to prevent inter-penetration from [?].

3.2. Velocity Correction Preventing Inter-penetration

Up to now, the slide line has been treated in a symmetric manner concerning the upper and lower sides. However, classical treatment of slide line defines a "master" and "slave" side. In general, the materials on both sides dictate such dichotomy. Without loss of generality, let us assume that the lower (red) side in our Figures is declared as slave. Because inter-penetration can occur the slave side of the slide line is forced to follow the master side – the so-called "put-back-on step" puts back any point from the slave side onto the master side [?].

The last ingredient proposed by Caramana [?] relates to this put-back-on step. In fact, the solution of the inter-penetration problem is recast into a velocity boundary condition. More precisely, the update of velocity of slide points coming from (13) is given by

$$\vec{v}_p^{\,n+1,\dagger} = \vec{v}_p^{\,n} + \frac{\Delta t}{m_p}\left(\vec{F}_p + \vec{G}_p\right), \qquad \vec{v}_{p'}^{\,n+1,\dagger} = \vec{v}_{p'}^{\,n} + \frac{\Delta t}{m_{p'}}\left(\vec{F}_{p'} + \vec{G}_{p'}\right), \tag{18}$$

[1] Indeed let us remark that in the case the normals are collinear and opposite, $\hat{c}_p = -\hat{c}_{p'}$, the sum of $\vec{G}_p/a_p$ and $\vec{G}_{p'}/a_{p'}$ (make the force intensive with respect to a's) is identically zero. With the normals being different there is again an error at truncation level because of the curvature along the slide line.

where the final master velocity is not modified $\vec{v}_{p'}^{\,n+1} = \vec{v}_{p'}^{\,n+1,\dagger}$, while the slave velocity is altered in the following way,

$$\vec{v}_p^{\,n+1} = \vec{v}_p^{\,n+1,\dagger} + \left[\left(\vec{v}_{p'}^{\,n+1,\dagger} + \vec{v}_{p'}^{\,n}\right)\cdot\hat{c}_{p'}\right]\hat{c}_{p'} - \left[\left(\vec{v}_p^{\,n+1,\dagger} + \vec{v}_p^{\,n}\right)\cdot\hat{c}_{p'}\right]\hat{c}_{p'}\,, \tag{19}$$

which guarantees that the projection of the time centered velocities $\vec{v}_{p'}^{\,n+1/2} = (\vec{v}_{p'}^{\,n+1} + \vec{v}_{p'}^{\,n})/2$, $\vec{v}_p^{\,n+1/2} = (\vec{v}_p^{\,n+1} + \vec{v}_p^{\,n})/2$ on the normal $\hat{c}_{p'}$ coincide, i.e. $\vec{v}_{p'}^{\,n+1/2}\cdot\hat{c}_{p'} = \vec{v}_p^{\,n+1/2}\cdot\hat{c}_{p'}$. In words, this correction removes the excessive velocity in the direction of inter-penetration (slide line normal) from the final velocity. Consequently, this correction prevents the slave node to move in this direction more than the corresponding master node does. This correction is weaker than the explicit put-back-on step [?] and does not bring so strong disturbance into the compatible Lagrangian schemes. While this velocity correction provides a way to fulfill the inter-penetration requirement it also affects the energy conservation as this correction is not recast into the compatible formulation.

3.3. Energy discrepancy

In the context of a compatible Lagrangian scheme and away from boundary conditions the total energy is conserved up to machine precision. During one time step $\Delta t = t^{n+1} - t^n$, internal energy in zone z changes as

$$m_z\left(\varepsilon_z^{n+1} - \varepsilon_z^n\right) = -\sum_{p\in P(z)} \vec{F}_p^z\cdot\Delta\vec{r}_p\,, \tag{20}$$

where $\Delta\vec{r}_p = \Delta t(\vec{v}_p^{\,n+1} + \vec{v}_p^{\,n})/2$ represents nodal motion during the time step. Kinetic energy in point p changes as

$$\frac{1}{2}\,m_p\left((\vec{v}_p^{\,n+1})^2 - (\vec{v}_p^{\,n})^2\right). \tag{21}$$

Due to compatibility of the scheme, both energy changes summed over the entire mesh should be the same. However, the presence of a slide line may interfere with this equilibrium. The sliding forces are naturally treated by the energy update of the compatible scheme, but the velocity correction can introduce some discrepancy in the energy conservation. In [?] the author presents a way to measure this discrepancy. The work done by the slide line on a time step Δt upon a point p of the slide line is computed as

$$\Delta W_p^{n,n+1} = \left[\frac{1}{2}\,m_p\left((\vec{v}_p^{\,n+1})^2 - (\vec{v}_p^{\,n})^2\right)\right] - \left[\sum_{z\in\mathcal{Z}(p)}\vec{F}_p^z\cdot\Delta\vec{r}_p\right], \tag{22}$$

where the internal energy change was transformed from a sum over cells to a sum over nodes. By construction of the compatible scheme, the total energy E is conserved, but the work done by the slide line remains so that

$$E^{n+1} - E^n = \sum_{p\text{ in slide line}} \Delta W_p^{n,n+1}\,. \tag{23}$$

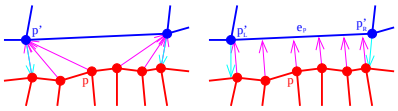
To compute the overall energy discrepancy during the entire simulation, one can accumulate over time,

$$E^{n+1} - E^0 = \sum_{i=1}^{n+1}\sum_{p\text{ in slide line}} \Delta W_p^{i-1,i}\,. \tag{24}$$

This last equation represents the energy balance that must be satisfied up to machine precision and represents the total energy discrepancy brought into the simulation due to the velocity correction of the slide line treatment.

4. Interpolated Interaction

The approach [?] described in the previous Section employs the standard point-to-point interaction where the pressure forces $\vec{F}$, velocities $\vec{v}$, characteristic lengths a_p, and the outer normals $\hat{c}$ on the opposite side of the slide line are simply taken from the nearest opposite point. This approach works reasonably well and it is the best choice from the parallelization point of view, however, it can lead to staircase-shape of the slide line when different aspect ratios are present on each side of the slide line, so the original approach is unlikely to be suitable for problems with high local curvature. This is caused by interaction of several points with a particular point from the opposite side, causing the set of points behaving similarly, while the very next point interacting with a neighboring point from the other side behaves in a significantly different way, as shown in Figure 3 (a). Therefore, we suggest here the interpolated interaction, where all the mentioned



(a) point-to-point interaction (b) interpolated interaction

Figure 3: (a) Point-to-point interaction where many points (p) can interact with one point from the opposite side (p'). (b) Interpolated interaction where points interact with edge e_p, its left and right points denoted by p'_L and p'_R respectively.

quantities from the opposite side of the slide line are interpolated from the adjacent point values. For completeness, let us note that the interpolated interaction is used in practice in various numerical codes, however, not many details are available in open literature. For example, in [?] and [?], the authors treat the interaction of contact nodes with an edge from the other side in the context of Lagrangian multipliers. For example, in [?], the authors accumulate portions of nodal quantities (such as mass) from several opposite-side nodes. In [?], the nodal quantities are distributed to the opposite-side nodes using weighting by mass fractions. And finally in [?], the authors interpolate scalar quantities and normals of vector quantities while projecting the vector components to the normal direction. Similarly, in the approach suggested here, all quantities are interpolated along the edge, however, each component of vector quantities is interpolated separately.

In the point-to-point interaction, a closest point p' from the opposite side has to be determined for any point p on a slide line. In the interpolated interaction, an edge e_p belonging to the opposite side of the slide line and interacting with p has to be determined. Let us note that we reuse the information about the closest point p', and e_p is always one of the slide line edges connected to p'. The relative position of p on e_p is found by the standard projection formula

$$\alpha_p^{e_p} = \max\left(0, \min\left(1, \alpha_p^{e_p,\dagger}\right)\right), \tag{25}$$

$$\alpha_p^{e_p,\dagger} = \frac{(x_p - x_{p'_L})\,(x_{p'_R} - x_{p'_L}) + (y_p - y_{p'_L})\,(y_{p'_R} - y_{p'_L})}{\|p'_R - p'_L\|}\,, \tag{26}$$

where p'_L and p'_R denote the left and right vertexes of e_p. This situation is shown in Figure 3 (b). The interpolation of, for example, mass is then performed by a linear function

$$m_{p'} = m_{p'_L} + \alpha_p^{e_p}\left(m_{p'_R} - m_{p'_L}\right), \tag{27}$$

so the value is different for each node interacting with e_p and smoothly changes from $m_{p'_L}$ to $m_{p'_R}$ along the edge, making the interaction more continuous.

Similarly, as shown in (27) for nodal mass, the remaining nodal quantities from the opposite side of the slide line are interpolated in the same way, using the same pre-computed $\alpha_p^{e_p}$ parameters. These are: nodal

mass $m_{p'}$, characteristic length $a_{p'}$, nodal normals $\hat{c}_{p'}$, and nodal force $\vec{F}_{p'}$ in contact force (16), the same quantities with p subscript instead of p' in contact force (17), and normals $\hat{c}_{p'}$ and velocities $\vec{v}_{p'}^{\,n+1,\dagger}$ and $\vec{v}_{p'}^{\,n}$ in slave velocity update (19). The vector quantities are interpolated component by component, while the interpolated normals are re-normalized to be unit vectors again. This approach leads to a significant improvement of the staircase-shape problem, see Figure 8(a) and 11(b).

5. Numerical Surface Tension

In many tests containing sliding, we can observe the evolution of the Kelvin-Helmholtz instability caused by the shear of different density fluids along each other which can eventually cause the failure of the simulation. This situation can be caused by real physics or wrong model, which is the case for example for gasses, where the disturbance of the interface is the consequence of the missing model for fluid mixing. In simulations of solids, this problem is typically avoided by introducing a model for material stress/strain, such as [? ? ?], which causes the material to behave more rigidly; the instability modes are suppressed. We have decided to adopt the technology of numerical surface tension for interface stabilization which is most adequate for description of behavior of liquids. In the formulation of slide line treatment using the Lagrangian multipliers or the penalty method, one can stabilize the interface by an inherited stabilization parameter [?]. Alternative surface stabilization techniques involve for example the viscous interface dumping described in [?].

In our approach, we incorporate the surface tension effect in the form of tension forces in a similar manner, as described in [?]. Let us emphasize here that (contrary to [?]) these forces in our approach do not represent real physical surface tension, it is only a numerical technique for stabilization of a slide line.

The numerical surface tension force acting from zone z on point p, is constructed as

$$\vec{F}_{zp}^{\,\mathrm{NST}} = s_p\, \sigma_{zp}\, \kappa_p\, \hat{c}_p\,, \tag{28}$$

where the sign s_p defines the orientation of the force, σ_{zp} represents the pressure gradient in the interface normal direction, κ_p represents the local curvature of the slide line, and the nodal unit normal $\hat{c}_p$ defines the direction of the surface tension force. Due to $\hat{c}_p$ term, the numerical surface tension force only acts in the direction normal to the slide line, no tangential component exists. $\vec{F}_{zp}^{\,\mathrm{NST}}$ can be incorporated into the subzonal pressure forces $\vec{F}_{zp}$, and have to be included through this force into the slide line contact forces $\vec{G}_p$.

The orientation of the numerical surface tension force is defined as

$$s_p = \begin{cases} -1 & \text{for } \alpha_p \in \langle 0, \pi) \\ +1 & \text{for } \alpha_p \in \langle \pi, 2\,\pi \rangle \end{cases}, \tag{29}$$

where α_p stands for the internal angle spanned by the slide line edges attached to p. If the slide line is locally convex from one side, the internal angle is acute, therefore $s_p < 0$, and the node is pulled inside the mesh. In the opposite situation, in case of locally concave mesh from one side of the slide line, α_p is obtuse, so s_p is positive, and the node is pushed outside of the mesh. In both cases, the direction leads to straightening of the slide line segment. Let us also present an alternative definition of s_p, avoiding the strong jump in the situation of the straight angle,

$$s_p = \tanh(C(\alpha_p - \pi))\,. \tag{30}$$

Here, the constant C defines the width of the transition region. Let us note that this definition of s_p is smoother, and moreover, $s_p \approx 0$ for $\alpha_p \approx \pi$ what makes the whole numerical surface tension machinery more robust.

The term σ_{zp} represents the pressure gradient in the vicinity of the slide line. As the pressure force $\vec{F}_{zp}^{\,\mathrm{NST}}$ is computed using (8), the pressure gradient can be approximated by it. As we are only interested in the size of the pressure gradient in the direction normal to the slide line, it can be computed as

$$\sigma_{zp} \approx \nabla p_{zp} = \left| \vec{F}_{zp} \cdot \hat{c}_p \right|. \tag{31}$$

The last term which we need to define is the local curvature of the slide line κ_p. We compute the curvature in point p as the maximum curvature of a Bezier curve defined by p and its two neighbors in the slide line, as shown in Figure 4. The Bezier curve is described by the following parametric formula

Figure 4: Part of the mesh attached to a slide line approximated by a Bezier curve. Enumeration of slide line points shown.

$$x(t) = (1-t)^2\, x_1 + 2\,t\,(1-t)\, x_2 + t^2\, x_3\,, \tag{32}$$
$$y(t) = (1-t)^2\, y_1 + 2\,t\,(1-t)\, y_2 + t^2\, y_3\,, \tag{33}$$

where $x_{1,2,3}$ and $y_{1,2,3}$ are the coordinates of the nodes as enumerated in Figure 4, and where the parameter $t \in \langle 0, 1)$. The curvature is defined by the standard formula

$$\kappa_p(t) = \frac{|x'\, y'' - y'\, x''|}{\sqrt{\left(x'^2 + y'^2\right)^3}}\,, \tag{34}$$

where the $'$ symbol represents the derivative with respect to t, and can be computed analytically. The maximum curvature $\kappa_p = \max_{t \in \langle 0,1)} \kappa_p(t)$ is found by the golden section search.

In certain configurations (typically when slide line edges of very different lengths are connected to p), this approach can produce excessive numerical surface tension force due to possibly unbounded value of curvature κ_p. To fix this problem, we have adopted the following limiting approach. If the numerical surface tension force $\vec{F}_{zp}^{\,\mathrm{NST}}$ is too big compared to the hydrodynamic pressure force $\vec{F}_{zp}$, we perform limiting to a certain amount of $\vec{F}_{zp}$,

$$\vec{F}_{zp}^{\,\mathrm{NST,lim}} = \beta\, \frac{\|\vec{F}_{zp}\|}{\|\vec{F}_{zp}^{\,\mathrm{NST}}\|}\, \vec{F}_{zp}^{\,\mathrm{NST}}\,. \tag{35}$$

In practical simulations, we use 10 % of the pressure force, i.e. $\beta = 1/10$. Next to it, in practical calculations, one may want to incorporate a switch which enables the numerical surface tension force only if s_p changes its sign in the neighborhood of p. This avoids the straightening of the slide line in smooth regions, while it still prevents its pathologic zigzagging.

Let us note that the numerical surface tension mechanism can change physics in the vicinity of the interfaces by straightening the material interface. This straightening eliminates the development of instabilities on the interface, which may or may not be desirable. The usage of this mechanism therefore depends on a particular simulation. For problems containing pure sliding of materials along each other, the numerical surface tension mechanism helps to stabilize the interface and increases robustness of the calculation. On the other hand, for simulations of instabilities, the numerical surface tension mechanism can eliminate the growth of the instability completely, so this mechanism is not suitable at all. In general, this mechanism should be used only as little as possible to avoid excessive interface straightening.

6. Numerical Examples

In this Section, we present several numerical examples to demonstrate the behavior of the original method and its comparison with the improved method combining the interpolated interaction and the numerical surface tension.

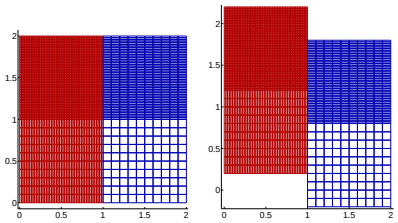


Figure 5: (a) Initial 40×50 and 10×50 meshes of the pure sliding sanity check. (b) Final meshes.

6.1. Pure Sliding

The first numerical example is a sanity check testing the robustness of the methods and their ability to maintain straight material interface. The $\langle 0, 2)^2$ computational domain is split in the middle by a vertical slide line into two different non-uniform computational meshes, with different mesh resolution in each quadrant of the domain [?]. The initial meshes are shown in Figure 5 (a). All fluid quantities are constant in the entire domain except the vertical velocity which is $+0.02$ in the left mesh and -0.02 in the right mesh. The final time is $t = 10$.

All methods keep the straight slide line exactly, no violation occurs, as shown in Figure 5 (b). This is confirmed by the ΔW energy discrepancy which is zero up to machine accuracy for all methods. Due to non-equidistant nature of the meshes, this test checks the infrastructure of the slide line framework, especially the mechanism of scaling the contact forces to the segment size across the interface by the parameter a_p.

6.2. Saltzman-like Piston

Another sanity check is inspired by the standard Saltzman piston problem, however, a standard uniform orthogonal mesh is used instead of the skewed Saltzman mesh. The whole $\langle 0, 1) \times \langle 0, 0.1)$ domain is covered by a 100×10 mesh which is split in the middle of the domain by a horizontal or vertical slide line, as shown in Figure 6 (a,b). The fluid has a unit density and zero (in practice, 10^{-8} is used) pressure, and it is static except the left boundary which is moving with unit velocity representing a piston compressing the fluid. The value of $\gamma = 5/3$ is used everywhere, the final time is $t = 0.98$. The whole problem is rotated by a non-trivial angle $\pi/6$ to avoid interference with the axes directions. (The non-rotated problem produced perfect results for all methods and is consequently skipped.)

Interesting parts of the final computational meshes are shown in Figure 6 (c,d). No difference among the methods can be visually observed, so only images for the interpolated interaction with the numerical surface tension are presented. As we can see, the 1D nature of the problem is preserved perfectly in both cases for all methods. For the horizontal slide line test, no discrepancy occurs at all, the ΔW energy discrepancy is zero up to machine accuracy for all methods. In case of vertical slide line, small discrepancy appears for all methods ($\Delta W \approx 10^{-9}$), and a small oscillation in density shows up close to the slide line. This is caused by the shock wave passing the interface and consequent velocity correction, which did not appear in case of horizontal slide line as no velocity correction was performed due to normal direction of the slide line and the shock wave. However, the 1D symmetry as well as the shock wave velocity is not affected.

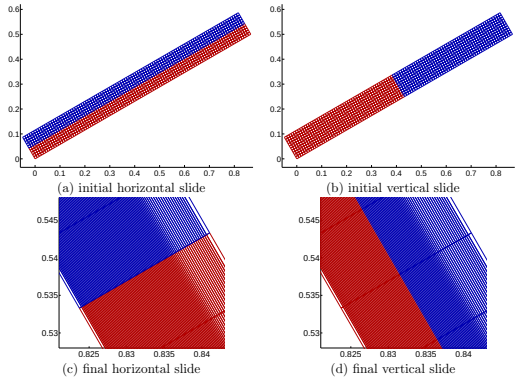


Figure 6: Saltzman-like piston with a slide line: initial (a,b) and final (c,d) meshes for Saltzman-like piston problem separated by a horizontal (a,c) and vertical (b,d) slide line.

6.3. Sliding Rings

In this Section, we describe a problem in which an inner low-density ring slides along an outer heavier ring. This problem is similar to the pure sliding test presented in Section 6.1, however, this is not just a sanity check any more as real hydrodynamic effects take place here due to the centrifugal forces. Both initial meshes contain 100×20 computational cells. The outer mesh uniformly spans from $-\pi/4$ to $5/4\,\pi$ in the angular direction and from 2 to 3 in the radial direction. The inner mesh spans from 0 to $\pi/2$ in the angular direction and from 1 to 2 in the radial direction, so there is a 1/3 aspect ratio between the meshes. The initial meshes are shown in Figure 7. Both meshes contain initially uniform unit pressure, the inner (slave) mesh with uniform unit density slides by a unit angular velocity along the static outer (master) ring with density 10^4. The simulation stops in time 0.65, just before the original method fails.

The results of the simulations are shown in Figure 8. The original method provides significantly worse results when compared with the improved method what is caused by the staircase-shape problem due to a different aspect ratio of the meshes, as described in Section 4. This problem propagates though the computational mesh and can be seen as strong disturbances in the mesh and also in the density (and other quantities) profiles. The interpolated interaction helps significantly and the simulation can continue. As the slide line shape is maintained smooth in this case, the numerical surface tension does not change the behavior significantly due to low curvature term κ_p in formula (28).

Note a significant inter-penetration in the original method, which is eliminated by the interpolated interaction. As for the energy discrepancy due to velocity correction, it is of the order of 10^{-3} for the original method, and one order of magnitude lower for the improved methods. Clearly, this problem demonstrates the ability of the interpolated interaction mechanism to suppress the staircase shape problem and stabilize the interface.

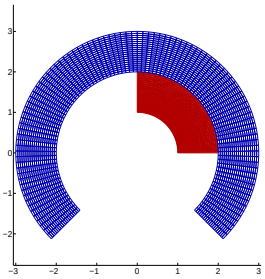
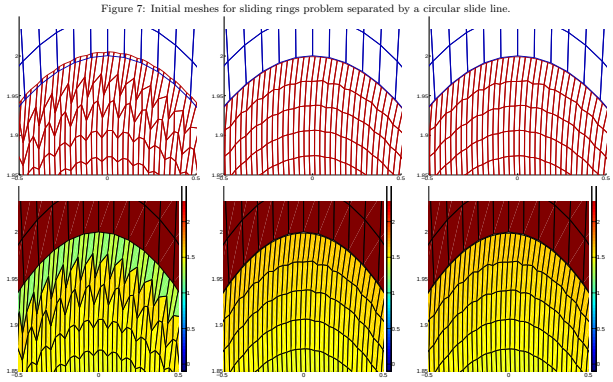


Figure 7: Initial meshes for sliding rings problem separated by a circular slide line.

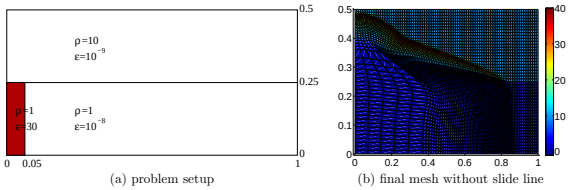


(a) original (b) interpolated interaction (c) interpolated + tension

Figure 8: Zoom to the interesting part along the slide line for the sliding rings problem (aspect ratio not preserved). Comparison of computational meshes and density field for the original approach and the improved method with interpolated interaction and surface tension shown.

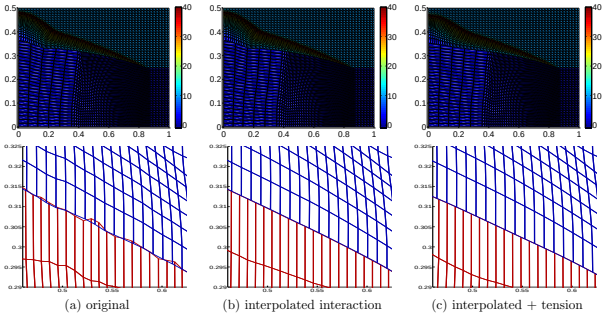
6.4. Explosion with Sliding

In this Section, we demonstrate the behavior of the methods on a more complex test coming from [**?**], representing a realistic problem including a shock wave along the slide line, a shock wave in the direction



(a) problem setup (b) final mesh without slide line

Figure 9: Problem of explosion with sliding. (a) Initial data. (b) Simulation on a single mesh with no slide line present.

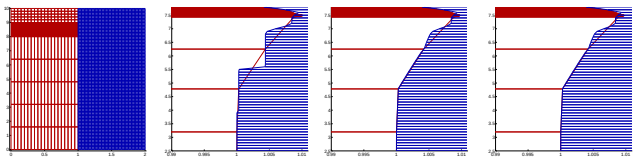


(a) original (b) interpolated interaction (c) interpolated + tension

Figure 10: Results for problem of explosion with sliding. Density profiles and zoom to the slide line regions (aspect ratio not preserved) are shown for the original approach and the improved method with interpolated interaction and surface tension.

normal to the slide line, and sliding at the same time. The computational domain $\langle 0, 1\rangle \times \langle 0, 0.5\rangle$ is divided by a horizontal slide line into two static uniform meshes of 100×25 cells. The lower (slave) mesh has a unit density, pressure is 20 in cells left from $x = 0.05$, and $2/3\ 10^{-8}$ right from it. This high pressure generates a shock wave moving to the right. The upper (master) mesh has density 10 and pressure $2/3\ 10^{-8}$ everywhere. The value of $\gamma = 5/3$ is used in the entire domain. The initial configuration is shown in Figure 9 (a). The simulation is stopped in time $t = 0.4$.

Figure 9 (b) presents the simulation performed on a single computational mesh without a slide line. Note a significant mesh distortion along the material interface due to shear flow, causing high numerical error and eventually degeneracy of the computational time-step. The results of the simulations with a slide line are shown in Figure 10. All methods provide better results than the one-mesh approach, the meshes slide along each other, and the horizontally moving shock wave is nicely captured as no mesh distortion is present. The vertically-moving shock waves from the original method and the improved method are almost identical in



(a) initial (b) original (c) interpolated interact. (d) interpolated + tension

Figure 11: Computational meshes for the bullet in a channel test at $t = 0.1$. (a) Initial mesh of the bullet in a channel test. (b-d) Comparison of the original approach with the improved method with interpolated interaction and surface tension.

the density pictures, when numerical surface tension is added, it is slightly flatter. The void region in the left part of the mesh is bigger for the original approach, it is reduced by interpolated interaction, and almost eliminated by adding the numerical surface tension. The density profile of the original method is almost identical with the profile presented in [**?**].

When looking at the mesh segments along the slide line in the post-shock region for the original method, one can see the development of serious disturbances due to the staircase-shape problem again. This problem is significantly reduced by switching to the interpolated interaction again (Figure 10 (b),(c)). As for the energy discrepancy, it is comparable for all methods for this problem, $\Delta W = 8.8\ 10^{-3}$ for the original method, it drops to $\Delta W = 7.1\ 10^{-3}$ for the interpolated interaction, and increases slightly to $\Delta W = 7.5\ 10^{-3}$ when numerical surface tension is added.

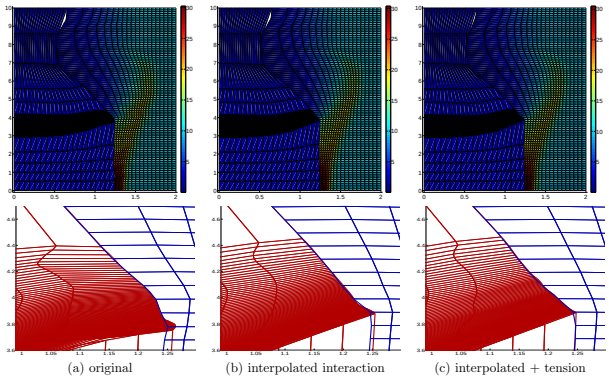
6.5. Bullet in Channel

The last test is inspired by real physical experiments, dealing with ablative acceleration of dense plasma in a channel [**?**], which were performed at the Prague Asterix Laser System (PALS). The initial mesh is shown in Figure 11 (a). The $\langle 0, 2\rangle \times \langle 0, 10\rangle$ computational domain is split by a vertical slide line. The right (master) part is covered by a initially static uniform mesh of 20×100 cells with density equal to 10 and pressure 1, representing the channel boundary. The left (slave) part of the domain represents the inside of the channel. It is divided into three parts: the uppermost air (5 cells between $y = 9$ and $y = 10$), the bullet (90 cells between $y = 8$ and $y = 9$), and air (5 cells between $y = 0$ and $y = 8$). The density is 1 in the heavy bullet and 0.1 in the air. The air is initially static, the bullet moves down with the velocity 7. The value of $\gamma = 5/3$ is used in the entire domain. The simulation stops at time $t = 1.17$, just before the original approach fails.

Due to the bullet motion, the lower air is compressed, the pressure here is increased, and the slide line deforms due to the developed pressure gradient. This problem involves a strong difference in aspect ratios across the slide line, which drives the staircase-shape problem very strongly.

Figure 11 (b),(c),(d) presents the situation in a very early stage of the simulation at $t = 0.1$. The original method suffers from severe distortions. We can clearly identify sets of points from the master side interacting with the particular points on the slave side. On the other hand, the interpolated interaction demonstrates reasonable shift of the master nodes along the edges of the slave boundary. At this early stage of the simulation, the numerical surface tension does not have any significant influence as the shape of the slide line is still smooth.

Figure 12 presents the density field in the entire domain at the final time $t = 1.17$, as well as the part of the meshes close to the slide line. The largest voids between the meshes appear in the case of the original method, but the global picture (bullet shape, shock wave position, etc.) is the same for all methods. Looking at the mesh part along the slide line, one can notice the significant inter-penetration when the



(a) original (b) interpolated interaction (c) interpolated + tension

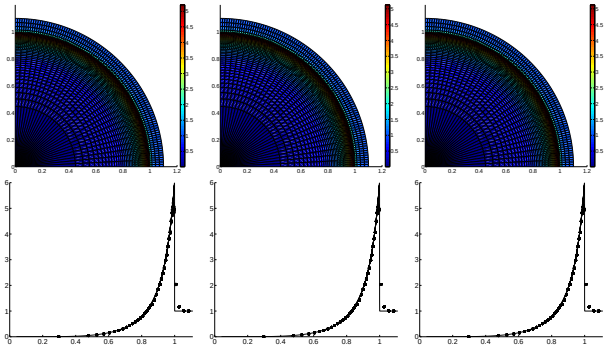
Figure 12: The density profiles in the entire domain and mesh segments in the interesting region close to the slide line (aspect ratios not preserved). Comparison of the original approach with the improved method with interpolated interaction and surface tension.

original method is used, which is reduced by the interpolated interaction and almost eliminated by adding the numerical surface tension. The ΔW energy discrepancy is comparable for all methods.

6.6. Sedov Explosion with an Interface

In this section, we demonstrate the symmetry violations introduced by the slide line machinery for an initially symmetric problem. This problem resembles the standard Sedov point explosion problem on polar computational mesh. The initial mesh covers the radius $r \in \langle 1/100, 1.1\rangle$ and the angular interval is $\theta \in \langle 0, \pi/2\rangle$. Small radius around the origin prevents the degenerate points to appear. The computational mesh is split by a slide line at $r = 1/2$, where the outer mesh acts as its master side. Both meshes have 20 computational cells in the radial direction. The outer mesh contains 100 cells while the inner mesh has only 31 cells in the angular direction, so the nodes do not coincide at the slide line. The entire domain contains unit-density zero-pressure gas (in practice, $p = 10^{-10}$ is used) with $\gamma = 1.4$, except the innermost ring of cells where the value of $p = 114.359$ is used. This value is found to exactly correspond to the standard version of the Sedov problem presented for example in [**?**]. The final time of the simulation is $t = 1$.

When this problem is solved on a single mesh without any slide line, the solution preserves its symmetry up to machine accuracy. We measure the non-symmetry as the ratio of the angular momentum and the total momentum. The solutions when a slide line is present is shown in Figure 13. As we can see, small decrease in the density value is visible at the slide line (around $r = 0.9$ in Figure 13) for all slide line treatments. Let us note this dip is not present for single mesh simulation and probably originates from the energy discrepancy due to velocity correction at the slide line. However when meshes with termination lines or hanging nodes are employed, the preservation of symmetry becomes tricky [**?**] especially when nodes on both sides do not coincide as in our case. Even visually, we can observe some non-symmetry of the solution between the slide line and the shock front, especially for the original approach. The relative momentum in the angular



(a) original (b) interpolated interact. (c) interpolated + tension

Figure 13: The density profiles in the entire mesh (upper figures) and scatter plot of all cell densities as a function of radius (lower figures) against the analytic solution for the original approach and the improved method with interpolated interaction and surface tension.

direction is $4.008 \cdot 10^{-4}$ for the original approach, $3.230 \cdot 10^{-4}$ for the interpolated interaction, and $3.223 \cdot 10^{-4}$ for the interpolated interaction with the surface tension. The ΔW energy discrepancy is comparable for all methods. Let us note that for this problem, the surface tension switch mentioned in Section 5, which disables the tension in smooth regions was used. Without this switch, the surface tension force tries to make the slide line straight and symmetry of the problem is violated significantly.

6.7. Rayleigh-Taylor Instability with Sliding

In this section, we want to analyze the influence of the slide line treatment on the growth rate of the Rayleigh-Taylor instability problem [**?**] against the theoretical rate. The initial configuration is exactly adopted from [**?**]. This simulation is performed on a single 100×600 mesh without any slide line, and also on two 100×300 meshes separated by a slide line. The upper mesh containing the high density fluid acts as master in these calculations.

In Figure 14, we present the growth rates of the instability and their comparison with the analytic rate [**?**] during the linear phase of the instability growth. First of all, we can see that the simulation without the slide line follows almost exactly the analytic curve until $t = 0.6$. After that, the shear along the fluid interface decelerates the growth due to interface nodes belonging to both fluids and stick to each other. This problem is improved by introducing a slide line separating the fluids, so the interface nodes are freely moving along the interface, without further restrictions from the opposite side. At time $t = 1.3$, the growth rates in the calculations with the slide line exceed the single mesh results, better corresponding to the theoretical growth and the results from [**?**].

We can see that the results of the original method almost coincide with the results of the interpolated interaction. Let us note that the result with numerical surface tension is omitted here as it is not appropriate

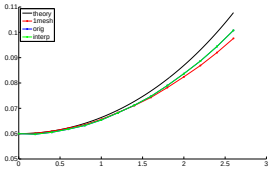


Figure 14: Comparison of analytic growth rate of the Rayleigh-Taylor instability with the numerical calculations performed on single mesh, and on two meshes separated by a slide line treated by original approach (blue) and the interpolated interaction (green). Blue and green curves are superimposed.

for this kind of problem. The numerical surface tension mechanism is designed to stabilize the interface and eliminate the instability, so the instability amplitude decreases instead of increasing according to theory.

7. Conclusions

We have reviewed the slide line treatment from [**?**] in the staggered Lagrangian framework. On the selected numerical examples, we have demonstrated some pathologies of this method in case of different aspect ratios of the computational meshes. This problem can be described as staircase-shape due to similar interaction of several mesh nodes from the finer side of the slide line with a particular node in the coarser side, and just next to them, another set of points interacting with a neighboring node in the coarser side. We describe the technique of interpolated interaction which eliminates this problem.

A second problem is the development of Kelvin-Helmholtz instability in the vicinity of the slide line due to a shear flow in this region. We suggest here a numerical surface tension formulation which suppresses this problem and stabilizes the slide line by making it more rigid. This technique is however devoted to cure hydrodynamical instability. Obviously if such instabilities must be kept or if more advance material strength model is used then this technique should not be employed.

On selected numerical examples, we demonstrate the advantages of the improved methods by comparison with the original approach. We show here that the interpolated interaction provides a much more regular shape of the slide line than the original method. It is also demonstrated that the numerical surface tension process does not affect the slide line shape when it is reasonably smooth, but starts to stabilize it when perturbations appear. In all tests shown here, the improved method provides better results than the original method [**?**].

Acknowledgments

The authors thank E.J. Caramana for clarifications of many details and discussion of many issues related to this topic. This research was supported in parts by the Czech Ministry of Education grants MSM 6840770022, MSM 6840770010, MEB021020, and LC528, and the Czech Science Foundation project P201/10/P086. The authors warmly thank the support of the French embassy in Prague, Czech Repulic under the Vltava program, which has provided the conditions for this work to reach maturity.

1.5 Uniting cell-centered and staggered Lagrangian schemes

In this section one presents the work done mainly by P.-H. Maire, P. Váchal and I on uniting cell-centered and staggered Lagrangian schemes into a common framework. The goal is to extrude the similarity of these supposedly different approaches. I need however to give credit to P.-H. Maire for his breakthrough idea on bridging the two approaches and his ability to divert fruitful tools from his work on cell-centered schemes [48, 49, 102, 50, 103, 104, 105, 51, 42]. This has led to still on-going research and several publications in 2D [23], and 3D [24], proceedings [21, 22] from international conferences. It is worth noticing that a very similar approach has been developed independently by A. Burbeau-Augoula in [106] almost at the same time. In this section I review the 2D and 3D publications on this subject.

2D cell-centered Riemann solver based artificial viscosity. In article [23] *Staggered Lagrangian Discretization Based on Cell-Centered Riemann Solver and Associated Hydrodynamics Scheme*, P.-H. Maire, P. Váchal and I have drawn the basics to design a new form of artificial viscosity for the compatible staggered Lagrangian scheme.
More precisely this work suggests a general formalism to derive staggered discretizations for Lagrangian hydrodynamics on general unstructured meshes in two dimensions. This unified formalism uses the concept of subcell mass and force from the compatible staggered Lagrangian scheme community and a Riemann solver based artificial viscosity from cell-centered Lagrangian scheme community, see [104, 105] for details.
This artificial viscosity form is formulated invoking Galilean invariance and thermodynamic consistency. Moreover the satisfaction of entropy inequality is ensured by using a subcell-based positive definite tensor, M_{cp} for cell c and point p. This tensor is the core of the scheme as it uniquely defines the artificial viscosity and the nature of the scheme *per se*. Let us remind the final form of the subcell force (1.55) for the compatible staggered Lagrangian scheme

$$\boldsymbol{F}_{cp} = \boldsymbol{F}_{cp}^{\text{press}} + \boldsymbol{F}_{cp}^{q} + \boldsymbol{F}_{cp}^{\Delta P},$$

which is constituted of the pressure force $\boldsymbol{F}_{cp}^{\text{press}}$ (1.35), the artificial viscous force $\boldsymbol{F}_{cp}^{q}$ (1.39, 1.40) and the anti-hourglass force $\boldsymbol{F}_{cp}^{\Delta P}$ (1.51). In our work a sufficient condition to obtain the satisfaction of the second law of thermodynamics is to set the subcell force as

$$\boldsymbol{F}_{cp} = \boldsymbol{F}_{cp}^{\text{press}} + \mathsf{M}_{cp}(\boldsymbol{U}_p - \boldsymbol{U}_c), \tag{1.73}$$

where M_{cp} is a 2×2 subcell-based matrix. A seemingly new degree of freedom, the cell-centered velocity $\boldsymbol{U}_c$, is in reality deduced from the point velocities and the subcell-based tensor. Indeed substituting the subcell force expression, $\boldsymbol{F}_{cp} = -L_{cp}P_c\boldsymbol{N}_{cp} + \mathsf{M}_{cp}(\boldsymbol{U}_p - \boldsymbol{U}_c)$, into the Galilean invariance condition, $\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp} = \boldsymbol{0}$, leads to the following system satisfied by the cell-centered velocity $\boldsymbol{U}_c$

$$\mathsf{M}_c\boldsymbol{U}_c = \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{cp}\boldsymbol{U}_p, \tag{1.74}$$

where $\mathsf{M}_c = \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{cp}$ is a symmetric positive definite matrix. Once the definition of the subcell matrix M_{cp} is known, one can solve the previous system to get a unique expression of the cell-centered velocity. By analogy with the node-centered approximate Riemann solver introduced in the context of cell-centered Lagrangian discretization [50], we present one cell-centered approximate Riemann solver. This solver allows to determine one particular form of the subcell matrix M_{cp}. To

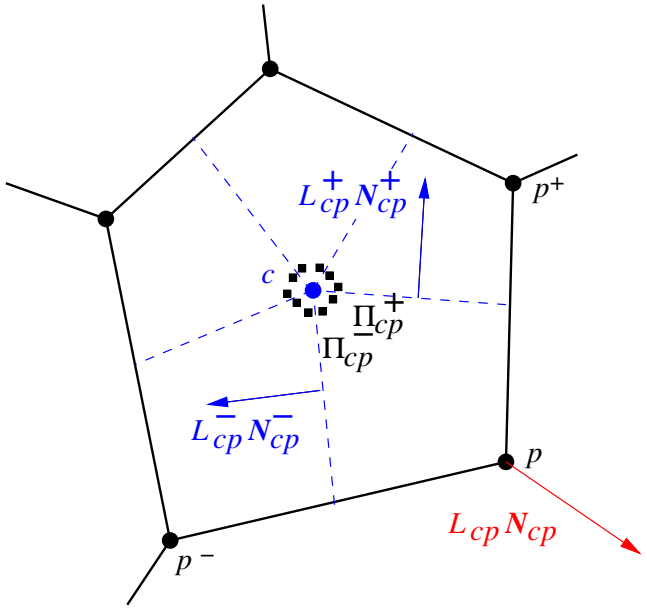


FIGURE 1.14 – *Notation used in the cell-centered Riemann solver. Two pressures per subcell (■) are introduced at the cell center : Π_{cp}^{+}, Π_{cp}^{-}. They are related to the outward normal vectors $L_{cp}^{+}\boldsymbol{N}_{cp}^{+}$, $L_{cp}^{-}\boldsymbol{N}_{cp}^{-}$. In total, $2 \mid \mathcal{P}(c) \mid$ pressures are introduced within cell Ω_c.*

this end, let us introduce two pressures at the cell center per subcell denoted by Π_{cp}^{-}, Π_{cp}^{+}. These pressures are related to the normals $\boldsymbol{N}_{cp}^{+}$, $\boldsymbol{N}_{cp}^{-}$ which are the unit outward normals to the subcell boundaries inside the cell, refer to Fig. 1.14. The subcell force is then defined as

$$\boldsymbol{F}_{cp} = L_{cp}^{-}\Pi_{cp}^{-}\boldsymbol{N}_{cp}^{-} + L_{cp}^{+}\Pi_{cp}^{+}\boldsymbol{N}_{cp}^{+}. \tag{1.75}$$

The cell-centered pressures are obtained by means of the half-Riemann problems

$$P_c - \Pi_{cp}^{-} = Z_{cp}^{-}\left(\boldsymbol{U}_c - \boldsymbol{U}_p\right)\cdot\boldsymbol{N}_{cp}^{-}, \tag{1.76}$$
$$P_c - \Pi_{cp}^{+} = Z_{cp}^{+}\left(\boldsymbol{U}_c - \boldsymbol{U}_p\right)\cdot\boldsymbol{N}_{cp}^{+}, \tag{1.77}$$

where Z_{cp}^{-}, Z_{cp}^{+} denote the swept mass fluxes, and $\boldsymbol{U}_c$ is the cell-centered velocity which remains to be computed. The swept mass fluxes, Z_{cp}^{-}, Z_{cp}^{+}, are defined following Dukowicz [88] as

$$Z_{cp}^{-} = \rho_c\left[\sigma_c + c_Q\Gamma_c \mid (\boldsymbol{U}_c - \boldsymbol{U}_p)\cdot\boldsymbol{N}_{cp}^{-} \mid\right], \qquad Z_{cp}^{+} = \rho_c\left[\sigma_c + c_Q\Gamma_c \mid (\boldsymbol{U}_c - \boldsymbol{U}_p)\cdot\boldsymbol{N}_{cp}^{+} \mid\right]. \tag{1.78}$$

Here, σ_c is the isentropic sound speed, c_Q a user-defined parameter (usually set to 1 in our simulations) and Γ_c a material dependent coefficient, which for a γ gas law is defined by

$$\Gamma_c = \begin{cases} \frac{\gamma+1}{2} & \text{if } (\nabla\cdot\boldsymbol{U})_{cp} < 0, \\ 0 & \text{if } (\nabla\cdot\boldsymbol{U})_{cp} \geq 0, \end{cases} \tag{1.79}$$

where $(\nabla\cdot\boldsymbol{U})_{cp} = -\frac{1}{V_{cp}}L_{cp}\boldsymbol{N}_{cp}\cdot(\boldsymbol{U}_c - \boldsymbol{U}_p)$ is the subcell contribution to the velocity divergence. In case of rarefaction wave, we recover the acoustic approximation whereas in case of shock wave we get the well known two-shock approximation.

Using (1.76)-(1.77) the subcell force is rewritten

$$\boldsymbol{F}_{cp} = \left(L_{cp}^{-}\boldsymbol{N}_{cp}^{-} + L_{cp}^{+}\boldsymbol{N}_{cp}^{+}\right) P_c + \mathsf{M}_{cp}\left(\boldsymbol{U}_p - \boldsymbol{U}_c\right), \tag{1.80}$$

where

$$\mathsf{M}_{cp} = Z_{cp}^{-}L_{cp}^{-}(\boldsymbol{N}_{cp}^{-} \otimes \boldsymbol{N}_{cp}^{-}) + Z_{cp}^{+}L_{cp}^{+}(\boldsymbol{N}_{cp}^{+} \otimes \boldsymbol{N}_{cp}^{+}) \tag{1.81}$$

is a 2×2 symmetric positive definite matrix. Because $L_{cp}^{-}\boldsymbol{N}_{cp}^{-} + L_{cp}^{+}\boldsymbol{N}_{cp}^{+} = -L_{cp}\boldsymbol{N}_{cp}$ then the subcell force writes like (1.73) where the subcell matrix is given by (1.81). The generic form of the subcell force has been retrieved. Our expression of the subcell matrix is directly linked to the half-Riemann invariants (1.76)-(1.77). The cell-centered velocity $\boldsymbol{U}_c$ is obtained by solving the system $\mathsf{M}_c\boldsymbol{U}_c = \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{cp}\boldsymbol{U}_p$, recalling that $\mathsf{M}_c = \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{cp}$ and that M_{cp} is given by (1.81). M_c is symmetric positive definite which ensures its invertibility. Remark that this system is non-linear due to the dependency of the swept mass flux on the cell-centered velocity. This non-linear system can be solved using an iterative procedure such as fixed point or Newton algorithms. In practice, few iterations are needed to get convergence, in fact we only use two iterations. Once the cell-centered velocity is known, the subcell force is deduced from equation (1.73). The present cell-centered approximate Riemann solver can be viewed as a two-dimensional extension of the work initiated by Christensen in one-dimensional framework [87]. The viscous part of subcell force is an important potential link between staggered and cell-centered Lagrangian schemes. While some of the existing artificial viscosity implementations can be reformulated by means of the proposed symmetric positive definite tensor, others still seem to resist this simple interpretation. From this viewpoint there remains enough space for deeper investigation with the prospect of finding similarities and differences between the Godunov and Lagrange like methods.

An elegant way to incorporate the anti-hourglass-like forces (1.51) within the framework consists of incorporating subcell pressure effects by the substitution of P_{cp} into the half-Riemann problems. In other words, one replaces P_c in (1.76-1.77) by P_{cp} as follows

$$P_{cp} - \Pi_{cp}^{-} = Z_{cp}^{-}\left(\boldsymbol{U}_c - \boldsymbol{U}_p\right)\cdot\boldsymbol{N}_{cp}^{-}, \tag{1.82}$$

$$P_{cp} - \Pi_{cp}^{+} = Z_{cp}^{+}\left(\boldsymbol{U}_c - \boldsymbol{U}_p\right)\cdot\boldsymbol{N}_{cp}^{+}. \tag{1.83}$$

The swept mass fluxes are also modified using the subcell density ρ_{cp} and sound speed σ_{cp} as

$$Z_{cp}^{\pm} = \rho_{cp}\left[\sigma_{cp} + c_Q\Gamma_c \mid \left(\boldsymbol{U}_c - \boldsymbol{U}_p\right)\cdot\boldsymbol{N}_{cp}^{\pm} \mid\right].$$

The corresponding subcell force is modified accordingly

$$\boldsymbol{F}_{cp} = -L_{cp}P_{cp}\boldsymbol{N}_{cp} + \mathsf{M}_{cp}\left(\boldsymbol{U}_p - \boldsymbol{U}_c\right). \tag{1.84}$$

Then, the system solving the cell-centered velocity rewrites as

$$\boldsymbol{U}_c = \mathsf{M}_c^{-1}\sum_{p\in\mathcal{P}(c)}\left(\mathsf{M}_{cp}\boldsymbol{U}_p - L_{cp}P_{cp}\boldsymbol{N}_{cp}\right). \tag{1.85}$$

Let us provide an interpretation of the two terms that determine the cell-centered velocity. The first term at the right-hand side is a weighted interpolation of nodal velocities at cell center, whereas the second corresponds to a discretization of the pressure gradient at cell center. This interpretation is obtained by computing the pressure gradient integral over the cell as

$$(\boldsymbol{\nabla}P)_c = \frac{1}{V_c}\int_{\partial\Omega_c} P\boldsymbol{N}\,\mathrm{d}S = \frac{1}{V_c}\sum_{p\in\mathcal{P}(c)}\int_{\partial\Omega_{cp}\cap\partial\Omega_c} P\boldsymbol{N}\,\mathrm{d}S = \frac{1}{V_c}\sum_{p\in\mathcal{P}(c)} L_{cp}P_{cp}\boldsymbol{N}_{cp}. \tag{1.86}$$

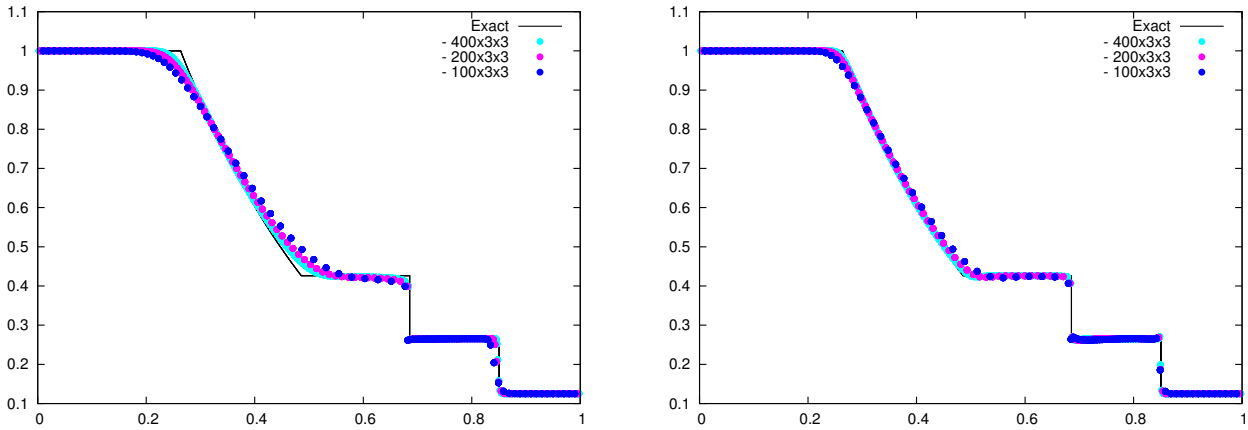


FIGURE 1.15 – *Numerical results from paper [23]. Sod problem at* $t_{final} = 0.2$ *for* $100, 200, 400$ *cells in x direction — Cell-centered density for the generic scheme and the scheme using piecewise linear velocity.*

Then the cell-centered velocity reads

$$\boldsymbol{U}_c = \sum_{p \in \mathcal{P}(c)} \mathsf{M}_c^{-1} \mathsf{M}_{cp} \boldsymbol{U}_p - V_c \mathsf{M}_c^{-1} \left(\boldsymbol{\nabla} P\right)_c . \tag{1.87}$$

This formula degenerates to the previous formula (1.74) in case of uniform subcell pressure over the cell. The extra pressure gradient term induced by the subcell pressures acts as a supplementary viscous term that is usually present in approximate Riemann solver.

An extension to higher order of accuracy in space using piecewise linear reconstruction of velocity field is also developed. The extension in time is obtained with the classical predictor-corrector scheme. One noticable new feature is the vector limitation procedure which is frame independent and thus preserves desirable properties like rotational symmetry. Performance of the new method is demonstrated on a set of classical and demanding numerical tests, using various structured and unstructured computational meshes, Sod, Sedov, Noh, and Saltzman problems and the linear phase of a Rychtmyer-Meshkov instability.

In Fig. 1.15 one reproduces the results obtained by the generic scheme and its extension using piecewise linear velocity on the Sod problem run with the 2D code. These clearly show the improvement gained by the later.

3D cell-centered Riemann solver based artificial viscosity. The promissing approach developed in [23], that is to say the cell-centered Riemann solver based artificial viscosity in 2D, has been further extended by P. Váchal, P.-H. Maire and I in three dimensions. This article appropriately entitled *3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver based artificial viscosity* [24] is depicted in this subsection.

The framework designed in 2D in [23] is extended to 3D. As the original framework was designed for unstructured 2D mesh, its extension to 3D is almost trivial. Besides the natural complication to face when developing a 3D code, we had to extend the notion of reconstruction and limitation of vector field to 3D. In fact the frame invariant limitation demands the definition of two directions in the plane perpendicular to the point velocity which is considered. Several tests have shown that this choice has a significant effect on robustness and/or accuracy. The main focus of this paper was to obtain the more complete and comprehensive picture of the efficiency of the 3D method

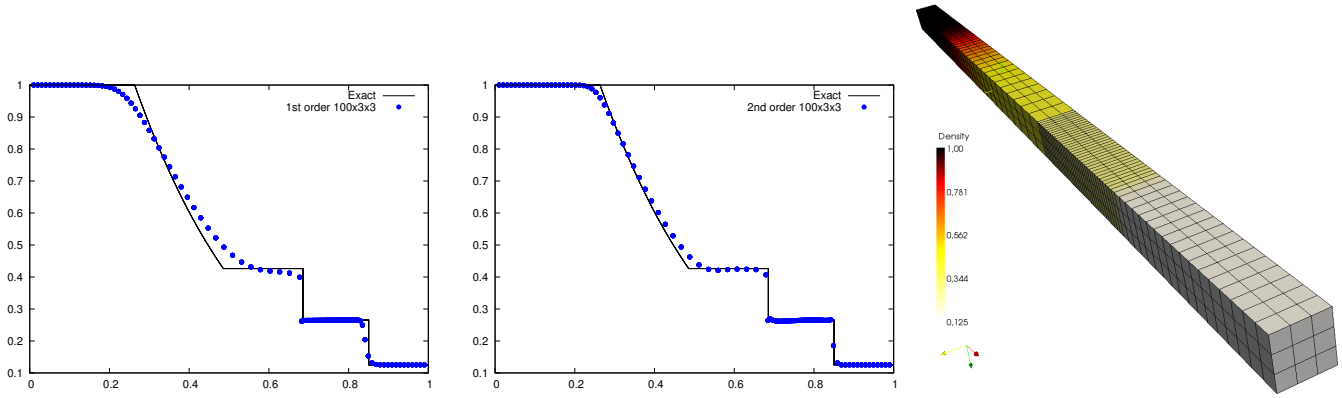


Figure 1.16 – *Numerical results from paper [24]. Sod problem at $t_{final} = 0.2$ for 100 cells in x direction and 3 in y and z directions — Left : generic scheme — Middle : generic scheme using piecewise linear velocity — Right : 3D view. Cell-centered density as a function of x for all cells and 3D view.*

implemented into a simulation code *via* test cases. As for any 3D extension of an existing method great care has to be paid to the implementation details. We then ran sanity checks with the 3D code to retrieve 1D Sod results as in Fig. 1.16. In Fig.1.17 we also show that 2D Sedov results are also retrieved by the 3D code. One has run several 3D problems : Sedov, Noh, Saltzman and Rayleigh-Taylor on hexaedric and polar meshes. The generic compatible staggered Lagrangian scheme and its extension using piecewise linear velocity using a cell-centered Riemann solver based artificial viscosity have been compared on the 3D Sedov problem with a 3D implementation of the compatible staggered Lagrangian scheme using the edge based artificial viscosity popularized in [57]. We have shown that the new approach is able to reproduce almost perfectly spherical symmetry whereas the edge based artificial viscosity presents some spurious mesh instability, see Fig. 1.18

These papers are reproduced in the following pages.

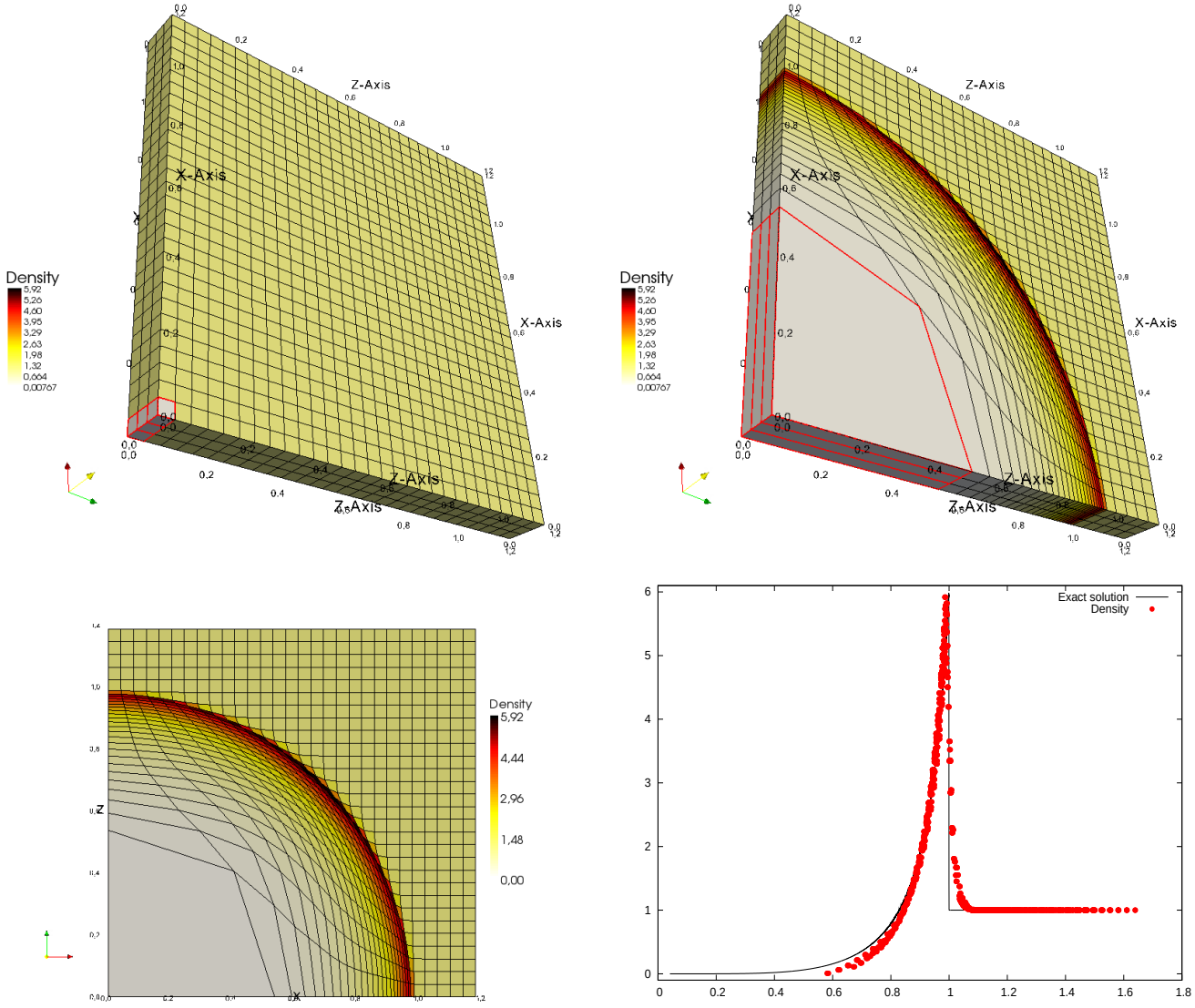


FIGURE 1.17 – *Numerical results from paper [24]. Sedov problem at* $t_{final} = 1$ *— Top : Density map and mesh on a* $30 \times 3 \times 30$ *Cartesian grid at initial and final time (the red highlighted cells are the ones with initial high energy) — Bottom-left : 2D x-z plane at* $y = 0$*, density map and mesh — Bottom-right : Density as a function of cell radius (all cells are displayed).*

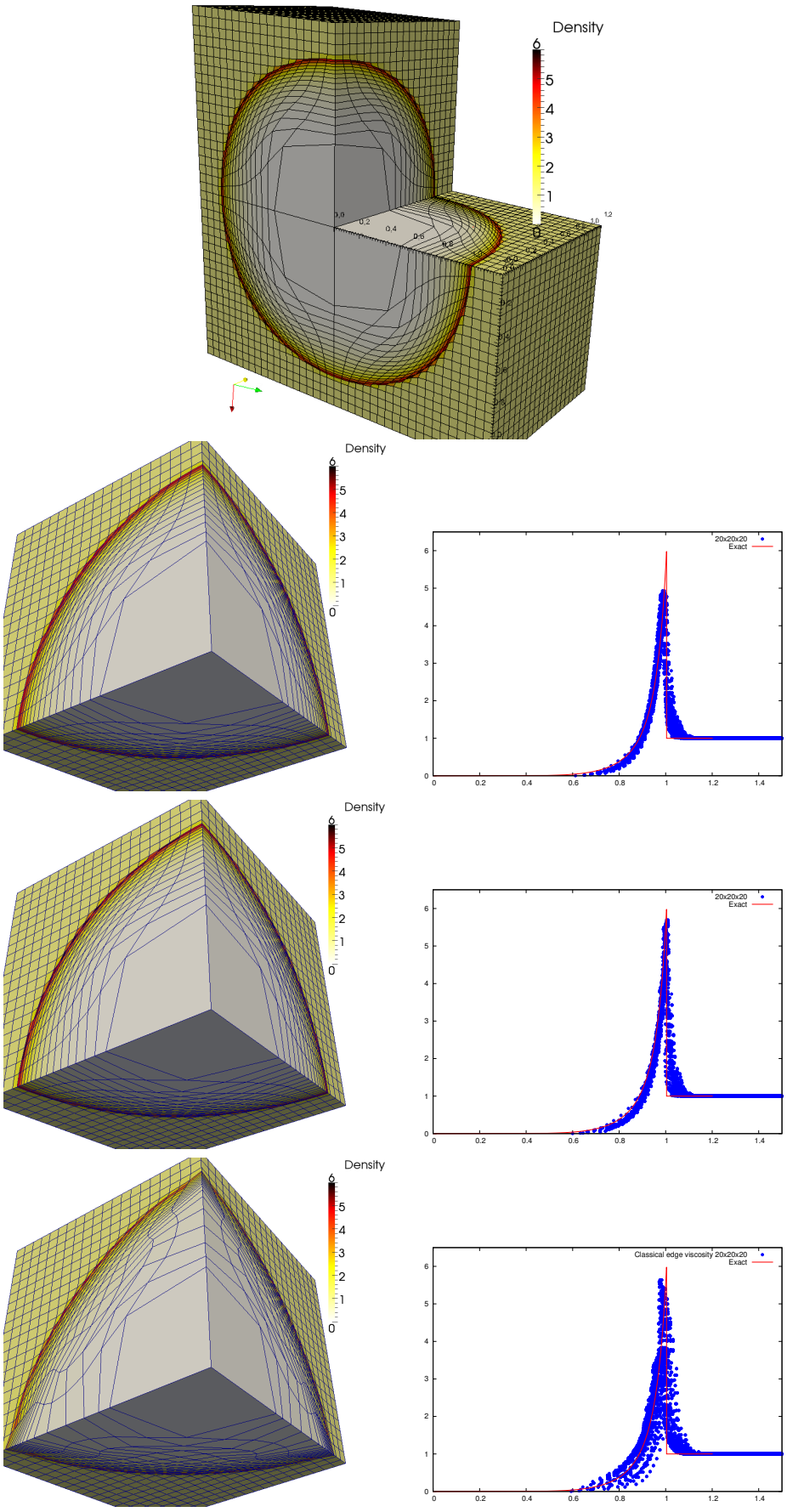


Figure 1.18 – *Numerical results from paper [24]. 3D Sedov problem on a* $20 \times 20 \times 20$ *hexahedral mesh at final time* $t = 1.0$ *— Density and mesh for the generic scheme at the very top line — Density as a function of radius for all cells in the domain for the remaining lines — Top-line : generic scheme — Middle-line : scheme with piecewise linear reconstruction of velocity — Bottom-line : Classical compatible staggered scheme with edge viscosity.*

Commun. Comput. Phys.
doi: 10.4208/cicp.170310.251110a

Vol. **10**, No. 4, pp. 940-978
October 2011

Staggered Lagrangian Discretization Based on Cell-Centered Riemann Solver and Associated Hydrodynamics Scheme

Pierre-Henri Maire[1], Raphaël Loubère[2,*] and Pavel Váchal[3]

[1] *CEA-CESTA, BP 2, 33114, Le Barp, France.*
[2] *Institut de Mathématiques de Toulouse (IMT) Université Paul-Sabatier and CNRS, Toulouse, France.*
[3] *FNSPE, Czech Technical University (CTU) in Prague, Czech Republic.*

Received 17 March 2010; Accepted (in revised version) 25 November 2010

Available online 22 June 2011

Abstract. The aim of the present work is to develop a general formalism to derive staggered discretizations for Lagrangian hydrodynamics on two-dimensional unstructured grids. To this end, we make use of the compatible discretization that has been initially introduced by E. J. Caramana et al., in J. Comput. Phys., 146 (1998). Namely, momentum equation is discretized by means of subcell forces and specific internal energy equation is obtained using total energy conservation. The main contribution of this work lies in the fact that the subcell force is derived invoking Galilean invariance and thermodynamic consistency. That is, we deduce a general form of the sub-cell force so that a cell entropy inequality is satisfied. The subcell force writes as a pressure contribution plus a tensorial viscous contribution which is proportional to the difference between the nodal velocity and the cell-centered velocity. This cell-centered velocity is a supplementary degree of freedom that is solved by means of a cell-centered approximate Riemann solver. To satisfy the second law of thermodynamics, the local subcell tensor involved in the viscous part of the subcell force must be symmetric positive definite. This subcell tensor is the cornerstone of the scheme. One particular expression of this tensor is given. A high-order extension of this discretization is provided. Numerical tests are presented in order to assess the efficiency of this approach. The results obtained for various representative configurations of one and two-dimensional compressible fluid flows show the robustness and the accuracy of this scheme.

AMS subject classifications: 52B10, 65D18, 68U05, 68U07

Key words: Lagrangian scheme, Riemann solver, artificial viscosity, hydrodynamics, compressible flow.

*Corresponding author. *Email addresses:* maire@celia.u-bordeaux1.fr (P.-H. Maire), raphael.loubere@math.univ-toulouse.fr (R. Loubère), vachal@galileo.fjfi.cvut.cz (P. Váchal)

 ©2011 Global-Science Press

1 Introduction

In Lagrangian hydrodynamics methods, a computational cell moves with the flow velocity. In practice, this means that the cell vertices move with a computed velocity, the cell faces being uniquely specified by the vertex positions. Thus, Lagrangian methods can capture contact discontinuity sharply in multi-material fluid flows. However, in the Lagrangian framework, one has to discretize not only the gas dynamics equations but also the vertex motion in order to move the mesh. Moreover, the numerical fluxes of the physical conservation laws must be determined in a compatible way with the vertex velocity so that the geometric conservation law (GCL) is satisfied, namely the rate of change of a Lagrangian volume has to be computed coherently with the node motion. This critical requirement is the cornerstone of any Lagrangian multi-dimensional scheme.

The most natural way to solve this problem employs a staggered discretization in which position, velocity and kinetic energy are centered at points, while thermodynamic variables (density, pressure and specific internal energy) are defined within cells. The dissipation of kinetic energy into internal energy through shock waves is ensured by an artificial viscosity term. Since the seminal works of von Neumann and Richtmyer [33], and Wilkins [34], many developments have been made in order to improve the accuracy and the robustness of staggered hydrodynamics [8, 11, 12]. More specifically, the construction of a compatible staggered discretization leads to a scheme that conserves total energy in a rigorous manner [9, 10].

An alternative to the previous discretizations is to derive a Lagrangian scheme based on the Godunov method [18]. In the Godunov-type method approach, all conserved quantities, including momentum, and hence cell velocity, are cell-centered. The cell-face quantities, including a face-normal component of the velocity, are available from the solution of an approximate Riemann problem at each cell face. However, it remains to determine the vertex velocity in order to move the mesh. In the early work [1] the flux computation was not compatible with the node displacement, and hence the GCL was not satisfied. This incompatibility generated additional spurious components in the vertex velocity field whose correction required expensive treatment [17]. An important achievement concerning the compatibility between flux discretization and vertex velocity computation has been introduced in [15,27]. In these papers, the authors present schemes in which the interface fluxes and the node velocity are computed coherently thanks to an approximate Riemann solver located at the nodes. This original approach leads to first-order conservative schemes which satisfy a local semi-discrete entropy inequality. The multi-dimensional high-order extension of these schemes are developed in [13,25,26,28].

The staggered discretization of variables (kinematic variables located at nodes, thermodynamic ones at cell centers) allows the scheme to fulfill naturally the GCL compatibility requirement and at the same time to construct a discrete divergence operator. The discretizations of momentum and specific internal energy are derived from each other by use of the important concepts of subcell mass, subcell force and total energy conservation [10]. This compatible hydrodynamics algorithm is thus designed to conserve

momentum and total energy exactly in discrete form by using the adjointness property of the discrete gradient and divergence operators. The dissipation of kinetic energy into internal energy through shock waves is ensured by means of an artificial viscosity which can be edge based [12] or tensorial [8]. This mechanism leads to a dissipation that is coherent with the second law of thermodynamics. The subcell pressure method is also used for control of hourglass type motion [11]. Finally, the time integration method is a predictor-corrector technique which is detailed in [10]. The extension of this compatible Lagrangian hydrodynamics algorithm to unstructured grids, where each zone is a polygon with an arbitrary number of sides, has been presented in [9].

Adopting the important concept of the subcells, we are proceeding in the opposite direction than designers of the staggered methods: instead of postulating a form of the artificial viscosity force and the anti-hourglass force, the force is derived from first principles by requiring Galilean invariance and thermodynamic consistency. In other words, if staggered and cell-centered approaches are two paths to the same objective, then the artificial viscosity term (explicit or implicit) should result as a difference of the cell-centered approach (which naturally contains dissipation on shocks) and the staggered approach with the artificial viscosity turned off. The hope is that such artificial viscosity term (if it exists) will be closely related to physical viscosity and thus will improve the method's performance (for example in [8], the authors blame the jets along Cartesian axes in Noh problem on the insufficiency of edge-based artificial viscosity model).

In Godunov methods, the dissipation of kinetic energy into internal energy is provided by solution of a Riemann problem. Our aim here is to use the same mechanism in the framework of a staggered scheme. The solution of the cell-centered Riemann problem provides an approximation of the cell-centered velocity $\boldsymbol{U}_c$, which will then be used to define the viscous part of the subcell force. This formulation allows a straightforward extension to second order in space by constructing linear velocity vector field approximation with frame invariant limitation, applicable on any mesh structure. At this point let us stress that careful and sensitive vector limitation is a key issue to effective exploitation of the improvement gained by frame invariant higher order extension, which is however a fact not always reflected in the design of existing methods. For example, dimensionally split limiters depend on Cartesian framework and thus fail to preserve rotational symmetry. As for temporal integration, we achieve second order in time by employing the predictor-corrector approach.

In the simplest case the resulting viscosity force can be expressed with the help of a symmetric positive definite matrix M_{cp}, so that the thermodynamic consistency is satisfied automatically by the viscous term $\mathsf{M}_{cp}(\boldsymbol{U}_c-\boldsymbol{U}_p)$. In particular, we are adopting the approximate Riemann solver by Dukowicz [16], which is based on the two-shock approximation and gives a viscous term not far from the classical formula by Kuropatenko [20, 34]. It turns out that a similar term (differing only in particular form of tensor M_{cp}) is implicitly contained in some cell-centered schemes such as [15].

Our method can be viewed as an extension of the work by Christensen [14], who noticed that under certain assumptions the staggered Lagrangian schemes with artificial

viscosity can be written in the same form as Godunov's scheme with HLL approximate Riemann solver and stressed the potential synergy of both approaches (e.g., higher-order extension of simple staggered scheme by techniques typically used in the Godunov community, such as TVD limiters). At this point let us remark, that the relationship between staggered Lagrangian and cell-centered Godunov methods from the viewpoint of shock-capturing mechanism has been discussed already in earlier works, e.g., by Wilkins [35] or Dukowicz [16]. Another step towards "bridging the Lagrange-Godunov conceptual gap" was done by Luttwak and Falcovitz [24], who also use a Riemann solver to provide necessary dissipation at shocks in the staggered scheme. The suggested SMG/Q method computes cell-centered velocity gradient to define a principal direction for limiter and shock detector in one. This approach is claimed to be superior to Christensen's split Q in multiple dimensions while being similar to it in the one-dimensional case. Moreover it can be used on structured as well as unstructured meshes. However, the authors themselves are still not happy with the uniaxial formulation of viscosity. Finally let us mention that since linking Godunov with staggered methods is an active research area, this topic was recently also investigated by Burbeau-Augoula [6], who introduced an additional degree of freedom by piecewise constant interpolation of selected variables on primary resp. dual cells. This establishes a connection between cell-centered and staggered formulation. The extra degree of freedom is then coupled to the nodal velocity by defining two half-Riemann problems per edge, which are subsequently treated by the HLL approximate Riemann solver. Assigning each half of the edge the velocity of its corresponding node (endpoint) provides a first order scheme. Besides other issues, on the way to frame invariant higher order extension of this method the strategy of proper limitation must be addressed.

The paper is organized as follows. First the governing equations and notations are stated. The compatible discretization is then derived from first principles. The fourth section deals with the definition of the fundamental object named subcell force. This previous derivation shows the necessity of the introduction of a cell-centered velocity as a new degree of freedom. This velocity is then determined in the fifth section through the use of a cell-centered approximate Riemann solver. High-order extension in space is also provided. The sixth section presents the high-order time discretization. In the last section, numerical results are proposed to assess the validity of this approach. Conclusions and perspectives are finally drawn.

2 Governing equations and notations

2.1 Governing equations

In Lagrangian framework, the two-dimensional gas dynamics equations write

$$\rho\frac{d}{dt}\left(\frac{1}{\rho}\right)-\nabla\cdot\boldsymbol{U}=0, \tag{2.1a}$$

77

$$\rho\frac{d}{dt}\boldsymbol{U}+\boldsymbol{\nabla}P=\boldsymbol{0}, \tag{2.1b}$$

$$\rho\frac{d}{dt}E+\nabla\cdot(P\boldsymbol{U})=0, \tag{2.1c}$$

where ρ is the density, $\boldsymbol{U}$ the velocity, E the specific total energy and $\frac{d}{dt}$ denotes the material derivative. The first equation expresses the volume conservation equation, whereas the second and third ones are the momentum and total energy conservation equations. Volume conservation equation is often referred to as the Geometric Conservation Law (GCL). The previous system is equipped with a thermodynamics closure, Equation of State (EOS), $P=P(\rho,\varepsilon)$, where the specific internal energy is given by $\varepsilon=E-\boldsymbol{U}^2/2$. Note that for smooth solutions energy equation can be rewritten as

$$\rho\frac{d}{dt}\varepsilon+P\nabla\cdot\boldsymbol{U}=0, \tag{2.2}$$

and, substituting volume equation yields

$$\rho\frac{d}{dt}\varepsilon+P\rho\frac{d}{dt}\Big(\frac{1}{\rho}\Big)=0. \tag{2.3}$$

Recalling Gibbs relation for temperature T and specific entropy S: $TdS=d\varepsilon+Pd(\frac{1}{\rho})$, and the second law of thermodynamics, namely $T\frac{dS}{dt}\geq 0$, implies that for non-smooth flows the following relation holds:

$$\rho\frac{d}{dt}\varepsilon+P\nabla\cdot\boldsymbol{U}\geq 0. \tag{2.4}$$

As a consequence, internal energy equation can be viewed as an entropy evolution equation since

$$\rho\frac{d}{dt}\varepsilon+P\rho\frac{d}{dt}\Big(\frac{1}{\rho}\Big)\geq 0. \tag{2.5}$$

The previous System (2.1a)-(2.1c) can therefore be rewritten as a non-conservative system by replacing the energy equation by (2.4). The last equations are the trajectory equations

$$\frac{d\boldsymbol{X}}{dt}=\boldsymbol{U}(\boldsymbol{X}(t),t),\qquad \boldsymbol{X}(0)=\boldsymbol{x}, \tag{2.6}$$

expressing the Lagrangian motion of any point initially located at position $\boldsymbol{x}$.

2.2 Notations

We use a staggered placement of variables in which position and velocity are defined at grid points while thermodynamic variables are located at cell centers, refer to Fig. 1. An

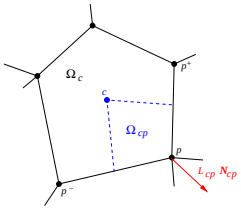


Figure 1: Fragment of a polygonal grid. Position and velocity are defined at grid points while thermodynamic variables are located at cell centers. A polygonal cell, Ω_c, is subdivided into subcells Ω_{cp}. Points are denoted by subscript p and ordered counterclockwise: p^-, p, p^+.

unstructured grid consisting of a collection of non-overlapping polygons is considered. Each polygonal cell is assigned a unique index c and is denoted Ω_c. Each vertex/point of the mesh is assigned a unique index p and we denote $\mathcal{C}(p)$ the set of cells sharing a particular vertex p. Each polygonal cell is subdivided into a set of subcells; each being uniquely defined by a pair of indices c and p and denoted Ω_{cp}. This subcell is constructed by connecting the cell center of Ω_c to the mid-points of cell edges impinging at point p. The union of subcells Ω_{cp} that share a particular vertex p allows to define the dual vertex-centered cell Ω_p related to point p with $\Omega_p=\bigcup_{c\in\mathcal{C}(p)}\Omega_{cp}$. Using the previous notation, we can define the primary grid $\bigcup_c\Omega_c$ and the dual grid $\bigcup_p\Omega_p$. The volumes of the primary and dual cells are functions of time t. Here, following [10], we make the fundamental assumption that the subcells are Lagrangian volumes. This means that the subcell mass m_{cp} is constant in time. Therefore, being given the initial density field $\rho^0(\boldsymbol{x})$ one deduces the initial mean density in cell c

$$\rho_c^0=\int_{\Omega_c(0)}\frac{\rho^0(\boldsymbol{x})}{V_c^0}\mathrm{d}\boldsymbol{x}, \tag{2.7}$$

where V_c^0 is the volume of cell Ω_c at time $t=0$. Subcell mass is defined as $m_{cp}=\rho_c^0\,V_{cp}^0$, where V_{cp}^0 is the initial volume of subcell Ω_{cp}. By summation of Lagrangian subcell masses one defines Lagrangian cell/point masses as

$$m_c=\sum_{p\in\mathcal{P}(c)}m_{cp},\qquad m_p=\sum_{c\in\mathcal{C}(p)}m_{cp}, \tag{2.8}$$

where $\mathcal{P}(c)$ is the set of counterclockwise ordered vertices of cell c. For a vertex p of cell Ω_c we denote its previous and next vertices by p^- and p^+.

3 Compatible discretization

We construct staggered Lagrangian schemes using the well known methodology of compatible discretization which has been presented in [4, 7, 10]. The cornerstone of this type of discretization is the subcell force that acts from subcell cp onto point p. In this approach, the discretization of the internal energy equation in terms of subcell forces is deduced from total energy conservation. Here, we fully derive a generic abstract form of the subcell force so that an entropy inequality is satisfied, which ensures that kinetic energy is dissipated into internal energy through shock waves. The subcell force writes as a pressure contribution plus a tensorial viscous contribution which is proportional to the difference between the vertex-centered and cell-centered velocities. The cell-centered velocity is a supplementary degree of freedom which is determined invoking the fundamental principle of Galilean invariance. To satisfy the second law of thermodynamics, the local subcell matrix involved in the viscous part of the subcell force must be symmetric positive definite. This matrix is the fundamental object that allows to properly define an artificial viscosity required to stabilize the scheme. We remark, that this new framework leads to a new form of artificial viscosity which is derived using first principle arguments.

3.1 Geometric conservation law (GCL)

Here, we use a discretization of the volume Eq. (2.1a) that is compatible with the GCL. By GCL compatibility we mean that we are deriving a discrete divergence operator for the volume equation by requiring consistency of the divergence of the velocity field with the time rate of change of volume of the cell, refer to [29]. By noticing that $m_c=\rho_c V_c$, where $\rho_c=\rho_c(t)$ and $V_c=V_c(t)$ are the cell density and volume, we can write

$$m_c\frac{d}{dt}\Big(\frac{1}{\rho_c}\Big)=\frac{d}{dt}V_c,$$

using the fact that the cell mass is constant in time. Moreover, remarking that the cell volume can be expressed as a function of the position vectors of its vertices as follows

$$V_c(t)=\sum_{p\in\mathcal{P}(c)}\frac{1}{2}(\boldsymbol{X}_p\times\boldsymbol{X}_{p^+})\cdot\boldsymbol{e}_z,$$

where $\boldsymbol{e}_z$ is the unit vector of the canonical basis in z direction, we deduce that the time rate of change of the cell volume writes

$$\frac{d}{dt}V_c=\sum_{p\in\mathcal{P}(c)}\boldsymbol{\nabla}_{\boldsymbol{X}_p}V_c\cdot\frac{d}{dt}\boldsymbol{X}_p.$$

Here, we have simply applied the chain rule differentiation. Setting

$$\frac{d}{dt}\boldsymbol{X}_p=\boldsymbol{U}_p,$$

where $\boldsymbol{U}_p$ is the vertex velocity, we rewrite this last equation as

$$\frac{d}{dt}V_c-\sum_{p\in\mathcal{P}(c)}L_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p=0, \tag{3.1}$$

where $L_{cp}\boldsymbol{N}_{cp}$, with $\boldsymbol{N}_{cp}^2=1$, stands for the corner vector defined by $L_{cp}\boldsymbol{N}_{cp}=\boldsymbol{\nabla}_{\boldsymbol{X}_p}V_c$. This corner vector is a fundamental geometric object which is nothing but the gradient of the cell volume at point p. Its explicit expression in terms of points coordinates writes

$$L_{cp}\boldsymbol{N}_{cp}=\frac{1}{2}\begin{pmatrix}Y_{p^+}-Y_{p^-}\\-(X_{p^+}-X_{p^-})\end{pmatrix},$$

where (X_p,Y_p) denote the coordinate of the position vector $\boldsymbol{X}_p$. This kind of formalism is well known and has been used in staggered and cell-centered (free Lagrange) discretizations long time ago [29, 32]. We note that (3.1) is compatible with the discrete version of the trajectory equation (2.6)

$$\frac{d}{dt}\boldsymbol{X}_p=\boldsymbol{U}_p,\qquad \boldsymbol{X}_p(0)=\boldsymbol{x}_p.$$

This leads to a compatible definition of the discrete divergence operator over cell c as

$$(\nabla\cdot\boldsymbol{U})_c=\frac{1}{V_c}\sum_{p\in\mathcal{P}(c)}L_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p. \tag{3.2}$$

We also emphasize that the corner vector $L_{cp}\boldsymbol{N}_{cp}$ satisfies the fundamental geometric identity

$$\sum_{p\in\mathcal{P}(c)}L_{cp}\boldsymbol{N}_{cp}=\boldsymbol{0}, \tag{3.3}$$

which is equivalent to the well known result that the summation of the outward normals to a closed polygonal contour is equal to zero.

Finally, we have obtained a compatible discretization of the volume equation (2.1a), which writes

$$m_c\frac{d}{dt}\Big(\frac{1}{\rho_c}\Big)-\sum_{p\in\mathcal{P}(c)}L_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p=0. \tag{3.4}$$

3.2 Momentum equation

The semi-discrete momentum equation over the dual cell Ω_p writes

$$m_p\frac{d}{dt}\boldsymbol{U}_p+\sum_{c\in\mathcal{C}(p)}\boldsymbol{F}_{cp}=\boldsymbol{0}. \tag{3.5}$$

Here, $\boldsymbol{F}_{cp}$ is the subcell force from cell c that acts on point p, which is defined by

$$\boldsymbol{F}_{cp}=\int_{\partial\Omega_p(t)\cap\Omega_c(t)} P\boldsymbol{N}\mathrm{d}l. \tag{3.6}$$

Momentum equation (3.5) is nothing but the Newton law applied to particle of mass m_p moving with velocity $\boldsymbol{U}_p$.

3.3 Specific internal energy equation

Here we derive a semi-discrete internal energy equation that ensures total energy conservation using the concept of subcell force, following the approach initially described in [10]. Let us introduce total kinetic energy and total internal energy

$$\mathcal{K}(t)=\sum_p \frac{1}{2}m_p\boldsymbol{U}_p^2(t), \qquad \mathcal{E}(t)=\sum_c m_c\varepsilon_c(t),$$

where ε_c is the cell-averaged specific internal energy. Total energy is then defined as $E(t)=\mathcal{K}(t)+\mathcal{E}(t)$. The conservation of total energy without taking into account boundary conditions simply writes

$$\frac{d}{dt}E=\frac{d}{dt}\mathcal{K}+\frac{d}{dt}\mathcal{E}=0.$$

The substitution of kinetic and internal energies recalling that cell/point masses are Lagrangian objects, i.e., they not depend on time, yields

$$\frac{d}{dt}\mathcal{K}+\frac{d}{dt}\mathcal{E}=\sum_c m_c\frac{d}{dt}\varepsilon_c+\sum_p m_p\frac{d}{dt}\boldsymbol{U}_p\cdot\boldsymbol{U}_p.$$

Using the semi-discrete momentum equation (3.5) yields

$$\sum_c m_c\frac{d}{dt}\varepsilon_c-\sum_p\sum_{c\in\mathcal{C}(p)}\boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p=0,$$

interchanging the order in the double sum one finally gets

$$\sum_c\Big(m_c\frac{d}{dt}\varepsilon_c-\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p\Big)=0. \tag{3.7}$$

A sufficient condition for total energy conservation is obtained by requiring the previous equation to hold in each cell c

$$m_c\frac{d}{dt}\varepsilon_c-\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p=0. \tag{3.8}$$

Once the subcell force is known, then momentum and internal energy can be updated using Eqs. (3.5) and (3.8).

3.4 Summary of the compatible discretization

We summarize the semi-discrete equations that govern the time rate of change of the primary variables $(\frac{1}{\rho_c},\boldsymbol{U}_p,\varepsilon_c)$:

$$\begin{aligned}
&m_c\frac{d}{dt}\Big(\frac{1}{\rho_c}\Big)-\sum_{p\in\mathcal{P}(c)}L_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p=0,\\
&m_p\frac{d}{dt}\boldsymbol{U}_p+\sum_{c\in\mathcal{C}(p)}\boldsymbol{F}_{cp}=\boldsymbol{0},\\
&m_c\frac{d}{dt}\varepsilon_c-\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p=0.
\end{aligned}$$

We point out that the mesh motion is given by the trajectory equations

$$\frac{d}{dt}\boldsymbol{X}_p=\boldsymbol{U}_p(\boldsymbol{X}_p(t),t), \qquad \boldsymbol{X}_p(0)=\boldsymbol{x}_p,$$

which is compatible with the GCL. The thermodynamic closure is given by the equation of state which writes $P_c=P(\rho_c,\varepsilon_c)$. We emphasize that this subcell-based compatible discretization ensures total energy conservation regardless of the subcell force form. Now, it remains to determine the general form of this force so that our semi-discrete scheme fulfills, first, the principle of being Galilean invariant, second, the principle of being consistent with the second law of thermodynamics.

4 Definition of the subcell force

Here we provide a definition of the subcell force using Galilean invariance and thermodynamic consistency.

4.1 Galilean invariance

Galilean invariance is a principle of relativity which states that the fundamental laws of physics are the same in all inertial frames. It is one of the key requirements of many physical models adopted in theoretical and computational mechanics. To fulfill Galilean invariance, the previously derived specific internal energy equation (3.8) must remain unchanged under a uniform translation of frame. Let $\boldsymbol{A}$ denote the uniform translation velocity. Then Eq. (3.8) transforms into

$$m_c\frac{d}{dt}\varepsilon_c-\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}\cdot(\boldsymbol{U}_p+\boldsymbol{A})=0.$$

By substituting (3.8) into this last equation leads to

$$\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}\cdot\boldsymbol{A}=0,$$

which must hold for all vectors $\boldsymbol{A}$. Therefore, specific internal energy equation remains invariant under uniform translation if and only if

$$\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}=\boldsymbol{0}. \tag{4.1}$$

We note that this result has been already quoted in [4] page 576. This condition also implies total momentum conservation without taking into account boundary conditions. To demonstrate this, it suffices to time-differentiate the global momentum defined as $\boldsymbol{\mathcal{Q}}=\sum_p m_p\boldsymbol{U}_p$:

$$\begin{aligned}
\frac{d}{dt}\boldsymbol{\mathcal{Q}}&=\sum_p m_p\frac{d}{dt}\boldsymbol{U}_p\\
&=-\sum_p\sum_{c\in\mathcal{C}(p)}\boldsymbol{F}_{cp}\quad\text{(thanks to momentum equation)}\\
&=-\sum_c\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}\quad\text{(by interchanging the double sums).}
\end{aligned}$$

Thus, $\frac{d}{dt}\boldsymbol{\mathcal{Q}}=\boldsymbol{0}$ due to condition (4.1), which completes the proof.

A corollary of the Galilean invariance condition is that specific internal energy equation (3.8) can also be rewritten into

$$m_c\frac{d}{dt}\varepsilon_c-\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}\cdot(\boldsymbol{U}_p-\boldsymbol{U}_c)=0, \tag{4.2}$$

where $\boldsymbol{U}_c$ is a piecewise constant cell-centered velocity that remains to be determined.

4.2 Thermodynamic consistency

We investigate the thermodynamic consistency of our semi-discrete scheme by computing the time rate of change of entropy in cell c. Using Gibbs formula, one gets

$$m_cT_c\frac{d}{dt}S_c=m_c\Big[\frac{d}{dt}\varepsilon_c+P_c\frac{d}{dt}\Big(\frac{1}{\rho_c}\Big)\Big], \tag{4.3}$$

where S_c and T_c are the specific entropy and temperature of cell c. Substituting into (4.3) the specific internal energy equation (4.2) and the volume equation (3.4) yields

$$\begin{aligned}
m_cT_c\frac{d}{dt}S_c&=\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}\cdot(\boldsymbol{U}_p-\boldsymbol{U}_c)+P_c\Big(\sum_{p\in\mathcal{P}(c)}L_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p\Big)\\
&=\sum_{p\in\mathcal{P}(c)}(\boldsymbol{F}_{cp}+L_{cp}P_c\boldsymbol{N}_{cp})\cdot(\boldsymbol{U}_p-\boldsymbol{U}_c).
\end{aligned}$$

Here, we have used the geometric identity $\sum_{p\in\mathcal{P}(c)}L_{cp}\boldsymbol{N}_{cp}=\boldsymbol{0}$. To satisfy the second law of thermodynamics the right-hand side of the last equation must be positive. A sufficient condition to obtain this consists in setting

$$\boldsymbol{F}_{cp}=-L_{cp}P_c\boldsymbol{N}_{cp}+\mathsf{M}_{cp}(\boldsymbol{U}_p-\boldsymbol{U}_c), \tag{4.4}$$

where M_{cp} is a subcell-based matrix.

Given this form, several features of M_{cp} can be drawn:

1. *Dimensionality.* M_{cp} has dimension of density times velocity times length.
2. *Entropy inequality satisfaction.* M_{cp} is positive semidefinite, i.e., $\mathsf{M}_{cp}\boldsymbol{U}\cdot\boldsymbol{U}\geq 0, \forall\boldsymbol{U}\in\mathbb{R}^2$. By substituting (4.4) into (4.3), we obtain the entropy inequality satisfied by our semi-discrete scheme
$$m_cT_c\frac{\mathrm{d}}{\mathrm{d}t}S_c=\sum_{p\in\mathcal{P}(c)}\mathsf{M}_{cp}(\boldsymbol{U}_p-\boldsymbol{U}_c)\cdot(\boldsymbol{U}_p-\boldsymbol{U}_c)\geq 0, \tag{4.5}$$
as the right-hand side is a positive semidefinite quadratic form.
3. *Galilean invariance.* M_{cp} is compatible with the principle of Galilean invariance: in a nutshell, M_{cp} must be invariant w.r.t. translation and transform as $\mathcal{R}\mathsf{M}_{cp}\mathcal{R}^t$ for a rigid rotation $\mathcal{R}$.
4. *Symmetry.* M_{cp} is symmetric, i.e., $\mathsf{M}_{cp}^t=\mathsf{M}_{cp}$.
5. *Locality.* M_{cp} is a locally defined matrix: the physical and geometric quantities involved in M_{cp} must be local in a neighborhood of the current subcell.

We remark that entropy production (4.5) within cell c is directly governed by the subcell matrix M_{cp} and the velocity jump between the nodal and the cell-centered velocity which still remains to be determined. This is the main topic of next section.

5 Cell-centered velocity computation

Using the previously derived generic form of the subcell force and the Galilean invariance condition, we develop a cell-centered solver to compute the cell-centered velocity.

5.1 Abstract formulation

Substituting the subcell force expression,

$$\boldsymbol{F}_{cp}=-L_{cp}P_c\boldsymbol{N}_{cp}+\mathsf{M}_{cp}(\boldsymbol{U}_p-\boldsymbol{U}_c),$$

into the Galilean invariance condition, $\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}=\boldsymbol{0}$, leads to the following system satisfied by the cell-centered velocity $\boldsymbol{U}_c$,

$$\mathsf{M}_c\boldsymbol{U}_c=\sum_{p\in\mathcal{P}(c)}\mathsf{M}_{cp}\boldsymbol{U}_p, \tag{5.1}$$

where $\mathsf{M}_c=\sum_{p\in\mathcal{P}(c)}\mathsf{M}_{cp}$ is a symmetric positive definite matrix. Once the definition of the subcell matrix M_{cp} is known, one can solve the previous system to get a unique expression of the cell-centered velocity. We shall see in the next paragraph one example of such a solver.

5.2 Cell-centered approximate Riemann solver

By analogy with the node-centered approximate Riemann solver introduced in the context of cell-centered Lagrangian discretization [26], we present one cell-centered approximate Riemann solver. This solver allows to determine one particular form of the subcell matrix M_{cp}. To this end, let us introduce two pressures at the cell center per subcell denoted by Π_{cp}^-, Π_{cp}^+. These pressures are related to the normals $\boldsymbol{N}_{cp}^+$, $\boldsymbol{N}_{cp}^-$ which are the unit outward normals to the subcell boundaries inside the cell, refer to Fig. 2. The subcell force is then defined as

$$\boldsymbol{F}_{cp}=L_{cp}^-\Pi_{cp}^-\boldsymbol{N}_{cp}^-+L_{cp}^+\Pi_{cp}^+\boldsymbol{N}_{cp}^+. \tag{5.2}$$

The cell-centered pressures are obtained by means of the half-Riemann problems

$$P_c-\Pi_{cp}^-=Z_{cp}^-(\boldsymbol{U}_c-\boldsymbol{U}_p)\cdot\boldsymbol{N}_{cp}^-, \tag{5.3a}$$
$$P_c-\Pi_{cp}^+=Z_{cp}^+(\boldsymbol{U}_c-\boldsymbol{U}_p)\cdot\boldsymbol{N}_{cp}^+, \tag{5.3b}$$

where Z_{cp}^-, Z_{cp}^+ denote the swept mass fluxes, and $\boldsymbol{U}_c$ is the cell-centered velocity which remains to be computed. The swept mass fluxes, Z_{cp}^-, Z_{cp}^+, are defined following Dukowicz [16] as

$$Z_{cp}^-=\rho_c\left[\sigma_c+c_Q\Gamma_c|(\boldsymbol{U}_c-\boldsymbol{U}_p)\cdot\boldsymbol{N}_{cp}^-|\right],\qquad Z_{cp}^+=\rho_c\left[\sigma_c+c_Q\Gamma_c|(\boldsymbol{U}_c-\boldsymbol{U}_p)\cdot\boldsymbol{N}_{cp}^+|\right]. \tag{5.4}$$

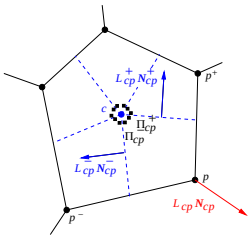


Figure 2: Notation used in the cell-centered Riemann solver. Two pressures per subcell (■) are introduced at the cell center: Π_{cp}^+, Π_{cp}^-. They are related to the outward normal vectors $L_{cp}^+\boldsymbol{N}_{cp}^+$, $L_{cp}^-\boldsymbol{N}_{cp}^-$. In total, $2|\mathcal{P}(c)|$ pressures are introduced within cell Ω_c.

Here, σ_c is the isentropic sound speed, c_Q a user-defined parameter (set to 1 in our simulations) and Γ_c a material dependent coefficient, which for a γ gas law is defined by

$$\Gamma_c=\begin{cases}\dfrac{\gamma+1}{2}, & \text{if } (\nabla\cdot\boldsymbol{U})_{cp}<0,\\ 0, & \text{if } (\nabla\cdot\boldsymbol{U})_{cp}\geq 0,\end{cases} \tag{5.5}$$

where

$$(\nabla\cdot\boldsymbol{U})_{cp}=-\frac{1}{V_{cp}}L_{cp}\boldsymbol{N}_{cp}\cdot(\boldsymbol{U}_c-\boldsymbol{U}_p)$$

is the subcell contribution to the velocity divergence. In case of rarefaction wave, we recover the acoustic approximation whereas in case of shock wave we get the well known two-shock approximation.

Using (5.3) the subcell force is rewritten

$$\boldsymbol{F}_{cp}=(L_{cp}^-\boldsymbol{N}_{cp}^-+L_{cp}^+\boldsymbol{N}_{cp}^+)P_c+\mathsf{M}_{cp}(\boldsymbol{U}_p-\boldsymbol{U}_c), \tag{5.6}$$

where

$$\mathsf{M}_{cp}=Z_{cp}^-L_{cp}^-(\boldsymbol{N}_{cp}^-\otimes\boldsymbol{N}_{cp}^-)+Z_{cp}^+L_{cp}^+(\boldsymbol{N}_{cp}^+\otimes\boldsymbol{N}_{cp}^+) \tag{5.7}$$

is a 2×2 symmetric positive definite matrix. Noticing that the subcell contour is closed, we deduce that $L_{cp}^-\boldsymbol{N}_{cp}^-+L_{cp}^+\boldsymbol{N}_{cp}^+=-L_{cp}\boldsymbol{N}_{cp}$. Finally, the subcell force writes

$$\boldsymbol{F}_{cp}=-L_{cp}P_c\boldsymbol{N}_{cp}+\mathsf{M}_{cp}(\boldsymbol{U}_p-\boldsymbol{U}_c),$$

where the subcell matrix is given by (5.7). We emphasize that we have recovered the generic form of the subcell force which has been previously derived. Moreover, we have given a particular expression of the subcell matrix which is directly linked to the half-Riemann invariants (5.3). Finally, the cell-centered velocity $\boldsymbol{U}_c$ is obtained by solving the system $\mathsf{M}_c\boldsymbol{U}_c=\sum_{p\in\mathcal{P}(c)}\mathsf{M}_{cp}\boldsymbol{U}_p$, recalling that $\mathsf{M}_c=\sum_{p\in\mathcal{P}(c)}\mathsf{M}_{cp}$ and that M_{cp} is given by (5.7). We note that M_c is symmetric positive definite which ensures its invertibility. We remark that this system is non-linear due to the dependency of the swept mass flux on the cell-centered velocity. This non-linear system can be solved using an iterative procedure such as fixed point or Newton algorithms. In practice, few iterations are needed to reach convergence. Once the cell-centered velocity is known, the subcell force is deduced from Eq. (4.4). The present cell-centered approximate Riemann solver can be viewed as a two-dimensional extension of the work initiated by Christensen in one-dimensional framework [14].

5.3 High-order extension

The previously defined cell-centered approximate Riemann solver utilizes piecewise constant nodal velocities defined over the subcells of a given cell. In this sense, this leads to a first-order approximation of the cell-centered velocity. To get a more accurate approximation, we construct a piecewise linear representation of the nodal velocity field to feed the cell-centered solver with the extrapolated velocity at the cell center.

5.3.1 Piecewise linear reconstruction of the velocity field

First, we introduce a piecewise linear representation of the velocity field over the dual grid by setting

$$\boldsymbol{U}_p(\boldsymbol{X})=\boldsymbol{U}_p+\nabla\boldsymbol{U}_p\cdot(\boldsymbol{X}-\boldsymbol{X}_p), \tag{5.8}$$

where $\nabla\boldsymbol{U}_p$ is the constant velocity tensor gradient over the dual cell Ω_p. To compute it we use a classical least squares approach by solving the following minimization problem

$$\nabla\boldsymbol{U}_p=\operatorname{argmin}\sum_{q\in\mathcal{N}(p)}\left[(\boldsymbol{U}_q-\boldsymbol{U}_p)-\nabla\boldsymbol{U}_p(\boldsymbol{X}_q-\boldsymbol{X}_p)\right]^2, \tag{5.9}$$

where $\mathcal{N}(p)$ is the set of neighbor vertices of vertex p. The solution of this minimization problem reads

$$\nabla\boldsymbol{U}_p=\mathsf{M}_p^{-1}\sum_{q\in\mathcal{N}(p)}(\boldsymbol{U}_q-\boldsymbol{U}_p)\otimes(\boldsymbol{X}_q-\boldsymbol{X}_p), \tag{5.10}$$

where matrix M_p is the symmetric positive definite matrix

$$\mathsf{M}_p=\sum_{q\in\mathcal{N}(p)}(\boldsymbol{X}_q-\boldsymbol{X}_p)\otimes(\boldsymbol{X}_q-\boldsymbol{X}_p). \tag{5.11}$$

We emphasize that this least squares approach is valid for any type of unstructured grid and preserves linear velocity field.

5.3.2 Frame invariant tensorial limitation for a vector field

Monotonicity is achieved thanks to a modification of the classical Barth-Jespersen slope limiter [2,3]. For vectors, limiting is usually applied separately to each component. However, such a procedure is frame dependent and thus leads to rotational symmetry distortion. Namely, component limiters do not preserve symmetry since a rotation of the coordinate axis produces different results. This drawback is crucial in the framework of Lagrangian hydrodynamics since we are dealing with moving mesh discretizations which are particularly sensitive about symmetry loss. To correct this flaw, we have to construct a limiting procedure which is frame invariant for vectors. One possible choice is to use the Vector Image Polygon (VIP) methodology derived in [23]. This method consists in constructing the VIP as the convex hull of the vector-space points corresponding to the neighbor vectors. If a slope-extrapolated vector lies inside the VIP, the slope is monotonicity preserving, otherwise slope limiting is required. On the other hand, the slope is set to zero by analogy with the scalar limitation. Here we develop an original procedure to perform a limitation of vector field which preserves rotational symmetry. To define a limiter for the velocity tensor gradient, we define in each dual cell Ω_p a local

orthonormal basis $(\boldsymbol{\xi}_p^{\parallel},\boldsymbol{\xi}_p^{\perp})$ which is assumed to be frame independent. Let us define the coordinates of the vectors $\boldsymbol{\xi}_p^{\parallel}$ and $\boldsymbol{\xi}_p^{\perp}$ in the canonical basis by setting

$$\boldsymbol{\xi}_p^{\parallel}=\begin{pmatrix}\xi_x\\ \xi_y\end{pmatrix},\qquad \boldsymbol{\xi}_p^{\perp}=\begin{pmatrix}-\xi_y\\ \xi_x\end{pmatrix},$$

where $\xi_x^2+\xi_y^2=1$ so that $(\boldsymbol{\xi}_p^{\parallel},\boldsymbol{\xi}_p^{\perp})$ is a direct orthonormal basis. Thus the transformation matrix from the canonical basis to the local basis reads

$$\mathsf{A}_p=\begin{pmatrix}\xi_x & \xi_y\\ -\xi_y & \xi_x\end{pmatrix}.$$

The transformation of the nodal velocity $\boldsymbol{U}_p$ to the local coordinates is

$$\boldsymbol{W}_p=\begin{pmatrix}W_p^{\parallel}\\ W_p^{\perp}\end{pmatrix}=\mathsf{A}_p\boldsymbol{U}_p=\begin{pmatrix}\boldsymbol{\xi}_p^{\parallel}\cdot\boldsymbol{U}_p\\ \boldsymbol{\xi}_p^{\perp}\cdot\boldsymbol{U}_p\end{pmatrix}. \tag{5.12}$$

Then we find the minimum and maximum value from projections of neighboring nodes' velocities into new directions:

$$W_p^{\parallel,\max}=\max_{k\in\mathcal{N}(p)}(\boldsymbol{\xi}_p^{\parallel}\cdot\boldsymbol{U}_k),\qquad W_p^{\perp,\max}=\max_{k\in\mathcal{N}(p)}(\boldsymbol{\xi}_p^{\perp}\cdot\boldsymbol{U}_k), \tag{5.13a}$$
$$W_p^{\parallel,\min}=\min_{k\in\mathcal{N}(p)}(\boldsymbol{\xi}_p^{\parallel}\cdot\boldsymbol{U}_k),\qquad W_p^{\perp,\min}=\min_{k\in\mathcal{N}(p)}(\boldsymbol{\xi}_p^{\perp}\cdot\boldsymbol{U}_k), \tag{5.13b}$$

where $\mathcal{N}(p)$ is the set of neighbor points of current point p. Now consider cell $c\in\mathcal{C}(p)$ centered at $\boldsymbol{X}_c$. Using the unlimited piecewise linear representation of the velocity field, the extrapolated values of the velocity at point $\boldsymbol{X}_c$ are given by

$$\boldsymbol{U}_{p,c}\equiv\boldsymbol{U}_p(\boldsymbol{X}_c)=\boldsymbol{U}_p+\nabla\boldsymbol{U}_p\cdot(\boldsymbol{X}_c-\boldsymbol{X}_p), \tag{5.14}$$

and its transformation into the local basis $(\boldsymbol{\xi}_p^{\parallel},\boldsymbol{\xi}_p^{\perp})$ produces

$$\boldsymbol{W}_{p,c}=\begin{pmatrix}W_{p,c}^{\parallel}\\ W_{p,c}^{\perp}\end{pmatrix}=\mathsf{A}_p\boldsymbol{U}_{p,c}. \tag{5.15}$$

From these values we define

$$\phi_{p,c}^{\parallel}=\begin{cases}\mathcal{L}\left(\frac{W_p^{\parallel,\max}-W_p^{\parallel}}{W_{p,c}^{\parallel}-W_p^{\parallel}}\right), & \text{if } (W_{p,c}^{\parallel}-W_p^{\parallel})>0,\\ \mathcal{L}\left(\frac{W_p^{\parallel,\min}-W_p^{\parallel}}{W_{p,c}^{\parallel}-W_p^{\parallel}}\right), & \text{if } (W_{p,c}^{\parallel}-W_p^{\parallel})<0,\\ 1, & \text{if } (W_{p,c}^{\parallel}-W_p^{\parallel})=0,\end{cases}$$
$$\phi_{p,c}^{\perp}=\begin{cases}\mathcal{L}\left(\frac{W_p^{\perp,\max}-W_p^{\perp}}{W_{p,c}^{\perp}-W_p^{\perp}}\right), & \text{if } (W_{p,c}^{\perp}-W_p^{\perp})>0,\\ \mathcal{L}\left(\frac{W_p^{\perp,\min}-W_p^{\perp}}{W_{p,c}^{\perp}-W_p^{\perp}}\right), & \text{if } (W_{p,c}^{\perp}-W_p^{\perp})<0,\\ 1, & \text{if } (W_{p,c}^{\perp}-W_p^{\perp})=0,\end{cases}$$

where $\mathcal{L}(\alpha)$ is a suitable limiting functional such as

$$\mathcal{L}(\alpha)=\min(\alpha,1) \quad \text{or} \quad \mathcal{L}(\alpha)=(\alpha^2+2\alpha)(\alpha^2+\alpha+2)^{-1}.$$

The slope limiters for node p are finally defined by $\phi_p^{\parallel}$ and $\phi_p^{\perp}$ as

$$\phi_p^{\parallel}=\min_{c\in\mathcal{C}(p)}\phi_{p,c}^{\parallel}, \qquad \phi_p^{\perp}=\min_{c\in\mathcal{C}(p)}\phi_{p,c}^{\perp}.$$

This pair of limiters is transformed back into the Cartesian coordinates:

$$\Phi_p=\mathsf{A}_p^{-1}\begin{pmatrix}\phi_p^{\parallel} & 0\\ 0 & \phi_p^{\perp}\end{pmatrix}\mathsf{A}_p=\begin{pmatrix}\xi_x^2\,\phi_p^{\parallel}+\zeta_y^2\,\phi_p^{\perp} & \xi_x\,\xi_y\,\phi_p^{\parallel}-\xi_x\,\xi_y\,\phi_p^{\perp}\\ \xi_x\,\xi_y\,\phi_p^{\parallel}-\xi_x\,\xi_y\,\phi_p^{\perp} & \zeta_y^2\,\phi_p^{\parallel}+\xi_x^2\,\phi_p^{\perp}\end{pmatrix}.$$

The limited tensor gradient is finally given by formula

$$\nabla\boldsymbol{U}_p^{\lim}=\Phi_p\nabla\boldsymbol{U}_p$$

and thus the limited velocity field reconstruction in the vicinity of node p is

$$\boldsymbol{U}_p(\boldsymbol{X})=\boldsymbol{U}_p+\nabla\boldsymbol{U}_p^{\lim}(\boldsymbol{X}-\boldsymbol{X}_p). \tag{5.16}$$

We claim that we have defined a tensorial limitation procedure for the velocity vector which is frame invariant and thus preserves rotational symmetry. In practice, we define the local basis utilizing the point velocity direction.

5.3.3 High-order cell-centered approximate Riemann solver

It consists in replacing the point velocity by its extrapolated value at cell center using the piecewise linear monotonic reconstruction. Namely, the system that solves $\boldsymbol{U}_c$ becomes

$$\mathsf{M}_c\boldsymbol{U}_c=\sum_{p\in\mathcal{N}(c)}\mathsf{M}_{cp}\boldsymbol{U}_p(\boldsymbol{X}_c),$$

where the swept mass fluxes entering the definition of the subcell matrices are also computed using the extrapolated velocity as

$$Z_{cp}^{\pm}=\rho_c\big[\sigma_c+c_Q\Gamma_c|(\boldsymbol{U}_c-\boldsymbol{U}_p(\boldsymbol{X}_c))\cdot\boldsymbol{N}_{cp}^{\pm}|\big].$$

The subcell force is modified accordingly:

$$\boldsymbol{F}_{cp}=-L_{cp}P_c\boldsymbol{N}_{cp}+\mathsf{M}_{cp}\big(\boldsymbol{U}_p(\boldsymbol{X}_c)-\boldsymbol{U}_c\big). \tag{5.17}$$

5.4 Subcell pressure based cell-centered Riemann solver

In Section 5.2, the cell-centered Riemann solver has been derived by considering a piecewise constant pressure inside the cell. Here we present a modification that takes into account the subcell pressures as in [11]. Indeed, the main assumption of the compatible formalism that has been used relies on the fact that subcells are Lagrangian volumes; namely subcell mass m_{cp} is constant in time. This main assumption leads to the following definition of subcell density

$$\rho_{cp}(t)=\frac{m_{cp}}{V_{cp}(t)}, \tag{5.18}$$

where $V_{cp}(t)$ is the subcell volume. To define the subcell pressure, $P_{cp}(t)$, we assume that the cell-centered specific internal energy is constant over the cell. Therefore, using the EOS, subcell pressure reads

$$P_{cp}(t)=P(\rho_{cp}(t),\varepsilon_c(t)). \tag{5.19}$$

To incorporate subcell pressure effects one substitutes P_{cp} into the half-Riemann problems that allow to define the subcell force. In other words, one replaces P_c in (5.3a)-(5.3b) by P_{cp} as follows

$$P_{cp}-\Pi_{cp}^{-}=Z_{cp}^{-}\big(\boldsymbol{U}_c-\boldsymbol{U}_p\big)\cdot\boldsymbol{N}_{cp}^{-}, \tag{5.20a}$$
$$P_{cp}-\Pi_{cp}^{+}=Z_{cp}^{+}\big(\boldsymbol{U}_c-\boldsymbol{U}_p\big)\cdot\boldsymbol{N}_{cp}^{+}. \tag{5.20b}$$

The swept mass fluxes are also modified by making use of the subcell density ρ_{cp} and sound speed σ_{cp} as

$$Z_{cp}^{\pm}=\rho_{cp}\big[\sigma_{cp}+c_Q\Gamma_c|(\boldsymbol{U}_c-\boldsymbol{U}_p)\cdot\boldsymbol{N}_{cp}^{\pm}|\big].$$

The corresponding subcell force is modified accordingly

$$\boldsymbol{F}_{cp}=-L_{cp}P_{cp}\boldsymbol{N}_{cp}+\mathsf{M}_{cp}\big(\boldsymbol{U}_p-\boldsymbol{U}_c\big). \tag{5.21}$$

Then, the system that solves the cell-centered velocity rewrites as

$$\boldsymbol{U}_c=\mathsf{M}_c^{-1}\sum_{p\in\mathcal{P}(c)}\big(\mathsf{M}_{cp}\boldsymbol{U}_p-L_{cp}P_{cp}\boldsymbol{N}_{cp}\big). \tag{5.22}$$

Let us a give an interpretation of the two terms that determine the cell-centered velocity. The first term at the right-hand side is simply a weighted interpolation of nodal velocities at cell center, whereas the second corresponds to a discretization of the pressure gradient at cell center. This interpretation is easy to obtain by computing the pressure gradient integral over the cell as

$$\big(\boldsymbol{\nabla}P\big)_c=\frac{1}{V_c}\int_{\partial\Omega_c}P\boldsymbol{N}\mathrm{d}S. \tag{5.23}$$

Using the subcell decomposition, (5.23) rewrites

$$\big(\boldsymbol{\nabla}P\big)_c=\frac{1}{V_c}\sum_{p\in\mathcal{P}(c)}\int_{\partial\Omega_{cp}\cap\partial\Omega_c}P\boldsymbol{N}\mathrm{d}S=\frac{1}{V_c}\sum_{p\in\mathcal{P}(c)}L_{cp}P_{cp}\boldsymbol{N}_{cp}.$$

With this result the cell-centered velocity reads

$$\boldsymbol{U}_c=\sum_{p\in\mathcal{P}(c)}\mathsf{M}_c^{-1}\mathsf{M}_{cp}\boldsymbol{U}_p-V_c\mathsf{M}_c^{-1}\big(\boldsymbol{\nabla}P\big)_c. \tag{5.24}$$

This formula degenerates to the previous formula (5.1) in case of uniform subcell pressure over the cell. The extra pressure gradient term induced by the subcell pressures acts as a supplementary viscous term that is usually present in approximate Riemann solver. Namely, (5.24) can be viewed as a two-dimensional generalization of an acoustic Riemann solver. Indeed, in the case of a one-dimensional flow aligned with a rectangular grid, for $c_Q=0$, one can show that the cell-centered velocity reduces to

$$\boldsymbol{U}_c=\Big(\frac{Z_l u_l+Z_r u_r}{Z_l+Z_r}-\frac{P_r-P_l}{Z_l+Z_r}\Big)\boldsymbol{e}_x,$$

where the subscripts l and r denote the left and right states of the one-dimensional Riemann problem on both sides of the interface.

We want to quote that the subcell pressure has been initially introduced in classical staggered discretization by Caramana and Shashkov [11] to control artificial grid distortions, such as the hourglass modes. Let us recall, that in the case of a logically rectangular grid, a quadrilateral cell has eight degrees of freedom. All but the two hourglass modes are physical, only for the hourglass modes does the subcell density differ from the cell density to which it belongs. The subcell pressure method uses this to calculate subcell forces that are proportional between the subcell and the cell pressures, and oppose the hourglass motion. In this approach, the subcell pressure force is defined as

$$\boldsymbol{F}_{cp}^{\Delta P}=L_{cp}(P_{cp}-P_c)\boldsymbol{N}_{cp}+\frac{1}{2}\big[(P_{cp}-P_{cp^-})L_{cp}^{-}\boldsymbol{N}_{cp}^{-}+(P_{cp}-P_{cp^+})L_{cp}^{+}\boldsymbol{N}_{cp}^{+}\big], \tag{5.25}$$

where P_{cp^-} and P_{cp^+} are the previous and next neighbor subcell pressures with respect to subcell cp.

6 Time discretization

The time discretization is performed with a classical two-step predictor-corrector scheme to gain second-order accuracy. Being given geometric quantities and physical variables at time t^n, we first predict the pressures that are later used in the corrector step to update physical and geometric variables. The full discretization in space and time is displayed in the following algorithm:

Predictor step

1. Piecewise monotonic linear reconstruction of the velocity field over the dual grid
$$\boldsymbol{U}_p^n(\boldsymbol{X})=\boldsymbol{U}_p^n+\nabla\boldsymbol{U}_p^{\lim}(\boldsymbol{X}-\boldsymbol{X}_p^n).$$
2. Compute $\boldsymbol{U}_c^n$ with the high-order cell-centered Riemann solver
$$\boldsymbol{U}_c^n=\big(\mathsf{M}_c^n\big)^{-1}\sum_{p\in\mathcal{P}(c)}\big[-\big(L_{cp}\boldsymbol{N}_{cp}\big)^n P_{cp}^n+\mathsf{M}_{cp}^n\boldsymbol{U}_p^n(\boldsymbol{X}_c^n)\big].$$
3. Compute subcell forces
$$\boldsymbol{F}_{cp}^n=-\big(L_{cp}\boldsymbol{N}_{cp}\big)^n P_{cp}^n+\mathsf{M}_{cp}^n\big(\boldsymbol{U}_p^n(\boldsymbol{X}_c^n)-\boldsymbol{U}_c^n\big).$$
4. Update internal energy
$$m_c\big(\varepsilon_c^{n+\frac{1}{2}}-\varepsilon_c^n\big)-\frac{\Delta t}{2}\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}^n\cdot\boldsymbol{U}_p^n=0.$$
5. Update vertex position
$$\boldsymbol{X}_p^{n+\frac{1}{2}}=\boldsymbol{X}_p^n+\frac{\Delta t}{2}\boldsymbol{U}_p^n.$$
6. Update volume and density
$$\rho_c^{n+\frac{1}{2}}=\frac{m_c}{V_c^{n+\frac{1}{2}}}, \qquad \rho_{cp}^{n+\frac{1}{2}}=\frac{m_{cp}}{V_{cp}^{n+\frac{1}{2}}}.$$
7. Compute predicted pressures
$$P_c^{n+\frac{1}{2}}=P\big(\rho_c^{n+\frac{1}{2}},\varepsilon_c^{n+\frac{1}{2}}\big), \qquad P_{cp}^{n+\frac{1}{2}}=P\big(\rho_{cp}^{n+\frac{1}{2}},\varepsilon_{cp}^{n+\frac{1}{2}}\big).$$

Corrector step

1. Piecewise monotonic linear reconstruction of the velocity field over the dual grid
$$\boldsymbol{U}_p^{n+\frac{1}{2}}(\boldsymbol{X})=\boldsymbol{U}_p^n+\nabla\boldsymbol{U}_p^{\lim}(\boldsymbol{X}-\boldsymbol{X}_p^{n+\frac{1}{2}}).$$
2. Compute $\boldsymbol{U}_c^{n+1/2}$ with the high-order cell-centered Riemann solver
$$\boldsymbol{U}_c^{n+\frac{1}{2}}=\big(\mathsf{M}_c^{n+\frac{1}{2}}\big)^{-1}\sum_{p\in\mathcal{P}(c)}\big[-\big(L_{cp}\boldsymbol{N}_{cp}\big)^{n+\frac{1}{2}}P_{cp}^{n+\frac{1}{2}}+\mathsf{M}_{cp}^{n+\frac{1}{2}}\boldsymbol{U}_p^{n+\frac{1}{2}}(\boldsymbol{X}_c^{n+\frac{1}{2}})\big].$$
3. Compute subcell forces
$$\boldsymbol{F}_{cp}^{n+\frac{1}{2}}=-\big(L_{cp}\boldsymbol{N}_{cp}\big)^{n+\frac{1}{2}}P_{cp}^{n+\frac{1}{2}}+\mathsf{M}_{cp}^{n+\frac{1}{2}}\big[\boldsymbol{U}_p^{n+\frac{1}{2}}(\boldsymbol{X}_c^{n+\frac{1}{2}})-\boldsymbol{U}_c^{n+\frac{1}{2}}\big].$$

4. Update momentum

$$m_p(\boldsymbol{U}_p^{n+1}-\boldsymbol{U}_p^n)+\Delta t\sum_{c\in\mathcal{C}(p)}\boldsymbol{F}_{cp}^{n+\frac{1}{2}}=\boldsymbol{0}.$$

5. Update internal energy

$$m_c\left(\varepsilon_c^{n+1}-\varepsilon_c^n\right)-\Delta t\sum_{p\in\mathcal{P}(c)}\boldsymbol{F}_{cp}^{n+\frac{1}{2}}\cdot\boldsymbol{U}_p^{n+\frac{1}{2}}=0,$$

with $\boldsymbol{U}_p^{n+1/2}=(\boldsymbol{U}_p^{n+1}+\boldsymbol{U}_p^n)/2$.

6. Update vertex position

$$\boldsymbol{X}_p^{n+1}=\boldsymbol{X}_p^n+\Delta t\boldsymbol{U}_p^{n+\frac{1}{2}}.$$

7. Update volume and density

$$\rho_c^{n+1}=\frac{m_c}{V_c^{n+1}},\qquad \rho_{cp}^{n+1}=\frac{m_{cp}}{V_{cp}^{n+1}}.$$

8. Compute pressures

$$P_c^{n+1}=P\left(\rho_c^{n+1},\varepsilon_c^{n+1}\right),\qquad P_{cp}^{n+1}=P\left(\rho_{cp}^{n+1},\varepsilon_{cp}^{n+1}\right).$$

We point out that in the corrector step, internal energy has been discretized using the time centered nodal velocity $\boldsymbol{U}_p^{n+1/2}$. This choice is required to ensure total energy conservation up to machine precision.

7 Numerical results

The 2D Cartesian geometry is chosen for all test cases which are mono-material simulations. Only the ideal gas EOS is employed even though the framework accepts any type of EOS. We provide results of the following test cases: the Sod shock tube, the Cartesian Sedov problem, the Saltzman piston problem, the Noh problem and a Richtmyer-Meshkov instability. The high-order scheme is used for every test except the Sod shock tube for which a comparison between first and high-order is provided.

7.1 Sod shock tube

The Sod problem is a 1D Riemann shock tube with a relatively mild discontinuity. Its solution consists of a left moving rarefaction fan, a right moving contact discontinuity and a right moving shock wave. The domain is filled with an ideal gas at rest with $\gamma=1.4$. The density/pressure values on the left side of the discontinuity are 1.0/1.0, while those on the right side are 0.125/0.1. The discontinuity is initially located at 0.5.

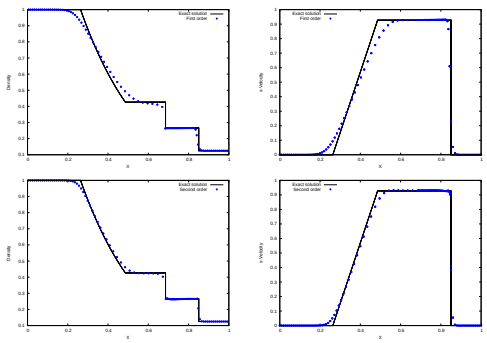


Figure 3: Sod problem at $t_{\text{final}}=0.2$ for 200 cells in x-direction. Top: first order scheme. Bottom: high order scheme. From left to right: Cell-centered density, nodal velocity in x-direction as a function of x.

We simulate this problem up to the final time $t_{\text{final}}=0.2$. In our numerical experiments, the computational domain is $\Omega=[0,1]\times[0,y_{\max}]$, where $y_{\max}$ is chosen so that the cells are initially squares. We are enforcing zero normal velocity at left and right of Ω and slip boundary at top and bottom. The run is made with $n_x=200$ computational cells in x direction and $n_y=10$ cells in y direction leading to $y_{\max}=0.2$.

In Fig. 3 is presented the first order (top) and high order extension (bottom). We display the cell density for all cells and the nodal velocity for all nodes *vs* the exact solution shown by solid line. The symmetry of the scheme is perfect and the quality of the high order results is close to the cell-centered Lagrangian scheme [26]. The first order scheme presents classical associated behavior-three-to-five cell shock spreading-whereas the shock wave is spread only on one or two cells for the second-order extension. In the compatible staggered Lagrangian scheme with classical artificial viscosity the shock is generally spread over three to five cells. Moreover the tail of the rarefaction does not suffer from the classical undershoot that can usually be seen on classical compatible staggered Lagrangian scheme.

7.2 Sedov blast wave problem

In this subsection we present the Sedov blast wave problem [31], which describes the evolution of a blast wave in a point-symmetric explosion. It is an example of a diverging

shock wave. We consider the cylindrically symmetric Sedov problem in Cartesian geometry. The total energy of the explosion is concentrated at the origin and has magnitude $E_{\text{total}}=0.244816$ similar to [22]. The material is an ideal gas with $\gamma=1.4$ and initially is at rest with an initial density equal to 1.0.

At final time $t_{\text{final}}=1.0$ the exact solution is a cylindrically symmetric diverging shock whose front is at radius $r=\sqrt{x^2+y^2}=1$ and has a density peak $\rho=6.0$. An exact solution is available as instance in [19]. In our numerical experiments E_{total} is concentrated in one cell located at the origin (that is, containing the vertex $(x,y)=(0,0)$). The specific internal energy of this cell, c, is defined as $\varepsilon_c=E_{\text{total}}/V_c$. Therefore the initial pressure for this cell is $p=(\gamma-1)\rho\varepsilon=0.4E_{\text{total}}/V_c$.

The high-order scheme is used. First we use a Cartesian grid of 30×30 cells on domain $\Omega=[0,1.2]\times[0,1.2]$ (see Fig. 4(a)). Next we use a polygonal grid (Voronoi tessellation) of a quarter of the disk of radius 1.2 (see Fig. 4(c)), this mesh has 775 cells being various polygons. In Fig. 4(a) are shown the mesh and the density colormap. The general geometric mesh quality is good. Moreover the symmetry of the shock wave is nicely preserved on both these meshes. Indeed the good symmetry preservation is shown on Fig. 4(b), (d) where the cell density as a function of cell radius is displayed for all cells in the domain together with the exact solution; not only the shock wave is very sharp but the cells are well distributed onto the exact curve.

7.3 Noh problem

In a quarter of the unit disk a gas ($\gamma=5/3$) is initiated with

$$\rho_0=1,\quad \varepsilon_0=10^{-6}\quad\text{and}\quad \mathbf{U}(x,y)=\left(\frac{-x}{\sqrt{x^2+y^2}},\frac{-y}{\sqrt{x^2+y^2}}\right).$$

A cylindrical shock wave is generated at the origin and further diverges. The final time is chosen at $t_{\text{final}}=0.6$. The exact solution at radius r and time t is given by the following relations, in which d identifies the geometry of the problem (1 for 1D Cartesian, 2 for 1D cylindrical, 3 for 1D spherical), ρ_0 is the uniform initial density, and u_0 is the uniform radial velocity ($u_0=\|\mathbf{U}\|=1$):

$$\{\rho,\varepsilon,u_r\}=\begin{cases}\left\{\rho_0\left(\frac{\gamma+1}{\gamma-1}\right)^d,\frac{1}{2}(u_0)^2,0\right\}, & \text{if } r<r_s,\\ \left\{\rho_0\left(1-\frac{u_0t}{r}\right)^d,0,u_0\right\}, & \text{if } r>r_s,\end{cases}\tag{7.1}$$

where the position of the shock r_s is given by

$$r_s=U_st,\ \text{with the shock speed}\ U_s=\frac{1}{2}(\gamma-1)|u_0|.\tag{7.2}$$

The exact solution is given by ($r_s=0.2$, $U_s=1/3$)

$$\{\rho,\varepsilon,u_r\}=\begin{cases}\left\{16,\frac{1}{2},0\right\}, & \text{if } r<0.2,\\ \left\{\left(1+\frac{3}{5}\frac{1}{r}\right),0,1\right\}, & \text{if } r>0.2.\end{cases}\tag{7.3}$$

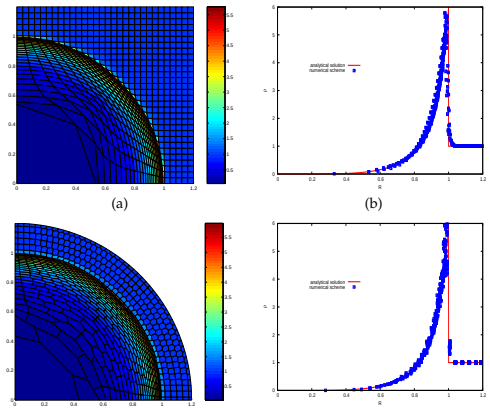


Figure 4: Sedov problem at $t_{\text{final}}=1$. (a) Density map and mesh on a 30×30 Cartesian grid. (b) Cell density as a function of cell center radius *vs* exact solution on a 30×30 Cartesian grid. (c) Density map and mesh on a polygonal grid. (d) Cell density as a function of cell center radius *vs* exact solution on a polygonal grid.

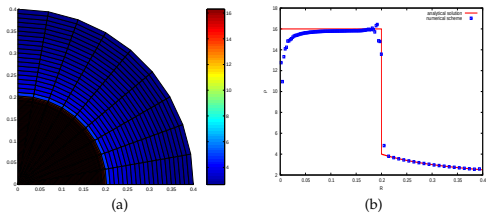


Figure 5: Noh problem at $t_{\text{final}}=0.6$ on a 100×9 polar grid. (a) Density map and mesh. (b) Cell density as a function of cell center radius *vs* exact solution.

This problem is simulated in its Cartesian version. First a polar mesh with 100 cells in radial direction and 9 cells in angular direction is considered. Fig. 5 shows the final mesh and density colormap on panel (a) and the cell density as a function of cell center radius compared to exact solution on panel (b). The symmetry of the shock wave and the mesh is almost perfect. Moreover the accuracy of the plateau is good and one observes a classical wall heating effect close to the origin. A slight overshoot can be noticed just after the shock wave which is, however, very sharp.

One classical issue of using polar mesh is the fact that to get a cell aspect ratio close to one on the external border of the disk we have to mesh the region close to the origin with triangles that have tremendously small internal angle. This leads not only to very big number of cells, but also to inaccuracy and sensitivity while discretizing equations on such long and a thin triangles, and last but not least, a small time step. A solution consists in using a non-conformal mesh as depicted in Fig. 6 panel (a); the mesh is made of quadrangles and every second angular line is stopped at a given radius ($r=0.3$ in the figure), leading to pentagonal cells. Such lines and associated cells are elsewhere referred as to termination lines [21] or dendritic cells. The extra nodes that are created on these pentagons by the termination lines are usually enslaved to their edge. A special treatment is set-up for these pentagonal cells [21] in classical compatible Lagrangian hydrodynamics scheme. This treatment is based on distribution of masses and forces on surrounding zones. In our approach, as for the cell-centered Lagrangian scheme [26], no special treatment is needed. The Riemann solver is naturally producing the correct point velocity (see Fig. 6(b)) contrary to the classical approach with no special treatment (see Fig. 9(a) on page 11 of [21]). Although some mesh imprint can be seen, the cell density as a function of cell center radius in Fig. 6(c) is still very well reproduced.

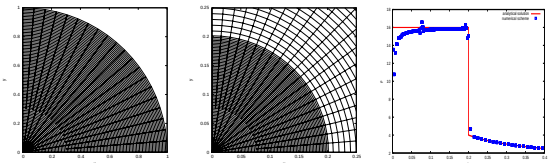
7.4 Saltzman piston

The domain $\Omega=[0,1]\times[0,0.1]$ is filled with a gas ($\gamma=5/3$) at rest $\rho=1$, $\varepsilon=10^{-6}$. The right boundary is a perfect wall, the left boundary is a right-moving piston with velocity $\boldsymbol{U}=(1,0)^t$. This piston sends a straight shock wave into the rectangular domain. This shock wave ultimately bounces onto the fixed right wall and onto the piston back and forth. At time $t=1.0$ the piston reaches the right wall.

An exact solution is defined by the value of the plateaus behind the shock. The analytical solution is characterized by a post-shock density plateau of 4 and a shock velocity of $4/3$ before the time $t=0.75$. This problem usually tests the robustness of Lagrangian numerical methods by using the Saltzman mesh defined by $n_x=100$ cells in x-direction and $n_y=10$ in y-direction and a deformation in x-direction as

$$x_{\text{deformed}}=x+(0.1-y)\sin(\pi x),$$

as can be seen in Fig. 7. In Fig. 8 is presented the mesh and the density colormap at time $t_{\text{final}}=0.6$. Unfortunately, our approach is not able to perform up to time $t>0.925$ as the top left corner cells are pinched and become tangled soon after this time. This



(a) (b) (c)

Figure 6: Noh problem at $t_{\text{final}}=0.6$ on a non-conformal grid. (a) Initial mesh. (b) Zoom on the final mesh. (c) Cell density as a function of cell center radius *vs* exact solution.

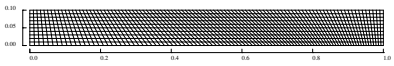


Figure 7: Saltzman piston. Initial skewed mesh.

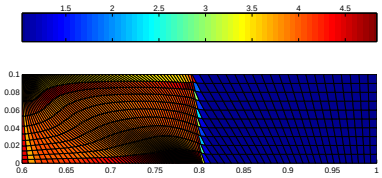


Figure 8: Saltzman piston at time $t_{\text{final}}=0.6$. Final mesh and density colormap.

quite disappointing result asks for more investigation on robustness issue and also on boundary condition treatment using staggered discretization.

7.5 Linear phase of Richtmyer-Meshkov instability

This test case is devoted to the study of the linear phase of the Richtmyer-Meshkov instability [30] for a piston-driven flow. This hydrodynamic instability occurs when a shock wave hits a perturbed interface separating two different fluids. For sufficiently small perturbations, analytical solutions can be derived using linear perturbation theory [36]. In this framework, the theory shows that the amplitude of the perturbation grows linearly as a function of time. We first study the unperturbed fluid configuration in the RMI prob-

lem, which is a collision of a shock wave with a flat contact discontinuity. Such collision produces a transmitted shock wave and a reflected wave that can be either a shock or a rarefaction. This shock-contact interaction defines a one-dimensional Riemann problem, which can be solved analytically.

In what follows, we will employ the configuration displayed in Fig. 9. The interface is located at $x=0$ and the computational domain corresponding to the shock tube is defined by $(x,y)\in[-5,4.2]\times[0,0.5]$, since the $y=0$ line is a symmetry axis for this problem. For the initial and boundary conditions described in Fig. 9, the incident piston-driven shock hits the interface at time $t=3.015$. This interaction leads to transmitted and reflected shock waves, which also later interact with the piston and the right boundary wall. The time history of the shock-contact interaction is displayed in Fig. 10(a) using a classical $(t-x)$ diagram. We run a computation for the unperturbed configuration with our high-order scheme using 460 equally spaced cells in the x-direction and one cell in the y-direction. The density as function of x-coordinate is plotted in Fig. 10(b) together with analytical solution at time $t=5$. We point out the very good agreement between numerical and analytical solutions. Moreover, we note that transmitted and reflected shocks are sharply resolved. In order to study the perturbed configuration, we initialize a cosinusoidal perturbation of the interface with a small amplitude a_0. Thus, the equation of the interface is written

$$x(y)=a_0\cos\Big(\frac{2\pi}{\lambda}y\Big),\quad \text{for } y\in\Big[-\frac{\lambda}{2},\frac{\lambda}{2}\Big], \tag{7.4}$$

where λ is the wavelength of the perturbation. The shape of the perturbed interface is displayed in Fig. 9. For a small enough initial amplitude, linear theory predicts that the perturbation amplitude, $a(t)$, grows linearly as a function of time, after the shock has interacted with the interface. Using direct two-dimensional simulation of the perturbed configuration, we shall recover this important result and compare the numerical perturbation amplitude with the one coming from the linear theory. The numerical simulations are made by meshing the computational domain, $(x,y)\in[-5,4.2]\times[0,0.5]$, with 460×25 equally spaced cells. Hence, we have set $\lambda=1$ and meshed only a half wavelength due to the symmetry of the problem w.r.t. x-axis. We set $a_0=10^{-4}$ and performed a computation utilizing our high-order scheme. The perturbed interface is prescribed by moving the vertices initially located on the line $x=0$ onto the curve defined by Eq. (7.4). The perturbation amplitude, $a(t)$, is computed using the following formula

$$a(t)=\frac{X_{\text{pert}}(t)-X_{\text{unpert}}(t)}{a_0}, \tag{7.5}$$

where $X_{\text{pert}}(t)$ (resp. $X_{\text{unpert}}(t)$) is the abscissa of a point located on the perturbed (resp. unperturbed) interface. Using this formula for the previous computations, we compute the corresponding perturbation amplitude and compare it to the reference one coming from the linear theory [36]. We have plotted in Fig. 11 the numerical perturbation amplitude as a function of time compared to the one coming from the linear theory. We

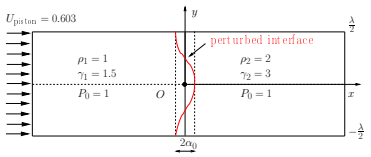
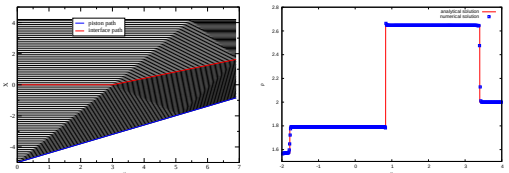


Figure 9: Richtmyer-Meshkov instability initialization.



(a) (b)

Figure 10: Richtmyer-Meshkov instability. (a) Time-X diagram where piston and interface paths are displayed. (b) Density as a function of X coordinate *vs* analytical solution at $t=5$.

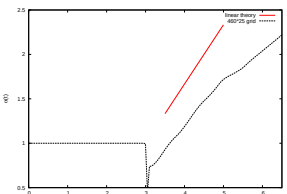


Figure 11: Shock-contact interaction problem. Numerical perturbation amplitude as a function of time *vs* linear theory.

remark that the high-order computation recovers quite well the linear theory. These results show the ability of our high-order Lagrangian scheme to simulate very accurately complex phenomena such as hydrodynamic instabilities.

8 Conclusions

This work suggests a general formalism to derive staggered discretizations for Lagrangian hydrodynamics on general unstructured meshes in two dimensions. The framework uses fundamental objects of compatible discretizations like Lagrangian subcell mass and subcell forces. Artificial viscosity form is formulated invoking Galilean invariance and thermodynamic consistency. The satisfaction of entropy inequality is ensured by using a subcell-based symmetric positive definite tensor, a particular example of which is given in this work. An extension to high order in space and time is demonstrated, including a vector limitation procedure which is frame independent and thus preserves desirable properties like rotational symmetry. Performance of the new method is demonstrated on a set of classical and demanding numerical tests, using various structured and unstructured computational meshes.

An important potential link between staggered and cell-centered Lagrangian schemes is the viscous part of subcell force. While some of the existing artificial viscosity implementations can be easily reformulated by means of the proposed symmetric positive definite tensor, others still seem to resist this simple interpretation. From this viewpoint there remains enough space for deeper investigation with the prospect of finding important similarities and differences between the Godunov-based and Lagrange-based methods.

Another promising idea to be explored is the use of a generalized Riemann problem [5] for simultaneous discretization in space and time, which would replace predictor-corrector temporal integration by a more elegant one-step scheme.

Last but not least, the plan is to extend all pieces of the existing machinery to two-dimensional axisymmetric [25] and further to three-dimensional geometry and thus open it for new practical applications.

Acknowledgments

The authors would like to thank Milan Kucharík, Richard Liska, Misha Shashkov and Burt Wendroff for fruitful discussions. The last author has been partly supported by the Czech Ministry of Education grants MSM 6840770022, MSM 6840770010, LC528 and the Czech Science Foundation grant P205/10/0814.

Appendix: 1D staggered Lagrangian scheme based on cell-centered Riemann solver

In this appendix we detail the 1D version of the numerical scheme proposed in this paper. In order to be complete we fully describe its derivation. The governing equations are

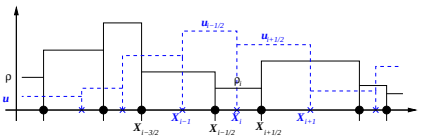


Figure 12: Notation for the 1D scheme description.

written in 1D as

$$\rho\frac{d}{dt}\left(\frac{1}{\rho}\right)-\frac{\partial}{\partial X}u=0,\qquad \rho\frac{d}{dt}u+\frac{\partial}{\partial X}P=0,\qquad \rho\frac{d}{dt}E+\frac{\partial}{\partial X}(Pu)=0. \tag{A.1}$$

Physical variables are expressed as functions of $X(t)$, which is the coordinate of a fluid particle at time t whose initial position is denoted by x. The trajectory equation then writes

$$\frac{d}{dt}X=u(X(t),t),\qquad X(0)=x. \tag{A.2}$$

A cell is labeled with an integer index as I_i and is the segment $I_i(t)=[X_{i-1/2}(t);X_{i+1/2}(t)]$ constituted of two consecutive points labeled with half-integers, see Fig. 12. $I=\bigcup_i I_i$ is a partition of the 1D computational domain. We denote $V_i(t)=X_{i+1/2}(t)-X_{i-1/2}(t)$ the volume of cell I_i. The displacement of the vertex $X_{i+1/2}(t)$ is given by the semi-discrete trajectory equation

$$\frac{d}{dt}X_{i+\frac{1}{2}}=u_{i+\frac{1}{2}},\qquad X_{i+\frac{1}{2}}(0)=x_{i+\frac{1}{2}}, \tag{A.3}$$

where $u_{i+1/2}=u(X_{i+1/2}(t),t)$ is the trajectory of the vertex, namely the fluid velocity at point $X_{i+1/2}$. Since cell I_i moves with the fluid velocity it may deform but can neither gain nor lose mass, thus the mass of cell I_i writes

$$m_i(t)=\int_{I_i(t)}\rho(X(t),t)\,\mathrm{d}X=\int_{I_i(0)}\rho(x,0)\,\mathrm{d}x=m_i(0), \tag{A.4}$$

where $I_i(0)=[x_{i-1/2},x_{i+1/2}]$ is the initial cell and $\rho(x,0)$ the initial density. Let

$$X_i(t)=\frac{1}{2}\left(X_{i-\frac{1}{2}}(t)+X_{i+\frac{1}{2}}(t)\right)$$

be the center of cell I_i. The median mesh is defined by the introduction of the dual cell $I_{i+1/2}(t)=[X_i(t),X_{i+1}(t)]$. Knowing that the cell center is computed in a Lagrangian manner we deduce that the mass of the dual cell, $m_{i+1/2}$, is constant and writes $m_{i+1/2}=(m_i+m_{i+1})/2$.

Volume conservation equation. Let $\rho_i(t)$ be the cell-averaged density

$$\rho_i(t)=\frac{1}{V_i(t)}\int_{I_i(t)}\rho(X(t),t)\mathrm{d}X. \tag{A.5}$$

It can also be written $\rho_i(t)=m_i/V_i(t)$ and therefore

$$m_i\frac{d}{dt}\left(\frac{1}{\rho_i}\right)=\frac{d}{dt}V_i=u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}, \tag{A.6}$$

which is the semi-discrete version of the volume conservation equation (A.1).

Momentum conservation equation. The semi-discrete momentum equation over the dual cell writes

$$m_{i+\frac{1}{2}}\frac{d}{dt}u_{i+\frac{1}{2}}+\left(P^*_{i+1}-P^*_i\right)=0, \tag{A.7}$$

where $P^*_i=P(X_i(t),t)$ represents the pressure at zone center X_i that remains to be determined.

Total energy conservation. Without taking into account boundary terms we introduce the total kinetic energy at time $t>0$ as

$$\mathcal{K}=\sum_{i+\frac{1}{2}}\frac{1}{2}m_{i+\frac{1}{2}}u^2_{i+\frac{1}{2}},$$

where the sum is performed over the dual cells $I_{i+1/2}$. We also introduce the total internal energy at t as $\mathcal{E}=\sum_i m_i\varepsilon_i$, where the sum is taken over the primal cells I_i and ε_i denotes the cell-averaged internal energy. Conservation of total energy writes

$$\frac{d}{dt}\mathcal{K}+\frac{d}{dt}\mathcal{E}=0. \tag{A.8}$$

Recalling that masses are Lagrangian objects, the previous equation rewrites

$$\sum_{i+\frac{1}{2}}\frac{1}{2}m_{i+\frac{1}{2}}u_{i+\frac{1}{2}}\frac{d}{dt}u_{i+\frac{1}{2}}=-\sum_i m_i\frac{d}{dt}\varepsilon_i. \tag{A.9}$$

Substituting the semi-discrete momentum equation previously derived we get

$$\sum_{i+\frac{1}{2}}u_{i+\frac{1}{2}}(P^*_{i+1}-P^*_i)=\sum_i m_i\frac{d}{dt}\varepsilon_i, \tag{A.10}$$

which yields after shifting the index of the first sum

$$\sum_i P^*_i\left(u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}\right)=\sum_i m_i\frac{d}{dt}\varepsilon_i. \tag{A.11}$$

Without taking into account the boundary terms, the total energy conservation writes

$$\sum_i m_i\frac{d}{dt}\varepsilon_i+P^*_i\left(u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}\right)=0. \tag{A.12}$$

A sufficient condition to ensure total energy conservation is obtained by writing the following evolution equation for the internal energy within cell I_i

$$m_i\frac{d}{dt}\varepsilon_i+P^*_i\left(u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}\right)=0. \tag{A.13}$$

The only unknown to determine is the pressure P^*_i. Let us note that using the volume conservation equation this last equation can be recast into the form

$$m_i\frac{d}{dt}\varepsilon_i+m_iP^*_i\frac{d}{dt}\left(\frac{1}{\rho_i}\right)=0, \tag{A.14}$$

which, recalling the Gibbs formula $TdS=d\varepsilon+Pd(\frac{1}{\rho})$, rewrites

$$m_iT_i\frac{d}{dt}S_i=-m_i(P^*_i-P_i)\frac{d}{dt}\left(\frac{1}{\rho_i}\right), \tag{A.15}$$

where S_i and T_i are the specific entropy and temperature in cell I_i. The right-hand side of this equation is nothing but the time rate of change of specific entropy dissipation. According to the second law of thermodynamics $T_i\frac{d}{dt}S_i\geq 0$, thus we should have

$$(P^*_i-P_i)\frac{d}{dt}\left(\frac{1}{\rho_i}\right)\leq 0.$$

Consequently for smooth flows, characterized by reversible thermodynamical process, such as rarefaction wave or isentropic compression, we should have $P^*_i=P_i$ since $\frac{d}{dt}(\frac{1}{\rho_i})<0$. The difference $P^*_i-P_i$ is the artificial viscosity first introduced by von Neumann and Richtmyer [33]. Let us now describe how to compute this term by solving a staggered Riemann problem. First the interfacial pressure P^*_i is computed at each cell center X_i by solving the Riemann problem characterized by the left state $(\rho_i,u_{i-1/2},P_i)$ and the right one $(\rho_i,u_{i+1/2},P_i)$. Note that only the velocity is discontinuous with a jump

$$\Delta u=u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}.$$

In order to simplify the notation let us denote with subscript L the left state (ρ,u_L,P). Accordingly the right state is (ρ,u_R,P) with subscript R and the jump in velocity is $\Delta u=u_R-u_L$. We note that the equation of state is the same both sides of the interface. The solution of the Riemann problem is a two-shock solution in the case that $\Delta u<0$ (see Fig. 13 top-right panel) and a two-rarefaction solution if $\Delta u>0$ (see Fig. 13 bottom-right

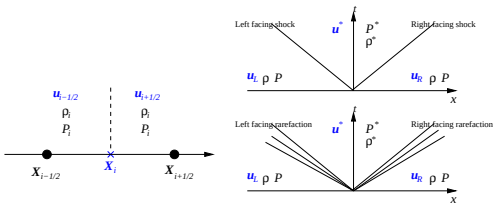


Figure 13: Left: States for the Riemann problem at cell center. Top-right: Two-shock Riemann solution. Bottom-right: Two-rarefaction Riemann solution.

panel). The solution is denoted (ρ^*,u^*,P^*). Since thermodynamical variables are continuous across the interface, there is no contact discontinuity.

Two-shock case. The Rankine-Hugoniot relations allow to compute the post-shock pressure and velocity. They write for the right-facing shock

$$M\Big(\frac{1}{\rho^*}-\frac{1}{\rho}\Big)=-(u^*-u_R), \tag{A.16a}$$
$$M(u^*-u_R)=P^*-P, \tag{A.16b}$$
$$\varepsilon^*-\varepsilon+\frac{P^*+P}{2}\Big(\frac{1}{\rho^*}-\frac{1}{\rho}\Big)=0, \tag{A.16c}$$

where $M>0$ denotes the mass swept by the wave per unit time. For the left-facing shock we only write

$$M\Big(\frac{1}{\rho^*}-\frac{1}{\rho}\Big)=-(u^*-u_L), \tag{A.17a}$$
$$M(u^*-u_L)=-(P^*-P), \tag{A.17b}$$

recalling that the third equation is exactly the same as (A.16c) since it is independent on the direction of propagation. Adding (A.16b) and (A.17b) yields $u^*=(u_R+u_L)/2$, hence using (A.16b) and the definition of Δu we get

$$P^*=P-\frac{1}{2}M\Delta u. \tag{A.18}$$

Using cell notation we deduce

$$P_i^*=P_i-\frac{1}{2}M\big(u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}\big), \tag{A.19}$$

thus the internal energy equation reads

$$m_i\frac{d}{dt}\varepsilon_i+P_i\big(u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}\big)=\frac{1}{2}M_i\big(u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}\big)^2. \tag{A.20}$$

This discretization is compatible with the second principle of thermodynamics.

The swept mass flux can be expressed as function of the physical variables by combining (A.16a) and (A.16b)

$$M^2=-(P^*-P)\Big(\frac{1}{\rho^*}-\frac{1}{\rho}\Big)^{-1}. \tag{A.21}$$

In the limit of the weak shock wave $P^*\to P$ and $\rho^*\to\rho$ and one can show that $M\to\rho c$ where $c^2=(\frac{\partial P}{\partial\rho})_s$ is the isentropic sound speed. In the general case for a γ gas law one can compute the analytical solution of the Riemann problem. Knowing that $P=(\gamma-1)\rho\varepsilon$ and using (A.16c) one gets

$$\frac{1}{\rho^*}=\frac{1}{\rho}\frac{(\gamma+1)P+(\gamma-1)P^*}{(\gamma+1)P^*+(\gamma-1)P}. \tag{A.22}$$

After substitution in the equation for M^2 we obtain

$$M^2=\rho\Big(\frac{\gamma+1}{2}P^*+\frac{\gamma-1}{2}P\Big), \tag{A.23}$$

thus using (A.16b) we show that P^* satisfies a quadratic equation whose physical solution is

$$P^*=P+\rho\Big(\frac{\gamma+1}{4}\frac{\Delta u}{2}-\sqrt{\Big(\frac{\gamma+1}{4}\Big)^2\Big(\frac{\Delta u}{2}\Big)^2+c^2}\Big)\frac{\Delta u}{2}. \tag{A.24}$$

This last equation allows to write

$$M=\rho\Big(-\frac{\gamma+1}{4}\frac{\Delta u}{2}+\sqrt{\Big(\frac{\gamma+1}{4}\Big)^2\Big(\frac{\Delta u}{2}\Big)^2+c^2}\Big). \tag{A.25}$$

Let us remark that this last equation recovers the Kuropatenko artificial viscosity [20] up to a factor one half in front of the velocity jump. Kuropatenko has in fact derived his formula computing the pressure jump produced by only one shock wave created by a velocity jump Δu. Here we are considering the solution of the Riemann problem corresponding to an initial velocity jump Δu. This initial discontinuity breaks up into two shock waves, each being characterized by a velocity jump of $\Delta u/2$. The same conclusion has been given by Luttwak and Falcovitz [24].

In the case of real material we propose the following ansatz for the swept mass flux

$$M=\rho\Big(-c_0\frac{\Delta u}{2}+c_1c\Big), \tag{A.26}$$

where c_0 and c_1 are constants depending on material properties. This amounts to pretend that the velocity of the shock wave is an affine function of the fluid velocity. This assumption is satisfied for many materials as it has been noticed by Dukowicz [16]. Returning to cell notation we finally get

$$P_i^*=P_i-\frac{1}{2}M_i\big(u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}\big), \tag{A.27}$$

where

$$M_i=\rho_i\Big(-c_0\frac{1}{2}\big(u_{i+\frac{1}{2}}-u_{i-\frac{1}{2}}\big)+c_1c_i\Big). \tag{A.28}$$

Let us remark that this form of swept mass flux is the one used in (5.4) in 2D. Moreover, in 1D, the subcell matrix (5.7) is a 1×1 matrix that indeed coincides with the swept mass flux, hence the notation M_i in the equation above. Finally this procedure can be viewed as the solution of the Riemann problem using an approximate solver.

Two-rarefaction case. In the case $\Delta u\geq0$ we simply can set $P_i^*=P_i$ in order to satisfy entropy conservation. It consists in canceling the entropy production in the case of rarefaction wave. This is the usual way of proceeding when dealing with classical artificial viscosity. However it is possible also to solve the Riemann problem in this case. Since $\Delta u\geq0$, the initial discontinuity breaks up into two rarefaction waves (see Fig. 13 bottom-right panel). The Riemann problem is solved using Riemann invariants:

- for the right-facing wave $u^*-u_R=\int_\rho^{\rho^*}\frac{d\overline{P}}{\rho(\overline{P})c(\overline{P})}$,
- for the left-facing wave $u^*-u_L=-\int_\rho^{\rho^*}\frac{d\overline{P}}{\rho(\overline{P})c(\overline{P})}$,

and by adding these last equations we get $u^*=(u_R+u_L)/2$. In the case of a γ gas law one can integrate the Riemann invariants to get

$$P^*=P\Big(1-\frac{\gamma-1}{2}\frac{\Delta u}{2c}\Big)^{\frac{2\gamma}{\gamma-1}}, \tag{A.29}$$

that is valid for

$$\frac{\Delta u}{2}\leq\frac{2}{\gamma-1}c.$$

We can rewrite the last equation as

$$P^*=P+P\Big[\Big(1-\frac{\gamma-1}{2}\frac{\Delta u}{2c}\Big)^{\frac{2\gamma}{\gamma-1}}-1\Big]=P-\frac{1}{2}M\Delta u, \tag{A.30}$$

where

$$M=P\Big[-\Big(1-\frac{\gamma-1}{2}\frac{\Delta u}{2c}\Big)^{\frac{2\gamma}{\gamma-1}}+1\Big]\Big(\frac{1}{2}\Delta u\Big)^{-1}>0. \tag{A.31}$$

In the limit of weak shock wave $\Delta u\to0$ and thus $M\to\rho c$. For real materials one can derive an approximate solution using the following quadrature for the integrals for the Riemann invariants

$$\int_\rho^{\rho^*}\frac{d\overline{P}}{\rho(\overline{P})c(\overline{P})}=\frac{P^*-P}{\rho c}, \tag{A.32}$$

hence u^* and P^* are solution of the 2×2 linear system

$$\rho c(u^*-u_R)=P^*-P,\qquad \rho c(u^*-u_L)=-(P^*-P), \tag{A.33}$$

in particular $u^*=(u_L+u_R)/2$ and $P^*=P-\rho c\frac{\Delta u}{2}$. This corresponds to the acoustic approximation.

Time discretization. A two-step Runge-Kutta discretization is used. Knowing all physical quantities at time t^n, we advance them up to time $t^{n+1}=t^n+\Delta t$.

- **Predictor step.** One computes

$$P_i^{*,n}=P_i^n-\frac{1}{2}M_i^n\Delta u_i^n,$$

and solves the internal energy equation

$$m_i(\varepsilon_i^{n+\frac{1}{2}}-\varepsilon_i^n)+\frac{\Delta t}{2}P_i^{*,n}\big(u_{i+\frac{1}{2}}^n-u_{i-\frac{1}{2}}^n\big)=0. \tag{A.34}$$

Then the mesh nodes are displaced:

$$X_{i+\frac{1}{2}}^{n+\frac{1}{2}}=X_{i+\frac{1}{2}}^n+\frac{\Delta t}{2}u_{i+\frac{1}{2}}^n.$$

The density is then computed:

$$\rho_{i+\frac{1}{2}}^{n+\frac{1}{2}}=m_i(V_i^{n+\frac{1}{2}})^{-1},$$

where

$$V_i^{n+\frac{1}{2}}=X_{i+\frac{1}{2}}^{n+\frac{1}{2}}-X_{i-\frac{1}{2}}^{n+\frac{1}{2}}.$$

Finally the predicted pressure is computed

$$P_i^{n+\frac{1}{2}}=P(\rho_i^{n+\frac{1}{2}},\varepsilon_i^{n+\frac{1}{2}}).$$

- **Corrector step.** One computes

$$P_i^{*,n+\frac{1}{2}}=P_i^{n+\frac{1}{2}}-\frac{1}{2}M_i^{n+\frac{1}{2}}\Delta u_i^n,$$

and solves momentum equation

$$m_{i+\frac{1}{2}}\Big(u_{i+\frac{1}{2}}^{n+1}-u_{i+\frac{1}{2}}^n\Big)+\Delta t\Big(P_{i+1}^{*,n+\frac{1}{2}}-P_i^{*,n+\frac{1}{2}}\Big)=0, \tag{A.35}$$

and computes

$$u_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left(u_{i+\frac{1}{2}}^{n+1} - u_{i+\frac{1}{2}}^{n}\right).$$

The internal energy is solved

$$m_i\left(\varepsilon_i^{n+1} - \varepsilon_i^n\right) + \Delta t P_i^{*,n+\frac{1}{2}}\left(u_{i+\frac{1}{2}}^{n+\frac{1}{2}} - u_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right) = 0. \tag{A.36}$$

Then the mesh nodes are displaced:

$$X_{i+\frac{1}{2}}^{n+1} = X_{i+\frac{1}{2}}^{n} + \Delta t u_{i+\frac{1}{2}}^{n+\frac{1}{2}}.$$

The density is then computed:

$$\rho_{i+\frac{1}{2}}^{n+1} = m_i\left(V_i^{n+1}\right)^{-1},$$

where

$$V_i^{n+1} = X_{i+\frac{1}{2}}^{n+1} - X_{i-\frac{1}{2}}^{n+1}.$$

High-order extension of this scheme is obtained using a piecewise linear conservative reconstruction of the velocity on each dual cell.

$$\forall X \in [X_i, X_{i+1}], \quad u_{i+\frac{1}{2}}(X) = u_{i+\frac{1}{2}} + \delta u_{i+\frac{1}{2}}\left(X - \widetilde{X}_{i+\frac{1}{2}}\right), \tag{A.37}$$

where $\widetilde{X}_{i+1/2} = (X_i + X_{i+1})/2$ and we note that generally $\widetilde{X}_{i+1/2} \neq X_{i+1/2}$ since the mesh may be non-uniform. The reconstruction is conservative in the sense that

$$\frac{1}{X_{i+1} - X_i}\int_{X_i}^{X_{i+1}} u_{i+\frac{1}{2}}(X)\,\mathrm{d}X = u_{i+\frac{1}{2}}. \tag{A.38}$$

As instance a least squares approach is utilized to compute the slope $\delta u_{i+1/2}$. Monotonicity is ensured by the use of any classical slope limiter. High-order reconstructed velocities

$$u_L = u_{i-\frac{1}{2}}(X_i) = u_{i-\frac{1}{2}} + \delta u_{i-\frac{1}{2}}\left(X_i - \widetilde{X}_{i-\frac{1}{2}}\right)$$

and

$$u_R = u_{i+\frac{1}{2}}(X_i) = u_{i+\frac{1}{2}} + \delta u_{i+\frac{1}{2}}\left(X_i - \widetilde{X}_{i+\frac{1}{2}}\right)$$

are further used in the Riemann solver as left and right states.

References

[1] F. L. Addessio, D. E. Carroll, J. K. Dukowicz, F. H. Harlow, J. N. Johnson, B. A. Kashiwa, M. E. Maltrud and H. M. Ruppel, CAVEAT: a computer code for fluid dynamics problems with large distortion and internal slip, Technical Report LA-10613-MS, Los Alamos National Laboratory, 1986.

Janenko, editor, Difference methods for solutions of problems of mathematical physics, Volume I, pages 116–149, Providence, RI, 1967. American Mathematical Society, Translated from the Proceedings of the Steklov Institute of Mathematics, No. 74 (in Russian) by J. M. Danskin.

[21] R. Loubère and E. J. Caramana, The force/work differencing of exceptional points in the discrete, compatible formulation of Lagrangian hydrodynamics, J. Comput. Phys., 216(1) (2006), 1–18.

[22] R. Loubère and M. J. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods, J. Comput. Phys., 209(1) (2005), 105–138.

[23] G. Luttwak and J. Falcovitz, Slope limiting for vectors: a novel vector limiting algorithm, Conference on Numerical methods for multi-material fluid flows; Pavia University on September 21-25, 2009. Available at `www.eucentre.it/multimat09/media/presentazioni_congresso/luttwak.pdf`.

[24] G. Luttwak and J. Falcovitz, Staggered Mesh Godunov (SMG) schemes for ALE hydrodynamics, Numerical methods for multi-material fluid flows; St Catherine's College Oxford 5th-8th September 2005. Available at `www.icfd.reading.ac.uk/talks/Luttwak.ppt`.

[25] P.-H. Maire, A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry, J. Comput. Phys., 228(18) (2009), 6882–6915.

[26] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured mesh, J. Comput. Phys., 228(7) (2009), 2391–2425.

[27] P.-H. Maire, R. Abgrall, J. Breil and J. Ovadia, A cell-centered Lagrangian scheme for compressible flow problems, SIAM J. Sci. Comput., 29(4) (2007), 1781–1824.

[28] P.-H. Maire and B. Nkonga, Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics, J. Comput. Phys., 228(3) (2009), 799–821.

[29] L. Margolin, M. Shashkov and P. Smolarkiewicz, A discrete operator calculus for finite difference approximations, Comput. Meth. Appl. Mech. Eng., 187 (2000), 365–383.

[30] R. D. Richtmyer, Taylor instability in shock acceleration of compressible fluids, Commun. Pure Appl. Math., 13 (1960), 297–319.

[31] L. I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, New York, 1959.

[32] M. J. Shashkov, Conservative Finite-Difference Methods on General Grids, CRC Press, Boca Raton, FL, 1996.

[33] J. von Neumann and R. D. Richtmyer, A method for the numerical calculations of hydrodynamical shocks, J. Appl. Phys., 21 (1950), 232–238.

[34] M. L. Wilkins, Calculation of elastic plastic flow, in B. Alder, S. Fernbach and M. Rotenberg, editors, Methods in Computational Physics, volume 3, pp. 211–263, Academic Press, New York, 1964.

[35] M. L. Wilkins, Use of artificial viscosity in multidimensional fluid dynamic calculations, J. Comput. Phys., 36(3) (1980), 381–403.

[36] Y. Yang, Q. Zhang and D. H. Sharp, Small amplitude theory of Richtmyer-Meshkov instability, Phys. Fluids, 6(5) (1994), 1856–1873.

[2] T. J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in D. Kroner, M. Ohlberger and C. Rohde, editors, An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws, pages 195–284, Berlin, 1997; Lecture Notes in Computational Science and Engineering, Springer.

[3] T. J. Barth and D. C. Jespersen, The design and application of upwind schemes on unstructured meshes, in AIAA paper 89-0366, 27th Aerospace Sciences Meeting, Reno, Nevada, 1989.

[4] A. L. Bauer, D. E. Burton, E. J. Caramana, R. Loubère, M. J. Shashkov and P. P. Whalen, The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics, J. Comput. Phys., 218(2) (2006), 572–593.

[5] M. Ben-Artzi and J. Falcovitz, Generalized Riemann problems in computational fluid dynamics, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, UK, 2003.

[6] A. Burbeau-Augoula, A node-centered artificial viscosity method for two-dimensional Lagrangian hydrodynamics calculations on a staggered grid, Commun. Comput. Phys., 8(4) (2010), 877–900.

[7] D. E. Burton, Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity, Advances in the Free Lagrange Method, Springer-Verlag, New-York, 1990.

[8] J. C. Campbell and M. J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys., 172(4) (2001), 739–765.

[9] J. C. Campbell and M. J. Shashkov, A compatible Lagrangian hydrodynamics algorithm for unstructured grids, Seluk J. Appl. Math., 4(2) (2003), 53–70.

[10] E. J. Caramana, D. E. Burton, M. J. Shashkov and P. P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys., 146(1) (1998), 227–262.

[11] E. J. Caramana and M. J. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comput. Phys., 142(2) (1998), 521–561.

[12] E. J. Caramana, M. J. Shashkov and P. P. Whalen, Formulations of artificial viscosity for multidimensional shock wave computations, J. Comput. Phys., 144(1) (1998), 70–97.

[13] G. Carré, S. Delpino, B. Després and E. Labourasse, A cell-centered Lagrangian hydrodynamics scheme in arbitrary dimension, J. Comput. Phys., 228(14) (2009), 5160–5183.

[14] R. B. Christensen, Godunov methods on a staggered mesh-an improved artificial viscosity, Technical Report UCRL-JC-105269, Lawrence Livermore National Laboratory, 1990.

[15] B. Després and C. Mazeran, Lagrangian gas dynamics in two dimensions and Lagrangian systems, Arch. Rational Mech. Anal., 178 (2005), 327–372.

[16] J. K. Dukowicz, A general, non-iterative Riemann solver for Godunov's method, J. Comput. Phys., 61 (1985), 119–137.

[17] J. K. Dukowicz and B. Meltz, Vorticity errors in multidimensional Lagrangian codes, J. Comput. Phys., 99 (1992), 115–134.

[18] S. K. Godunov, A. Zabrodine, M. Ivanov, A. Kraiko and G. Prokopov, Résolution numérique des problèmes multidimensionnels de la dynamique des gaz, MIR, 1979.

[19] J. R. Kamm and F. X. Timmes, On efficient generation of numerically robust Sedov solutions, Technical Report LA-UR-07-2849, Los Alamos National Laboratory, 2007.

[20] V. F. Kuropatenko, On difference methods for the equations of hydrodynamics, in N. N.

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com/journal/nmf). DOI: 10.1002/fld.3730

3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver-based artificial viscosity

Raphaël Loubère[1,*,†], Pierre-Henri Maire[2] and Pavel Váchal[3]

[1]*Institut de Mathématiques de Toulouse, CNRS, Université de Toulouse, Toulouse, France*
[2]*CEA-CESTA, BP 2, 33114, Le Barp, France*
[3]*Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Prague, Czech Republic*

SUMMARY

The aim of the present work is the 3D extension of a general formalism to derive a staggered discretization for Lagrangian hydrodynamics on unstructured grids. The classical compatible discretization is used; namely, momentum equation is discretized using the fundamental concept of subcell forces. Specific internal energy equation is obtained using total energy conservation. The subcell force is derived by invoking the Galilean invariance and thermodynamic consistency. A general form of the subcell force is provided so that a cell entropy inequality is satisfied. The subcell force consists of a classical pressure term plus a tensorial viscous contribution proportional to the difference between the node velocity and the cell-centered velocity. This cell-centered velocity is an extra degree of freedom solved with a cell-centered approximate Riemann solver. The second law of thermodynamics is satisfied by construction of the local positive definite subcell tensor involved in the viscous term. A particular expression of this tensor is proposed. A more accurate extension of this discretization both in time and space is also provided using a piecewise linear reconstruction of the velocity field and a predictor-corrector time discretization. Numerical tests are presented in order to assess the efficiency of this approach in 3D. Sanity checks show that the 3D extension of the 2D approach reproduces 1D and 2D results. Finally, 3D problems such as Sedov, Noh, and Saltzman are simulated. Copyright © 2012 John Wiley & Sons, Ltd.

Received 24 February 2012; Revised 18 July 2012; Accepted 20 August 2012

KEY WORDS: Lagrangian hydrodynamics; compressible flow; staggered scheme; 3D; artificial viscosity; Riemann solver; unstructured mesh

1. INTRODUCTION

The aim of the present work is twofold: to develop the 3D version of a general formalism, which was first derived in [1], and to numerically validate this formalism via a 3D staggered Lagrangian hydrodynamics code.

One classical staggered Lagrangian hydrodynamics compatible scheme, which is vastly used for designing 3D code, dates back to [2–4]. The staggered discretization of variables (kinematic variables located at nodes, thermodynamic ones at cell centers) allows the scheme to fulfill naturally the geometric conservation law (GCL) compatibility requirement and, at the same time, provides the construction of a natural discrete divergence operator. The discretizations of momentum and specific internal energy are derived from each other by use of the important concepts of subcell mass, subcell force, and total energy conservation [4]. This compatible hydrodynamics algorithm is thus designed to conserve momentum and total energy exactly in discrete form by using the adjointness property of the discrete gradient and divergence operators. The dissipation of kinetic energy into internal energy through shock waves is ensured by means of artificial viscosity, which can be edge based [5], tensorial [6], or more advanced [7, 8]. This mechanism leads to a dissipation that is compatible with the second law of thermodynamics. The subcell pressure method is also used for the control of hourglass-type motion [9]. Finally, the time integration method is a predictor-corrector technique, which is detailed in [4]. The extension of this compatible Lagrangian hydrodynamics algorithm to unstructured grids, where each zone is a polygon with an arbitrary number of sides, has been presented in [10]. The subcell formalism greatly simplifies the 3D version of the scheme [11].

More recently, cell-centered Lagrangian schemes were developed [12–14]. Their primary variables are the cell-centered density, the velocity, and the total energy. These schemes solve the difficulty of defining a nodal velocity to move the mesh consistently with the GCL. An approximate Riemann problem is constructed at each node, the solution of which provides the nodal velocity. A 3D version of such schemes has been presented for instance in [15–17].

In [1], the authors derived a staggered discretization for 2D Lagrangian hydrodynamics on unstructured grids. This derivation uses the concept of subcell discretization of the staggered Lagrangian schemes and also a cell-centered interpretation of the Riemann solver from the cell-centered Lagrangian schemes. This leads to an artificial viscosity force obtained by invoking the Galilean invariance and thermodynamic consistency. In Godunov methods, the dissipation of kinetic energy into internal energy is provided by a solution of a Riemann problem. The same mechanism is used in the framework of a 3D staggered scheme. The solution of the cell-centered Riemann problem provides an approximation of the cell-centered velocity $\boldsymbol{U}_c$, which is used to define the viscous part of the subcell force. Let us note that, independently, a very similar and interesting work in 2D has been published in [18]. This formulation allows a straightforward more accurate extension in space by constructing linear velocity vector field approximation with frame-invariant limitation, applicable on any mesh structure. A careful and sensitive vector limitation is a key issue to effective exploitation of the improvement gained by frame-invariant higher-order extension. We extend the approach proposed in [1] to obtain a 3D frame invariant vector limitation. As for temporal integration, we achieve second order in time by using a classical predictor-corrector time discretization. The 3D derivation of the scheme is very similar to the 2D one. The resulting viscosity force can be expressed with the help of a 3×3 positive definite matrix M_{cp}, so that the thermodynamic consistency is satisfied automatically by the viscous term $\mathsf{M}_{cp}(\boldsymbol{U}_c - \boldsymbol{U}_p)$. The method in [1] can be viewed as an extension of the work by Christensen [19], who noticed that under certain assumptions, the staggered Lagrangian schemes with artificial viscosity can be written in the same form as Godunov's scheme with Harten, Lax, van Leer (HLL) approximate Riemann solver and stressed the potential synergy of both approaches (e.g., higher-order extension of simple staggered scheme by techniques typically used in the Godunov community, such as Total Variation Diminishing (TVD) limiters). The relationship between staggered Lagrangian and cell-centered Godunov methods from the viewpoint of shock-capturing mechanism has been discussed already in earlier works, for example, by Wilkins [20] or Dukowicz [21].

The paper is organized as follows. First, the governing equations and notation are stated. The compatible discretization is then derived from first principles. The fourth section deals with the definition of the fundamental object named subcell force. This previous derivation shows the necessity of the introduction of a cell-centered velocity as a new degree of freedom. This velocity is then determined using a cell-centered approximate Riemann solver. A more accurate scheme is built using a piecewise linear limited reconstruction of the velocity field. This section also presents the second-order time discretization. In the fifth section, numerical results are provided. First, sanity test cases are shown to assess the validity of this approach when problems with 1D or 2D symmetries are run in 3D. In a second series of tests, genuinely 3D demanding test cases are proposed. The numerical results provided by our scheme are compared with exact solutions or solutions obtained by a classical staggered Lagrangian scheme with edge artificial viscosity possibly supplemented with several useful tools (symmetry corrections [11] or vorticity damping artificial viscosity [22]). Conclusion and perspectives are finally drawn.

*Correspondence to: Raphaël Loubère, Institut de Mathématiques de Toulouse, CNRS, Université de Toulouse, France.
†E-mail: raphael.loubere@math.univ-toulouse.fr

Copyright © 2012 John Wiley & Sons, Ltd.

2. GOVERNING EQUATIONS AND NOTATION

2.1. Governing equations

In a Lagrangian framework, the 3D gas dynamics equations write

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho}\right) - \nabla \cdot \boldsymbol{U} = 0, \tag{1}$$

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U} + \nabla P = \boldsymbol{0}, \tag{2}$$

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}E + \nabla \cdot (P\boldsymbol{U}) = 0, \tag{3}$$

where ρ is the density, $\boldsymbol{U}$ the velocity, E the specific total energy, and $\frac{\mathrm{d}}{\mathrm{d}t}$ the material derivative. Equation (1) expresses the volume conservation, and (2) and (3) are the momentum and total energy conservation laws. Volume conservation equation is referred to as the GCL. A thermodynamic closure (equation of state (EOS)) supplements the previous system by relation $P = P(\rho, \varepsilon)$. The specific internal energy is $\varepsilon = E - \frac{\boldsymbol{U}^2}{2}$. For smooth solutions, energy equation is reformulated into

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}\varepsilon + P\nabla \cdot \boldsymbol{U} = 0, \tag{4}$$

and using (1), one obtains

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}\varepsilon + P\rho \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho}\right) = 0. \tag{5}$$

The Gibbs relation for temperature T and specific entropy S writes $T\mathrm{d}S = \mathrm{d}\varepsilon + P\mathrm{d}\left(\frac{1}{\rho}\right)$, and the second law of thermodynamics, namely, $T\frac{\mathrm{d}S}{\mathrm{d}t} \geqslant 0$, implies that for non-smooth flows, we have

$$\rho \frac{\mathrm{d}}{\mathrm{d}t}\varepsilon + P\rho \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho}\right) \geqslant 0. \tag{6}$$

The previous system (1–3) is replaced in this work by the non-conservative system composed of Equations (1), (2), and (4). The last equations are the trajectory equations

$$\frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t} = \boldsymbol{U}(\boldsymbol{X}(t), t), \qquad \boldsymbol{X}(0) = \boldsymbol{x}, \tag{7}$$

expressing the Lagrangian motion of any point initially located at position $\boldsymbol{x}$.

2.2. Notation and geometrical consideration

An unstructured grid consisting of a collection of non-overlapping cells is considered. Each cell is assigned a unique index c and is denoted as Ω_c. Each point/vertex of the mesh is assigned a unique index p, and $\mathcal{C}(p)$ is the set of cells sharing a particular vertex p. Each cell is subdivided into subcells, each being uniquely defined by a pair of indices c and p and denoted as Ω_{cp}. This subcell is the volume defined by connecting the center of Ω_c to the mid-points of cell edges impinging at point p and the centers of the faces meeting at point p (Figure 2). In other words, $\Omega_{cp} = \Omega_p \cap \Omega_c$. The union of subcells Ω_{cp} sharing a particular point p defines the dual vertex-centered cell Ω_p related to point p as $\Omega_p = \bigcup_{c \in \mathcal{C}(p)} \Omega_{cp}$.

The primary grid is $\bigcup_c \Omega_c$ and the dual grid $\bigcup_p \Omega_p$. The set of faces f of a cell is denoted as $\mathcal{F}(c)$, and the set of faces of a subcell is $\mathcal{F}(cp)$. A face f is uniquely defined by a set of points. Note that a face may not be coplanar if more than three vertexes compose it. Consequently, a face having more than three vertexes is split into triangles. The surface A_f of face f is computed as the sum of the surfaces of the triangles composing it. Moreover, $\boldsymbol{N}_f$, the unit normal of face f, is deduced from the sum of normals of the triangles composing the face.

The method used to calculate volumes and surface vectors is to tetrahedralize the polyhedron of arbitrary order so that the volume of any domain can be formed in a general manner on an unstructured grid (Figure 1). The cell center point is defined as the average of the coordinates of the points that define the cell. A face center point is defined as the average of the coordinates of the points that define it.

For the sake of simplicity, we only describe the framework on a hexahedral mesh knowing that the extension to cells with more faces is immediate. In the case of a hexahedral cell c, the face composed of four points denoted in Figure 1 by p_1, p_2, p_7, p_8 has its center at $\boldsymbol{X}_{f,c} = \frac{1}{4}\sum_{k=1,2,7,8} \boldsymbol{X}_{p_k}$. The cell center is defined by $\boldsymbol{X}_c = \frac{1}{8}\sum_{k=1,\ldots,8} \boldsymbol{X}_{p_k}$. Moreover, the zone is divided into 24 tetrahedra, with two tetrahedra associated with each of the 12 edges. Three tetrahedra are shown in Figure 1. The 24 tetrahedra volumes are further summed to obtain an approximation of the cell volume that is called V_c. Subcell volume is computed by summing the volumes of the six tetrahedra impinging at point p and dividing the sum by factor 2. Let us emphasize that the face vectors satisfy the fundamental geometric identities

$$\sum_{f \in \mathcal{F}(c)} A_{f,c}\boldsymbol{N}_{f,c} = \boldsymbol{0}, \qquad \sum_{f \in \mathcal{F}(cp)} A_{f,cp}\boldsymbol{N}_{f,cp} = \boldsymbol{0}, \tag{8}$$

where $A_{f,c}, \boldsymbol{N}_{f,c}$ are the surface and the unit outward normal of face f in cell c. Similarly, $A_{f,cp}, \boldsymbol{N}_{f,cp}$ are the surface and unit outward normal of face f in subcell Ω_{cp}. Note that the faces of the subcell may not be coplanar either. Consequently, each subcell face is split into two triangles by the diagonal line passing through the cell center for internal faces (arrows on Figure 1) or the point for the external faces. The surface of the subcell face called $A^{\mathrm{int/ext}}_{f,cp}$ is therefore the sum of the surfaces of these two triangles. Similarly, $\boldsymbol{N}^{\mathrm{int/ext}}_{cp}$, the unit normal vector to the subcell face, is the sum of the normals to the triangles divided by its norm. Equation (8) is equivalent to the well-known result that the sum of the outward normals to a closed polyhedral surface is equal to zero.

The sets of internal and external (cell-boundary) faces with respect to cell c are respectively named $\mathcal{F}^{\mathrm{int}}(cp)$, $\mathcal{F}^{\mathrm{ext}}(cp)$, as depicted in Figure 2; and according to the second equation in (8), one has[‡]

$$\sum_{f \in \mathcal{F}^{\mathrm{ext}}(cp)} A^{\mathrm{ext}}_{f,cp}\boldsymbol{N}^{\mathrm{ext}}_{f,cp} = -\sum_{f \in \mathcal{F}^{\mathrm{int}}(cp)} A^{\mathrm{int}}_{f,cp}\boldsymbol{N}^{\mathrm{int}}_{f,cp}. \tag{9}$$

Copyright © 2012 John Wiley & Sons, Ltd.

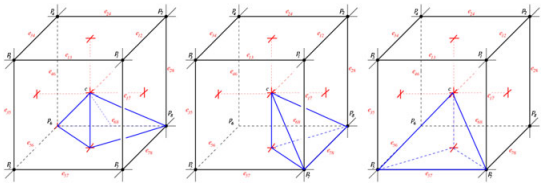


Figure 1. Tetrahedralization of polyhedron of arbitrary order (example of a hexahedron). Note that a face is generally not a simple plane—a hexahedron is split into 24 tetrahedra meeting at the center of the cell.

[‡]Another definition of interior faces is $\partial\Omega_p \cap \Omega_c$ and $\partial\Omega_c \cap \Omega_p$ for exterior faces.

Copyright © 2012 John Wiley & Sons, Ltd.

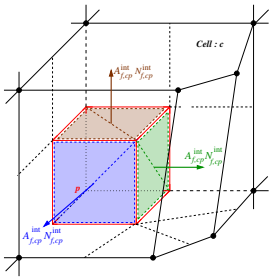


Figure 2. Fragment of an unstructured mesh. Position and velocity are defined at grid points (•), whereas thermodynamic variables are located at cell centers. A cell, Ω_c, is subdivided into hexahedral subcells Ω_{cp}. Points are denoted by subscript p and cells by subscript c. Each internal face of subcell Ω_{cp} (brown, blue, and green) has a superscript 'int'. Similarly, one uses 'ext' for external faces. A unit normal to face f denoted as $\boldsymbol{N}^{\text{int}}_{f,cp}$ is the sum of the normals to the triangles divided by its norm. The surface of face f is $A^{\text{int}}_{f,cp}$.

The previous equation defines the important notion of 3D corner vector as being

$$\boldsymbol{A}_{cp}\boldsymbol{N}_{cp} = \sum_{f\in\mathcal{F}^{\text{ext}}(cp)} A^{\text{ext}}_{f,cp}\boldsymbol{N}^{\text{ext}}_{f,cp}, \tag{10}$$

where $\boldsymbol{N}_{cp}$ is the unit vector in the direction of vector $\sum_{f\in\mathcal{F}^{\text{ext}}(cp)} A^{\text{ext}}_{f,cp}\boldsymbol{N}^{\text{ext}}_{f,cp}$, and A_{cp} is its norm.

We use a staggered placement of variables: Position and velocity are defined at grid points whereas thermodynamic variables are located at cell centers. In a moving framework, the volumes of the primary and dual cells are functions of time t. Following [4], we assume that subcells are Lagrangian entities. Consequently, subcell mass m_{cp} is constant in time. Being given V^0_c as the volume of cell Ω_c at time $t = 0$ and the initial density ρ^0_c, one defines the initial subcell mass as $m_{cp} = \rho^0_c V^0_{cp}$. By summation of Lagrangian subcell masses, one defines cell/point masses as

$$m_c = \sum_{p\in\mathcal{P}(c)} m_{cp}, \qquad m_p = \sum_{c\in\mathcal{C}(p)} m_{cp}, \tag{11}$$

where $\mathcal{P}(c)$ is the set of all points of cell c. These cell/point masses inherit the Lagrangian property of subcell masses, namely, they are constant in time.

3. COMPATIBLE DISCRETIZATION

Staggered Lagrangian schemes are constructed here using the methodology of compatible discretization [2, 4, 23]. The cornerstone of this discretization is the so-called subcell force acting from subcell Ω_{cp} onto point p. In this compatible approach, the discretization of the internal energy equation is deduced from total energy conservation. Closely following [1], we derive an abstract form of the subcell force in 3D so that an entropy inequality is satisfied. The subcell force consists of a classical pressure term plus a tensorial viscous contribution, which is proportional to the difference between the vertex-centered and cell-centered velocities. The cell-centered velocity is a

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

new degree of freedom that will be solved using the principle of the Galilean invariance. To satisfy the second law of thermodynamics, the local subcell matrix involved in the viscous part of the subcell force must be positive definite. This matrix is the fundamental object that entirely defines the artificial viscosity of the scheme required to stabilize the scheme.

3.1. Geometric conservation law

We build a discretization of the volume equation (1), which is compatible with the GCL. In other words, we are using a discrete divergence operator for the volume equation by requiring consistency of the divergence of the velocity field with the time rate of change of volume of the cell (refer to [24]). By noticing that $m_c = \rho_c V_c$ is constant in time but $\rho_c = \rho_c(t)$ and $V_c = V_c(t)$, we deduce

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) = \frac{\mathrm{d}}{\mathrm{d}t}V_c.$$

Moreover, because the cell volume can be expressed as a function of the position vectors of its vertexes (see [1]), we can write the time evolution of the cell volume as

$$\frac{\mathrm{d}}{\mathrm{d}t}V_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{A}_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p = 0, \tag{12}$$

where $\boldsymbol{A}_{cp}\boldsymbol{N}_{cp}$ is the corner vector defined in (10), and $\boldsymbol{U}_p$ is the node velocity.

A compatible definition of the discrete divergence operator over cell c is given by

$$(\nabla\cdot\boldsymbol{U})_c = \frac{1}{V_c}\sum_{p\in\mathcal{P}(c)} \boldsymbol{A}_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p. \tag{13}$$

This kind of formalism has already been used in staggered and cell-centered (free Lagrange) discretizations [24, 25]. Note that (12) is compatible with the discrete version of the trajectory equation (7), namely,

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}_p = \boldsymbol{U}_p, \quad \boldsymbol{X}_p(0) = \boldsymbol{x}_p.$$

Finally, a compatible discretization of the volume equation (1) writes

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \sum_{p\in\mathcal{P}(c)} \boldsymbol{A}_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p = 0. \tag{14}$$

3.2. Momentum equation

The semi-discrete momentum equation over the dual cell Ω_p is postulated to be

$$m_p \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_p + \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp} = \boldsymbol{0}. \tag{15}$$

Here $\boldsymbol{F}_{cp}$ is the subcell force from cell c that acts on node p, which is defined by

$$\boldsymbol{F}_{cp} = \int_{\partial\Omega_p(t)\cap\Omega_c(t)} P\boldsymbol{N}\,\mathrm{d}l. \tag{16}$$

Momentum equation (15) can be seen as the Newton law applied to a particle of mass m_p moving with velocity $\boldsymbol{U}_p$.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

3.3. Specific internal energy equation

The compatible staggered Lagrangian machinery relies on the conservation of total energy to obtain a semi-discrete internal energy equation following the approach described in [4]. The total energy is then defined as $E(t) = \mathcal{K}(t) + \mathcal{E}(t)$, where total kinetic energy and total internal energy are given by

$$\mathcal{K}(t) = \sum_p \frac{1}{2}m_p\boldsymbol{U}_p^2(t), \qquad \mathcal{E}(t) = \sum_c m_c\varepsilon_c(t),$$

and ε_c is the cell-averaged specific internal energy. The conservation of total energy neglecting boundary conditions writes $\frac{\mathrm{d}}{\mathrm{d}t}E = \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{K} + \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E} = 0$. The substitution of kinetic and internal energies yields

$$\frac{\mathrm{d}}{\mathrm{d}t}E = \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{K} + \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E} = \sum_c m_c\frac{\mathrm{d}}{\mathrm{d}t}\varepsilon_c + \sum_p m_p\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_p\cdot\boldsymbol{U}_p.$$

Using the semi-discrete momentum equation (15) yields

$$\sum_c m_c\frac{\mathrm{d}}{\mathrm{d}t}\varepsilon_c - \sum_p\sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p = 0,$$

which rewrites as

$$\sum_c \left(m_c\frac{\mathrm{d}}{\mathrm{d}t}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p\right) = 0. \tag{17}$$

A sufficient condition is obtained by requiring the previous equation to hold in each cell c

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p = 0. \tag{18}$$

To summarize, the semi-discrete equations for the primary variables $\left(\frac{1}{\rho_c}, \boldsymbol{U}_p, \varepsilon_c\right)$ are

$$\begin{aligned}
&m_c\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \sum_{p\in\mathcal{P}(c)} \boldsymbol{A}_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p = 0,\\
&m_p\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_p + \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp} = \boldsymbol{0},\\
&m_c\frac{\mathrm{d}}{\mathrm{d}t}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}\cdot\boldsymbol{U}_p = 0.
\end{aligned}$$

The mesh motion given by the trajectory equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}_p = \boldsymbol{U}_p(\boldsymbol{X}_p(t), t), \qquad \boldsymbol{X}_p(0) = \boldsymbol{x}_p,$$

is compatible with the GCL. The thermodynamic closure is provided by the EOS, which writes $P_c = P(\rho_c, \varepsilon_c)$. This subcell-based compatible semi-discretization ensures total energy conservation regardless of the subcell force form.

4. SUBCELL FORCE

Here we follow the definition of the subcell force from [1], where the Galilean invariance and thermodynamic consistency are invoked. The Galilean invariance is a principle of relativity,

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

which states that the fundamental laws of physics are the same in all inertial frames. To fulfill the Galilean invariance, the specific internal energy equation (18) must remain unchanged under a uniform translation denoted as $\boldsymbol{C}$. Equation (18) becomes

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\varepsilon_c - \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}\cdot(\boldsymbol{U}_p + \boldsymbol{C}) = 0.$$

Substitution of (18) into this last equation leads to

$$\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}\cdot\boldsymbol{C} = 0,$$

which must hold for all vectors $\boldsymbol{C}$. Therefore, specific internal energy equation remains invariant under uniform translation if and only if

$$\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp} = \boldsymbol{0}. \tag{19}$$

This condition also implies total momentum conservation without taking into account boundary conditions[§].

In [1], the authors derived a possible form of the subcell force as a sufficient condition to fulfill the second principle of thermodynamics. We refer the reader to [1] for the derivation and only recall the final subcell form

$$\boldsymbol{F}_{cp} = -A_{cp}P_c\boldsymbol{N}_{cp} + \mathsf{M}_{cp}(\boldsymbol{U}_p - \boldsymbol{U}_c). \tag{20}$$

In 3D, M_{cp} is a subcell-based 3×3 positive semi-definite matrix. Such form provides the following entropy inequality satisfied by the semi-discrete scheme [1]:

$$m_cT_c\frac{\mathrm{d}}{\mathrm{d}t}S_c = \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{cp}\left(\boldsymbol{U}_p - \boldsymbol{U}_c\right)\cdot\left(\boldsymbol{U}_p - \boldsymbol{U}_c\right) \geqslant 0, \tag{21}$$

where S_c and T_c are the specific entropy and temperature of cell c. Indeed, using the Gibbs formula, one obtains

$$m_cT_c\frac{\mathrm{d}}{\mathrm{d}t}S_c = m_c\left[\frac{\mathrm{d}}{\mathrm{d}t}\varepsilon_c + P_c\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right)\right]. \tag{22}$$

Substituting into (22) the specific internal energy equation (18) and the volume equation (14) yields

$$\begin{aligned}
m_cT_c\frac{\mathrm{d}}{\mathrm{d}t}S_c &= \sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp}\cdot(\boldsymbol{U}_p - \boldsymbol{U}_c) + P_c\left(\sum_{p\in\mathcal{P}(c)} \boldsymbol{A}_{cp}\boldsymbol{N}_{cp}\cdot\boldsymbol{U}_p\right)\\
&= \sum_{p\in\mathcal{P}(c)} (\boldsymbol{F}_{cp} + \boldsymbol{A}_{cp}P_c\boldsymbol{N}_{cp})\cdot(\boldsymbol{U}_p - \boldsymbol{U}_c)\\
&= \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{cp}\left(\boldsymbol{U}_p - \boldsymbol{U}_c\right)\cdot\left(\boldsymbol{U}_p - \boldsymbol{U}_c\right).
\end{aligned}$$

Here we have used the geometric identity $\sum_{p\in\mathcal{P}(c)} \boldsymbol{A}_{cp}\boldsymbol{N}_{cp} = \boldsymbol{0}$ to obtain the second line, and we have substituted the subcell force form (20) to obtain the third and final equation. As M_{cp} is a positive semi-definite matrix, (21) is fulfilled.

[§]Summing (15) over all points p yields $\sum_p\left(m_p\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_p + \sum_{c\in\mathcal{C}(p)} \boldsymbol{F}_{cp}\right) = \boldsymbol{0}$. Interchanging the sums yields $\sum_p m_p\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_p + \sum_c\sum_{p\in\mathcal{P}(c)} \boldsymbol{F}_{cp} = \boldsymbol{0}$, which, according to (19), leads to $\sum_p m_p\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_p = \boldsymbol{0}$, which means that the total momentum is conserved.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

Entropy production (21) within cell c is directly governed by the subcell matrix M_{cp} and the velocity jump between the nodal and the cell-centered velocity, which still remains to be determined. In [1], the authors provided an abstract formulation, which, by analogy with the node-centered approximate Riemann solver introduced in the context of cell-centered Lagrangian discretization [13], provides a cell-centered approximate Riemann solver. This solver allows to determine one particular form of the subcell matrix M_{cp}. To this end, we first compute the unknown cell center velocity by substituting the subcell force expression (20) into the Galilean invariance condition (19), leading to the following system satisfied by the cell-centered velocity $\boldsymbol{U}_c$:

$$\mathsf{M}_c \boldsymbol{U}_c = \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{cp}\boldsymbol{U}_p, \tag{23}$$

where $\mathsf{M}_c = \sum_{p\in\mathcal{P}(c)} \mathsf{M}_{cp}$ is a symmetric positive definite matrix. Once the definition of the subcell matrix M_{cp} is known, one can solve the previous system to obtain a unique expression of the cell-centered velocity.

There is no unique definition of subcell matrix neither in 2D nor in 3D. Consequently, we rely on the proposition made in [1], which we further extend in 3D. That is to say,

$$\mathsf{M}_{cp} = \sum_{f\in\mathcal{F}^{int}(cp)} Z_{f,cp}\, A_{f,cp}\, (\boldsymbol{N}_{f,cp}\otimes\boldsymbol{N}_{f,cp}) \tag{24}$$

is a 3×3 symmetric positive definite matrix where each matrix $(\boldsymbol{N}_{f,cp}\otimes\boldsymbol{N}_{f,cp})$ is symmetric positive definite, $A_{f,cp}$ is the surface of internal face f of subcell Ω_{cp}, and $\boldsymbol{N}_{f,cp}$ is its unit outward-pointing normal. The last ingredient of formula (24) is the swept mass flux for a given face f of subcell Ω_{cp} that we define following Dukowicz [21] as

$$Z_{f,cp} = \rho_c \left[\sigma_c + c_Q\Gamma_c \,|\, (\boldsymbol{U}_c - \boldsymbol{U}_p)\cdot\boldsymbol{N}_{f,cp}\,|\right]. \tag{25}$$

Here σ_c is the isentropic sound speed, c_Q a user-defined parameter (set to 1 in our simulations), and Γ_c a material-dependent coefficient, which, for a γ law gas, is defined by

$$\Gamma_c = \begin{cases} \dfrac{\gamma+1}{2} & \text{if } (\nabla\cdot\boldsymbol{U})_{cp} < 0, \\ 0 & \text{if } (\nabla\cdot\boldsymbol{U})_{cp} \geq 0. \end{cases} \tag{26}$$

Here $(\nabla\cdot\boldsymbol{U})_{cp} = -\frac{1}{V_{cp}} A_{cp}\boldsymbol{N}_{cp}\cdot(\boldsymbol{U}_c - \boldsymbol{U}_p)$ is the subcell related velocity divergence. In case of a rarefaction wave, we recover the acoustic approximation, whereas in case of a shock wave, we obtain the well-known two-shock approximation.

We note that M_c in (23) is symmetric positive definite, which ensures its invertibility. We remark that this system is non-linear because of the dependency of the swept mass flux on the cell-centered velocity. This non-linear system can be solved using an iterative procedure such as fixed point or Newton algorithms. In practice, few iterations are needed to reach convergence. Once the cell-centered velocity is known, the subcell force is deduced from Equation (20). The present cell-centered approximate Riemann solver can be viewed as a 3D extension of the work initiated by Christensen in 1D framework [19] and extended to 2D in [1].

5. IMPROVEMENTS

Anti-hourglass subpressure force. Subcell pressure forces have been designed in [9] in classical staggered discretization to counteract some parasitic hourglass grid motion that may otherwise appear. Following [9] or in the same fashion as in [1], subpressure force formalism is extended to 3D without further modification. If the technique from [1] is considered, then the cell pressure P_c must be replaced by the subcell pressure P_{cp} obtained from the subcell density ρ_{cp} and the

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

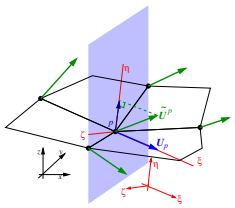


Figure 3. Definition of the orthonormal basis $(\boldsymbol{\xi}^p, \boldsymbol{\eta}^p, \boldsymbol{\zeta}^p)$ according to direction of flow at and around node p. The node velocity is chosen as the main direction, and the first basis vector is $\boldsymbol{\xi}^p = \frac{\boldsymbol{U}_p}{\|\boldsymbol{U}_p\|}$ (blue vector).

cell-based specific internal energy ε_c via the EOS. In other words, the subcell force for the first-order scheme is

$$\boldsymbol{F}_{cp} = -A_{cp}P_{cp}\boldsymbol{N}_{cp} + \mathsf{M}_{cp}\left(\boldsymbol{U}_p - \boldsymbol{U}_c\right). \tag{27}$$

We skip the whole description and refer the reader to [9] and [1].

Reconstruction and limitation of 3D velocity field. The previous solver utilizes piecewise constant nodal velocities defined over the subcells. In order to obtain a more accurate approximation in a monotone upstream-centered schemes for conservation laws sense, we construct a piecewise linear representation of the nodal velocity field. The extrapolated velocity at cell centers is further used in the solver.

We introduce a piecewise linear reconstruction of the velocity field over the dual grid by setting

$$\boldsymbol{U}_p(\boldsymbol{X}) = \boldsymbol{U}_p + \nabla\boldsymbol{U}_p\cdot(\boldsymbol{X} - \boldsymbol{X}_p), \tag{28}$$

where $\nabla\boldsymbol{U}_p$ is the velocity tensor gradient constant over the dual cell Ω_p obtained with a least-squares approach. This approach is valid for any type of unstructured grid and preserves linear velocity field. Monotonicity is achieved thanks to a modification of the classical Barth–Jespersen slope limiter [26] later popularized in [27][¶] as described in [1] and further extended in 3D.

Limiting for vectors is usually applied separately to each component. However, such a procedure is frame dependent. Consequently, component limiters may not preserve symmetry because a rotation of the coordinate axes may produce different results. This is critical in Lagrangian hydrodynamics especially in 3D because we are dealing with moving mesh discretizations, which are particularly sensitive to symmetry discrepancy. The limiter in [1] is a frame-invariant tensorial limitation for vector field. Because its original form did not rely on dimensionality argument, its 3D extension is immediate. More precisely in 2D, this limiter determines two subcell-based orthonormal directions, which are the ones given by the node velocity and its associated perpendicular direction. In 3D, we define the local basis $(\boldsymbol{\xi}^p, \boldsymbol{\eta}^p, \boldsymbol{\zeta}^p)$ according to the direction of the flow at and around node p. There is no unique way of defining these vectors. For example, we define the orthonormal basis as shown in Figure 3: The node velocity is chosen as the main direction, and the first basis vector is therefore $\boldsymbol{\xi}^p = \frac{\boldsymbol{U}_p}{\|\boldsymbol{U}_p\|}$. One possible choice is to define $\boldsymbol{\eta}^p$ as the second most representative direction of the flow, which is the average flow in the surrounding ($\widetilde{\boldsymbol{U}}_p$ in Figure 3) further projected onto the plane $\mathcal{P}^p$ perpendicular to $\boldsymbol{\xi}^p$. Now $\boldsymbol{\eta}^p$ is perpendicular to $\boldsymbol{\xi}^p$. The basis vectors

[¶]We use the 'Barth–Jespersen' name, which is usual in the community, although such limiter was first introduced in [26] by Desideri and Dervieux.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

$\boldsymbol{\eta}^p$ and $\boldsymbol{\zeta}^p$ form the orthonormal basis of plane $\mathcal{P}^p$. In the case of a uniform flow, we define $\boldsymbol{\eta}^p$ and $\boldsymbol{\zeta}^p$ as any two mutually perpendicular vectors in plane $\mathcal{P}^p$.

The transformation matrix from the canonical basis to the local basis $(\boldsymbol{\xi}^p, \boldsymbol{\eta}^p, \boldsymbol{\zeta}^p)$ is denoted as A_p. Consequently, the transformation of the nodal velocity $\boldsymbol{U}_p$ to the local coordinates is given by

$$\boldsymbol{W}_p = \begin{pmatrix} W_p^\xi \\ W_p^\eta \\ W_p^\zeta \end{pmatrix} = \mathsf{A}_p\boldsymbol{U}_p = \begin{pmatrix} \boldsymbol{\xi}_p^\xi\cdot\boldsymbol{U}_p \\ \boldsymbol{\xi}_p^\eta\cdot\boldsymbol{U}_p \\ \boldsymbol{\xi}_p^\zeta\cdot\boldsymbol{U}_p \end{pmatrix}. \tag{29}$$

Then we find the minimum and maximum values from projections of neighboring nodes' velocities into these new directions:

$$W_p^{\xi,\max} = \max_{k\in\mathcal{N}(p)}\left(\boldsymbol{\xi}_p^\xi\cdot\boldsymbol{U}_k\right), \quad W_p^{\eta,\max} = \max_{k\in\mathcal{N}(p)}\left(\boldsymbol{\xi}_p^\eta\cdot\boldsymbol{U}_k\right), \quad W_p^{\xi,\max} = \max_{k\in\mathcal{N}(p)}\left(\boldsymbol{\xi}_p^\zeta\cdot\boldsymbol{U}_k\right), \tag{30}$$

$$W_p^{\xi,\min} = \min_{k\in\mathcal{N}(p)}\left(\boldsymbol{\xi}_p^\xi\cdot\boldsymbol{U}_k\right), \quad W_p^{\eta,\min} = \min_{k\in\mathcal{N}(p)}\left(\boldsymbol{\xi}_p^\eta\cdot\boldsymbol{U}_k\right), \quad W_p^{\xi,\min} = \min_{k\in\mathcal{N}(p)}\left(\boldsymbol{\xi}_p^\zeta\cdot\boldsymbol{U}_k\right), \tag{31}$$

where $\mathcal{N}(p)$ is the set of neighbor points of current point p. Now consider cell $c\in\mathcal{C}(p)$ centered at $\boldsymbol{X}_c$. Using the unlimited piecewise linear representation of the velocity field, the extrapolated values of the velocity at point $\boldsymbol{X}_c$ are given by

$$\boldsymbol{U}_{p,c} \equiv \boldsymbol{U}_p(\boldsymbol{X}_c) = \boldsymbol{U}_p + \nabla\boldsymbol{U}_p\cdot(\boldsymbol{X}_c - \boldsymbol{X}_p), \tag{32}$$

and their transformation into the local basis $(\boldsymbol{\xi}^p, \boldsymbol{\eta}^p, \boldsymbol{\zeta}^p)$ is

$$\boldsymbol{W}_{p,c} = \begin{pmatrix} W_{p,c}^\xi \\ W_{p,c}^\eta \\ W_{p,c}^\zeta \end{pmatrix} = \mathsf{A}_p\boldsymbol{U}_{p,c}. \tag{33}$$

From these values, we define (κ stands for ξ, η, or ζ)

$$\phi_{p,c}^\kappa = \begin{cases} \mathcal{L}\left(\dfrac{W_p^{\kappa,\max} - W_p^\kappa}{W_{p,c}^\kappa - W_p^\kappa}\right) & \text{if } \left(W_{p,c}^\kappa - W_p^\kappa\right) > 0 \\ \mathcal{L}\left(\dfrac{W_p^{\kappa,\min} - W_p^\kappa}{W_{p,c}^\kappa - W_p^\kappa}\right) & \text{if } \left(W_{p,c}^\kappa - W_p^\kappa\right) < 0 \\ 1 & \text{if } \left(W_{p,c}^\kappa - W_p^\kappa\right) = 0 \end{cases},$$

where $\mathcal{L}(\alpha)$ is a limiting functional such as $\mathcal{L}(\alpha) = \min(\alpha, 1)$ or $\mathcal{L}(\alpha) = \frac{\alpha^2+2\alpha}{\alpha^2+\alpha+2}$. The slope limiters for node p are finally defined by $\phi_p^\xi, \phi_p^\eta, \phi_p^\zeta$ as

$$\phi_p^\xi = \min_{c\in\mathcal{C}(p)}\phi_{p,c}^\xi, \qquad \phi_p^\eta = \min_{c\in\mathcal{C}(p)}\phi_{p,c}^\eta, \qquad \phi_p^\zeta = \min_{c\in\mathcal{C}(p)}\phi_{p,c}^\zeta.$$

This triplet of limiters is transformed back into the Cartesian coordinates with

$$\Phi_p = \mathsf{A}_p^{-1}\begin{pmatrix} \phi_p^\xi & 0 & 0 \\ 0 & \phi_p^\eta & 0 \\ 0 & 0 & \phi_p^\zeta \end{pmatrix}\mathsf{A}_p. \tag{34}$$

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

The limited tensor gradient is finally given by formula

$$\nabla\boldsymbol{U}_p^{\lim} = \Phi_p\nabla\boldsymbol{U}_p,$$

and thus the limited velocity field reconstruction in the vicinity of node p is

$$\boldsymbol{U}_p(\boldsymbol{X}) = \boldsymbol{U}_p + \nabla\boldsymbol{U}_p^{\lim}(\boldsymbol{X} - \boldsymbol{X}_p). \tag{35}$$

At this point, let us remark that (34) is valid for any choice of non-coplanar vector set $(\boldsymbol{\xi}^p, \boldsymbol{\eta}^p, \boldsymbol{\zeta}^p)$. In case we choose orthonormal basis vectors, such as those described earlier and shown in Figure 3, the transformation matrix A_p is orthogonal, meaning that $\mathsf{A}_p^{-1} = \mathsf{A}_p^T$, and thus, we do not need to solve any linear system to limit the tensor gradient of velocity.

Finally, the extension of the cell-centered approximate Riemann solver consists in replacing the point velocity by its extrapolated value at cell center by using the piecewise linear monotonic reconstruction as

$$\mathsf{M}_c\boldsymbol{U}_c = \sum_{p\in\mathcal{N}(c)} \mathsf{M}_{cp}\boldsymbol{U}_p(\boldsymbol{X}_c),$$

where the swept mass fluxes are

$$Z_{f,cp} = \rho_c\left[\sigma_c + c_Q\Gamma_c \,|\, (\boldsymbol{U}_c - \boldsymbol{U}_p(\boldsymbol{X}_c))\cdot\boldsymbol{N}_{f,cp}\,|\right].$$

The subcell force is modified accordingly to

$$\boldsymbol{F}_{cp} = -A_{cp}P_c\boldsymbol{N}_{cp} + \mathsf{M}_{cp}\left(\boldsymbol{U}_p(\boldsymbol{X}_c) - \boldsymbol{U}_c\right). \tag{36}$$

Time discretization. The time discretization is performed with a predictor-corrector scheme to gain second-order accuracy in time. Being given the geometric quantities and physical variables at time t^n, one predicts the pressures that are later used in the corrector step to update physical and geometric variables. In the corrector step, internal energy is discretized using the time-centered nodal velocity $\boldsymbol{U}_p^{n+1/2}$. This choice is required to ensure total energy conservation up to machine precision. We refer the reader to [1] for more details about the algorithm.

6. NUMERICAL RESULTS

The purpose of this section is to provide numerical evidence to assess the validity of the proposed approach in 3D. We will compare the results obtained by the original scheme (referred to as 'original') and/or the scheme using piecewise linear velocity field reconstruction (referred to as 'with velocity reconstruction') against analytical solutions when available. In a first series of sanity checks, we will prove that the implementation of the 3D scheme reproduces 1D or 2D known results; these sanity tests are the 1D Sod problem and the 2D Sedov problem run on 3D hexahedral grids. In a second series of tests, we will run the 3D Sedov, 3D Noh, and 3D Saltzman problems, the exact solutions of which are known.

6.1. Sanity checks

1D Sod problem We run the 1D Sod problem with the 3D code. The Sod problem is a 1D Riemann shock tube whose solution consists of a left-moving rarefaction fan, a right-moving contact discontinuity, and a right-moving shock wave. The domain is filled with an ideal gas at rest with $\gamma = 1.4$. The discontinuity is located at $x = 0.5$ at $t = 0$. The density/pressure values on the left side of the discontinuity are 1.0/1.0. Those on the right side are 0.125/0.1. The final time is $t_{\text{final}} = 0.2$. In our numerical experiments, the computational domain is $\Omega = [0;1]\times[0;0.03]\times[0;0.03]$ paved with a regular hexahedral mesh. We are enforcing zero normal velocity as boundary conditions for $N_x = 100$ and $N_y = N_z = 3$.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

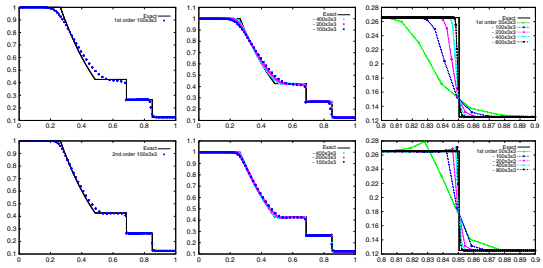


Figure 4. Sod problem at $t_{\text{final}} = 0.2$ for N_x cells in x direction, and 3 in y and z directions. Cell-centered density as a function of x for all cells versus exact solution (black thick line). Top line: original scheme results; bottom line: scheme with velocity reconstruction results. Left panels: $N_x = 100$ results; middle panels: comparison of $N_x = 100, 200, 400$ results; right panels: enlarged image of the shock wave for $N_x = 50, 100, 200, 400$, and 800.

Figure 4 presents the original scheme results (top) and the scheme with velocity reconstruction results. We display the cell density for all cells *vs* the exact solution with solid line (left panels) and a 3D view on the mesh cells colored by density (right panels). The symmetry of the scheme is perfect, and the quality of the scheme with velocity reconstruction is close to the cell-centered Lagrangian scheme [13]. The same comments as the ones from [1] do apply here.

2D Sedov blast wave problem The second sanity check is the Sedov blast wave problem (see [28]), which describes the evolution of a blast wave in a point-symmetric explosion in 2D. We consider the cylindrical symmetric Sedov problem in Cartesian geometry. The total energy is concentrated at the origin with magnitude $E_{\text{total}} = 0.244816$. An ideal gas with $\gamma = 1.4$ initially at rest is considered. The initial density is equal to 1.0. At final time $t_{\text{final}} = 1.0$, the exact solution is a cylindrically symmetric diverging shock whose front radius is $r = \sqrt{x^2 + z^2} = 1$. It has a density peak $\rho = 6.0$. In the case of a hexahedral mesh in (x, y, z) coordinate system on domain $\Omega = [0.0, 1.2] \times [0.0, 0.12] \times [0.0, 1.2]$, the total energy E_{total} is concentrated in cells such that $x_c = z_c = 0$ (see the red cells on the top-left panel of Figure 5).

The mesh is made of $30\times3\times30$ hexahedra in x, y, z directions, respectively. The results displayed in Figure 5 show the density color map and the mesh at final time on a 3D view (top) obtained by the second-order scheme. The bottom-left panel shows the mesh and the density on the 2D x–z plane at $y = 0$, and the bottom-right panel displays the 1D cut at $z = x$ in green and 1D cuts at $z = 3x$ and $z = x/3$ from the 2D x–z plane at $y = 0$. The results proposed by the 3D version of the method are very similar to the 2D results obtained in [1, Figure 4(a, b)]. This proves that the 3D code can reproduce 2D results without alteration.

6.2. 3D Sedov problem

In the 3D version of the Sedov problem, the total energy of the explosion is concentrated at the origin and has magnitude $E_{\text{total}} = 0.851072$ for $\gamma = 1.4$ similarly to [29]. At final time $t_{\text{final}} = 1.0$, the exact solution is a spherically symmetric diverging shock whose front is at radius $r = \sqrt{x^2 + y^2 + z^2} = 1$ and has a density peak $\rho = 6.0$.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

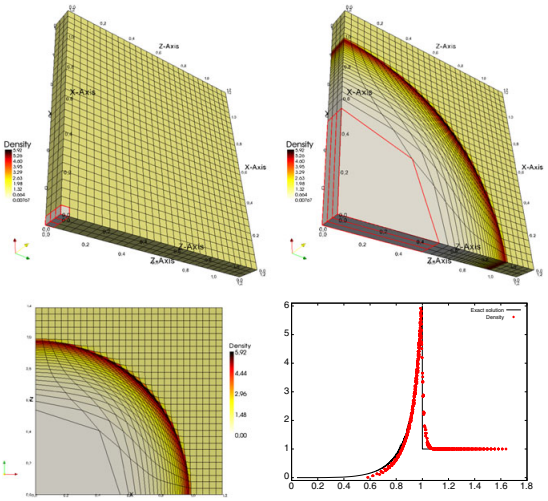


Figure 5. Sedov problem at $t_{\text{final}} = 1$. Scheme with velocity reconstruction. Top: density map and mesh on a $30 \times 3 \times 30$ Cartesian grid at initial and final time (the red highlighted cells are the ones with initial high energy). Bottom-left: 2D x–z plane at $y = 0$, density map and mesh. Bottom-right: density as a function of cell radius (all cells are displayed).

In the case of a hexahedral mesh in (x, y, z) coordinate system on domain $\Omega = [-1.2, 1.2]^3$, the total energy E_{total} is concentrated in cells c in contact with the origin (that is, containing the vertex $\boldsymbol{O} = (0, 0, 0)$). The specific internal energy of these cells is defined as $\varepsilon_c = E_{\text{total}}/(8\ V_c)$, where V_c is the volume of one cell (initially the same for all cells). Therefore, the initial pressure for this cell is $p = (\gamma - 1)\rho\varepsilon = 0.4\ E_{\text{total}}/(8\ V_c)$.

Figure 6 displays the color map of density and mesh at final time $t = 1.0$ when a $20 \times 20 \times 20$ hexahedral mesh is used on one-eighth of the cube. The results of the scheme with velocity reconstruction have been mirrored two times in order to see the cylindrical symmetry of the shock wave in this figure. The 3D Sedov problem on the full cube has also been simulated, and the symmetry is also perfectly reproduced (not displayed in this paper). In the next set of results (Figure 7), we ran the same problem for the original scheme, the scheme with velocity reconstruction, and a classical staggered compatible Lagrangian scheme [11] with anti-hourglass subpressure forces [9] (merit factor 1) with edge artificial viscosity [5]. The mesh is a $20 \times 20 \times 20$ hexahedral mesh. The density as a function of cell radius is plotted for these three runs versus the exact solution. We can observe the lack of cylindrical symmetry on the results for the original method with edge artificial viscosity, and this can be seen especially along the axes, whereas the two schemes proposed in this paper almost exactly reproduce the cylindrical symmetry of the shock wave. As expected, the

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

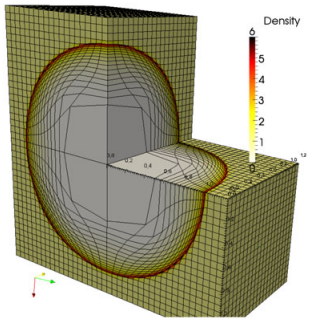


Figure 6. 3D Sedov problem. Scheme with velocity reconstruction. Density map and mesh on a $20\times20\times20$ hexahedral mesh at final time $t = 1.0$.

scheme with velocity reconstruction produces more accurate results. The panels of Figure 8 present the density as a function of radius for all cells in the domain at final time versus the exact solution for $10 \times 10 \times 10$ (column (a)), $20 \times 20 \times 20$ (column (b)), and $40 \times 40 \times 40$ (column (c)) meshes. The top line presents the original scheme whereas the bottom line presents the scheme with velocity reconstruction. These plots show that the 3D code results converge to the exact solution when the grid is refined.

In Figure 9, we compare the results without anti-hourglass force obtained by the classical compatible staggered scheme with edge viscosity and the schemes without or with velocity reconstruction (see Section 5). The mesh is made of $20 \times 20 \times 20$ hexahedra. We present the 2D view of the y–z plane of the density and mesh at final time $t = 1.0$ for the classical edge viscosity at the top-left corner, original scheme at the top-right corner, and the scheme with velocity reconstruction at the bottom-left corner. For comparison purposes, we also present the scheme with velocity reconstruction supplemented by the anti-hourglass force (merit factor 1) on the bottom-right corner. For all calculations without anti-hourglass force, the mesh is tangled. For the schemes presented in this paper, the central cell and the first layer of neighbor cells are tangled, whereas inappropriate jets are observed along axes when classical edge viscosity is used. In fact, these jets were already visible in Figure 7 but with a little amplitude as the anti-hourglass force tends to damp them. Because our implementation only relies on cell volume, all these calculations are completed without code failure, neither negative cell energy nor negative density is encountered. The density as a function of cell radius is also compared with the exact solution in Figure 7(a–c). Notice that this first cell tangling effect is not cured by mesh refinement, and some kind of anti-hourglass force or stiff artificial viscous force must be employed. In our implementation, the smallest merit factor that ensures that the mesh remains valid is 0.06.

6.3. 3D Saltzman problem

This is the extended version of the Saltzman piston as previously defined in [11] and rephrased in [22]. Let us note that the $100 \times 10 \times 10$ mesh is completely 3D in its setup because the Saltzman skewing of the grid is made to change parity uniformly in the third dimension (Figure 10). The equations to obtain the skewed grid are recalled in [17]. In 2D or 3D, this problem is used to test

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

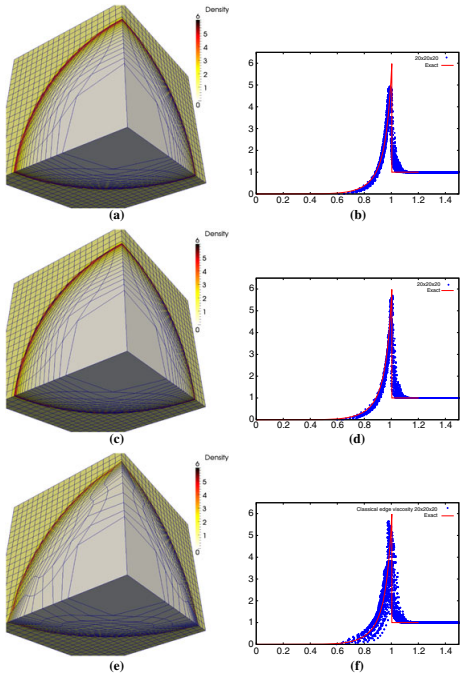


Figure 7. 3D Sedov problem. Density as a function of radius for all cells in the domain. Left: 3D view; right: density as a function of cell radius. Final time $t = 1.0$ on a hexahedral $20 \times 20 \times 20$ mesh. (a and b) Original scheme with anti-hourglass forces. (c and d) Scheme with velocity reconstruction with anti-hourglass forces. (e and f) Classical compatible staggered scheme with edge viscosity with anti-hourglass forces.

the robustness of Lagrangian schemes. For this problem, we use the polytropic index $\gamma = 5/3$. The initial state is $\rho_0 = 1$, $\varepsilon_0 = 10^{-6}$. The plane originally at $x = 0$ is driven by a piston with a unit normal velocity. We set reflective boundary conditions elsewhere. The exact solution is a y–z planar shock wave moving at speed $4/3$ in the x direction. The final time is $t = 0.6$. In Figure 10, we present the results obtained by the original scheme and the scheme with velocity reconstruction. More precisely, the density map and mesh are displayed in panels (a) and (b). Moreover, the density

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

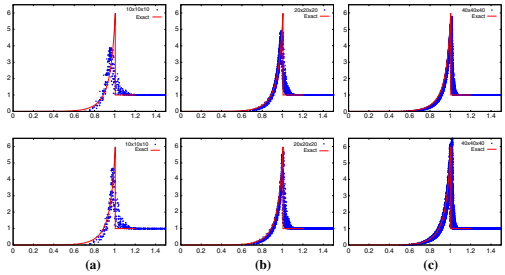


Figure 8. 3D Sedov problem. Density as a function of radius for all cells in the domain at final time $t = 1.0$. Top line: original scheme. Bottom line: scheme with velocity reconstruction. Column (a): $10\times10\times10$ mesh. Column (b): $20\times20\times20$ mesh. Column (c): $40\times40\times40$ mesh. We show the exact solution by a red thick line.

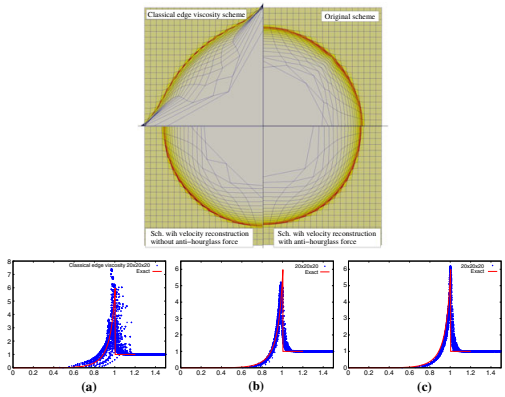


Figure 9. 3D Sedov problem. 2D view in the y–z plane of the density and mesh at final time $t = 1.0$ on a hexahedral $20\times20\times20$ mesh without anti-hourglass forces. Top panel: classical edge viscosity at the top-left corner, original scheme at the top-right corner, scheme with velocity reconstruction at the bottom-left corner, and scheme with velocity reconstruction supplemented with anti-hourglass force (merit factor 1) at the bottom-right corners. Bottom panels: density as a function of radius for all cells in the domain for classical edge viscosity (a), original schemes (b), and scheme with velocity reconstruction (c). See Figure 8 bottom-line panel (b) for the results of the scheme with velocity reconstruction supplemented with anti-hourglass force.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

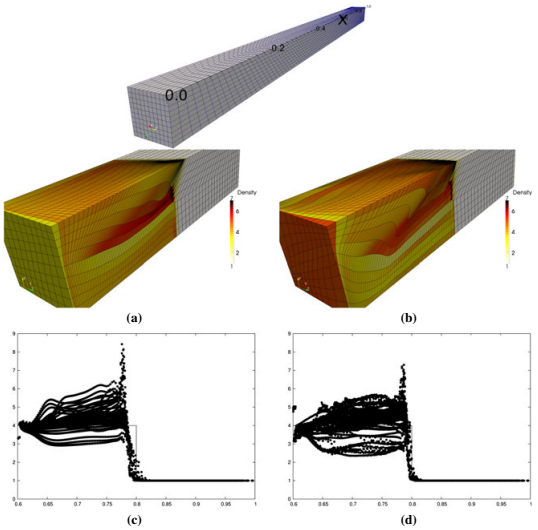


Figure 10. 3D Saltzman piston problem at final time $t = 0.6$. Top panel: initial mesh. Middle panels: (a) density and mesh for the original scheme, and (b) density and mesh for the scheme with velocity reconstruction. Bottom panels: (c) density as a function of x for all cells for the original scheme, and (d) density as a function of x for all cells for the scheme with velocity reconstruction.

as a function of radius for all cells is shown in panels (c) and (d). The expected behavior for the two schemes is observed, namely, the original scheme is dissipative, hence it damps the numerical vorticity that effectively leads to a more regular mesh. On the other hand, the scheme with velocity reconstruction, being more accurate and less dissipative, does not damp enough the parasitic instabilities. This leads to a less regular numerical solution.

We remark also that some modification of the $\mathbf{M}_{cp}$ matrix permits to improve the numerical results for instance by neglecting the off-diagonal terms. It has indeed the tendency to add more dissipation to the solution and, as a consequence, leads to a more robust scheme.

6.4. 3D Noh problem

In an eighth of the unit domain, a gas ($\gamma = 5/3$) is initiated with $\rho_0 = 1$, $\varepsilon_0 = 10^{-6}$, and $\mathbf{U}(x,y) = \left(\frac{-x}{\sqrt{x^2+y^2+z^2}}, \frac{-y}{\sqrt{x^2+y^2+z^2}}, \frac{-z}{\sqrt{x^2+y^2+z^2}}\right)$ (see [30]). A spherical shock wave is

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

generated at the origin and further propagates. The final time is chosen at $t_{\text{final}} = 0.6$. At this time, the exact solution is given by the position of the shock at $r_s = 0.2$, and the post-shock density plateau is 64.

The initial mesh is made of $30\times30\times30$ regular hexahedral cells on the unit cube (i.e., one-eighth of the domain).

The results obtained with the classical 3D compatible staggered Lagrangian scheme with edge artificial viscosity are comparable with the ones displayed in [22, Figure 4(a)]: Jets along the axes dramatically alter the symmetry of the solution. (Such non-symmetry along the axes was already triggered in the 3D Sedov problem in Figures 7(e) and 9.) These results are already well documented; the edge artificial viscosity in 2D or 3D may produce excessive numerical vorticity that sometimes leads to a degradation of the solution. Please note that this is not a boundary condition issue as such behavior is also observed when the problem is run on the full $[-1,1]^3$ domain. Using some vorticity damping artificial viscosity helps somehow to improve the general behavior [22]; however, this extra damping has lethal effect when physical instability is to be simulated.

In Figure 11, we present the density map and mesh for the original scheme and the scheme with velocity reconstruction. The latter scheme develops some numerical instabilities, which damage the numerical solution (see panel (b)). Let us remark that the numerical instabilities seem to develop along the lines $\pm x = \pm y = \pm z$ contrary to the ones from classical staggered Lagrangian schemes that develop along the axes x, y, z. Nevertheless, these instabilities have a lethal effect on the computation when the time grows. Moreover, these instabilities are still present when a refined grid is

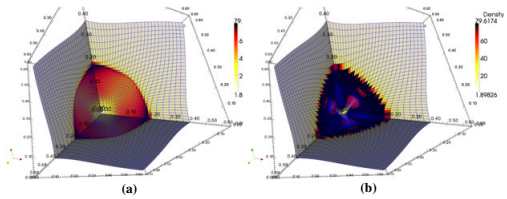


Figure 11. Noh problem on a $30\times30\times30$ hexahedral grid. Density map and mesh: (a) original scheme, (b) scheme with velocity reconstruction. The color scale is the same as that of the original scheme. The maximal density in a pinched cell along the axis is about 415.

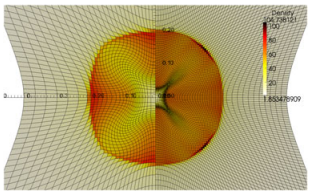


Figure 12. Noh problem on $30\times30\times30$ and $60\times60\times60$ hexahedral grids. Density map and mesh for the original scheme. Left: $30\times30\times30$ resolution (with top-bottom mirroring). Right: $60\times60\times60$ resolution (with top-bottom mirroring). The same color resolution is used for both simulations. The maximal density for the $60\times60\times60$ simulation is about 100.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

used. The original scheme is able to produce a correct numerical solution because its formulation leads to a more dissipative behavior. In Figure 12, we present the results of the original scheme for $30\times30\times30$ and $60\times60\times60$ hexahedral grids. The goal is to show the convergence of the numerical results towards the exact spherical shock wave whose radius is $r_s = 0.2$ at final time 0.6.

7. CONCLUSION

This work has proposed a 3D version of the previous work from [1]. It consists in a staggered discretization for Lagrangian hydrodynamics on general unstructured meshes. The framework uses fundamental objects of compatible discretizations such as Lagrangian subcell mass and subcell forces. An artificial viscosity form is formulated invoking the Galilean invariance and thermodynamic consistency as in [1]. The satisfaction of entropy inequality is ensured by using a subcell-based positive definite tensor, which is the cornerstone of the numerical schemes in 2D and 3D. Formally, the extension of the 2D scheme with piecewise constant velocity field in 3D is immediate, and only few specific points required clarification. A more accurate version of the scheme is further presented using a piecewise linear velocity field reconstruction and a vector limitation procedure that is frame independent. This limitation requires the choice of two directions compared with only one in 2D. A 3D code has been constructed on this framework in order to numerically study the behavior of such Lagrangian schemes.

The approach has been validated on sanity checks; namely, the 1D Sod problem and the 2D Sedov problems run with the 3D code. Then genuine 3D test cases have been run on hexahedral grids. The 3D Sedov problem showed that spherical symmetry of a shock wave can be preserved. Moreover, it shows the improvement obtained by the scheme with velocity reconstruction on regular grids. For more demanding test cases (3D Noh on hexahedral grid) or on less regular grids (3D Saltzman), the original scheme reveals its robustness. In fact, the extra dissipation of this scheme reduces its sensitivity to numerical instabilities, which are usually not sufficiently damped by the scheme with velocity reconstruction.

In the near future, we plan to investigate the behavior of different subcell-based matrices, which lead to different numerical schemes. We have observed that some modifications of the matrix presented in this work enhance the dissipative behavior of the resulting scheme. We plan to adapt the form of the subcell-based matrix within a cell depending for instance on a local measure of mesh regularity.

Finally, a last word on 3D implementation: A 3D code on moving mesh, no matter how much effort is dedicated to a proper implementation, is never a direct extension of its 2D counterpart.

ACKNOWLEDGEMENTS

The first author would like to acknowledge the fruitful and enlightening discussions that he shared with Ed Caramana. He also would like to acknowledge the financial support of the ANR (L'Agence Nationale de la Recherche in France) under the program JCJC 'ALE INC(ubator) 3D'. The last author has been partly supported by the Czech Science Foundation grant P201/12/P554 and by RVO, 68407700. The authors would like to acknowledge the support of the French embassy in Prague, Czech Republic, under the Vltava program; it provided the conditions for this work to reach maturity.

REFERENCES

1. Loubère R, Maire P-H, Váchal P. Staggered Lagrangian hydrodynamics based on cell-centered Riemann solver. *Communication in Computational Physics* 2010; **10**(4):940–978.
2. Burton DE. *Exact Conservation of Energy and Momentum in Staggered-grid Hydrodynamics With Arbitrary Connectivity*, Advances in the Free Lagrange Method. Springer-Verlag: New-York, 1990.
3. Burton DE. Multidimensional discretization of conservation laws for unstructured polyhedral grids. *Technical Report UCRL-JC-118306*, Lawrence Livermore National Laboratory, 1990.
4. Caramana EJ, Burton DE, Shashkov MJ, Whalen PP. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal of Computational Physics* 1998; **146**(1):227–262.
5. Caramana EJ, Shashkov MJ, Whalen PP. Formulations of artificial viscosity for multidimensional shock wave computations. *Journal of Computational Physics* 1998; **144**(1):70–97.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

6. Campbell JC, Shashkov MJ. A tensor artificial viscosity using a mimetic finite difference algorithm. *Journal of Computational Physics* 2001; **172**(4):739–765.
7. Kolev, Tz V, Rieben RN, A tensor artificial viscosity using a finite element approach. *Journal of Computational Physics* 2009:8336–8366.
8. Lipnikov K, Shashkov M. A mimetic tensor artificial viscosity method for arbitrary polyhedral meshes. *Procedia Computer Science* 2010; **1**(1):1932–1929.
9. Caramana EJ, Shashkov MJ. Elimination of artificial grid distortion and hourglass-type motions by means of, Lagrangian subzonal masses and pressures. *Journal of Computational Physics* 1998; **142**(2):521–561.
10. Campbell JC, Shashkov MJ. A compatible Lagrangian hydrodynamics algorithm for unstructured grids. *Selçuk Journal of Applied Mathematics* 2003; **4**(2):53–70.
11. Caramana EJ, Rousculp CL, Burton DE. A compatible, energy and symmetry preserving, Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry. *Journal of Computational Physics* 2000; **157**:89–119.
12. Després B, Mazeran C. Lagrangian gas dynamics in two dimensions and Lagrangian systems. *Archive for Rational Mechanics and Analysis* 2005; **178**:327–372.
13. Maire P-H. A high-order cell-centered, Lagrangian scheme for two-dimensional compressible fluid flows on unstructured mesh. *Journal of Computational Physics* 2009; **228**(7):2391–2425.
14. Maire P-H, Abgrall R, Breil J, Ovadia J. A cell-centered, Lagrangian scheme for compressible flow problems. *SIAM Journal of Scientific Computing* 2007; **29**(4):1781–1824.
15. Carré G, Del P, S, Després B, Labourasse E. A cell-centered, Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics* 2009; **228**(14):5160–5183.
16. Després B. Weak consistency of the cell-centered Lagrangian GLACE scheme on general meshes in any dimension. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(41):2669–2679.
17. Maire P-H, Nkonga B. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *Journal of Computational Physics* 2009; **228**(3):799–821.
18. Burbeau-Augoula A. A node-centered artificial viscosity method for two-dimensional Lagrangian hydrodynamics calculations on a staggered grid. *Communication in Computational Physics* 2010; **8**(4):877–900.
19. Christensen RB. Godunov methods on a staggered mesh—an improved artificial viscosity. *Technical Report UCRL-JC-105269*, Lawrence Livermore National Laboratory, 1990.
20. Wilkins ML. Use of artificial viscosity in multidimensional fluid dynamic calculations. *Journal of Computational Physics* 1980; **36**(3):381–403.
21. Dukowicz JK. A general, non-iterative, Riemann solver for Godunov's method. *Journal of Computational Physics* 1985; **61**:119–137.
22. Caramana E, Loubère R. "curl-q": a vorticity damping artificial viscosity for essentially irrotational lagrangian hydrodynamics calculations. *Journal of Computational Physics* 2005; **215**:385–391.
23. Bauer AL, Burton DE, Caramana EJ, Loubère R, Shashkov MJ, Whalen PP. The internal consistency, stability, and accuracy of the discrete, compatible formulation of, Lagrangian hydrodynamics. *Journal of Computational Physics* 2006; **218**(2):572–593.
24. Margolin L, Shashkov M, Smolarkiewicz P. A discrete operator calculus for finite difference approximations. *Computer Methods in Applied Mechanics and Engineering* 2000; **187**:365–383.
25. Shashkov MJ. *Conservative Finite-Difference Methods on General Grids*. CRC Press: Boca Raton, FL, 1996.
26. Desideri JA, Dervieux A. Compressible flow solvers using unstructured grids. In *Computational Fluid Dynamics*, Vol. 1 of *von Karman Inst. Fluid. Dynam. Lecture Ser. 88-05*. von Karman Inst. Fluid Dynam.: Rhode-St-Genèse, 1988; 1–115.
27. Barth TJ, Jespersen DC. The design and application of upwind schemes on unstructured meshes. *AIAA paper 89-0366, 27th Aerospace Sciences Meeting*, Reno, Nevada; 1989.
28. Sedov LI. *Similarity and Dimensional Methods in Mechanics*. Academic Press: New York, 1959.
29. Loubère R, Shashkov MJ. A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods. *Journal of Computational Physics* 2005; **209**(1):105–138.
30. Noh WF. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics* 1987; **72**:78–120.

Copyright © 2012 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

$\sim$

This chapter was devoted to the description of a family of discrete compatible Lagrangian schemes and some of related investigations. As previously mentioned our primary goal was to build an ALE simulation code for compressible hydrodynamical flow. As such this family of staggered Lagrangian scheme is only one of the building brick of such a code. Presumably this is the most important brick as the Lagrangian scheme may be considered as the "engine" of any ALE code. As a consequence some of our investigations were specifically dedicated to better understand such family of numerical schemes.

The next chapter presents some of our investigations related to the two other parts of an ALE code, namely the rezone and remap steps.

Arbitrary-Lagrangian-Eulerian schemes

2

Contents

In numerical simulations of multidimensional fluid flow, the relationship between the motion of the computational grid and the motion of the fluid is an important issue. Two choices that are typically made representing either a Lagrangian framework, in which the mesh moves with the local fluid velocity, or an Eulerian framework, in which the fluid flows through a grid fixed in space. More generally, however, the motion of the grid can be chosen arbitrarily. The philosophy of the Arbitrary Lagrangian-Eulerian methodology (ALE ; cf. [107, 108, 52, 109, 110, 111, 112, 113]) is to exploit this degree of freedom to improve both the accuracy and the efficiency of the simulation. The main elements of many ALE algorithms are an explicit Lagrangian phase, a rezone phase in which a new grid is defined, and a remap phase which transfers the Lagrange solution onto the new grid [109]. Most ALE codes use a grid of fixed connectivity that, in two spatial dimensions, is formed by quadrilaterals or by a mix of quadrilaterals and triangles, the latter being considered as degenerate quadrilaterals. Ultimately, we are interested in the development of ALE methods for meshes whose connectivity may change during the calculation. In such methods, the total number of cells may change with time, as well as the number of edges bounding each cell, leading to the appearance of general polygonal cells.

As a first step toward this goal, in section 2.2, we present some of our contributions to the context of ALE methods on a mesh with fixed connectivity, but we allow the mesh to contain general polygonal cells. Extending the ALE methodology to this more general mesh is valuable in itself as it simplifies the setup process for computational domains with complex geometrical shapes and helps to avoid artificial mesh imprinting due to the restrictions of a purely quadrilateral mesh, [75, 114]. In this section we mainly focus on the remapping stage of the ALE methodology using the compatible staggered Lagrangian scheme presented in the previous chapter. This also includes the repair technique to conservatively corrected unphysical remapped variables. Then in section 2.3 we extend the fixed ALE methodology to a reconnection ALE, further called ReALE, to allow meshes

whose connectivity may change during the calculation. In this case only the rezone part of the ALE algorithm is revamped using the Voronoi tesselation machinery.
In section 2.4 we present some investigations we have performed to compare three ALE codes on representative and demanding test cases. Namely we have compared the following codes : CHIC at CELIA, University of Bordeaux, PALE code standing for 'Prague ALE', from Prague's team in Czech Republik, and ALE INC(ubator) at IMT, University of Toulouse. In section 2.5 we present some specifics techniques to deal with multi-material fluid flows. More specifically we focus on order-independent interface reconstruction technique to deal with more than three materials in mixed cells. Some numerical results are provided throughout the chapter to show the behaviors of the proposed techniques.

Let us begin by resetting the context and point why the research pursued was, to some degree, justified.

2.1 History and presentation

As previously stated the ALE methodology chosen for the code ALE INC(ubator) is built on three successive phases : Lagrangian, rezone and remap.

Lagrangian phase. We consider the discrete compatible staggered Lagrangian scheme presented and studied in the previous chapter, see chapter 1. Notice that this scheme employs staggered variables (cell-centered density and energy, vertex-based velocity) on general polygonal mesh. Moreover this scheme does need the subcell-based density and pressure which are further used to compute subpressure forces that fight back hourglass parasital grid motion, see the previous chapter for more details. These remarks are of great importance when the remap phase will be defined.

Rezone phase. The rezone phase consists in defining the new grid onto which the conservative variables are to be remaped. This new grid must be "better" than the previous Lagrangian grid. However providing a unambiguous definition of "better" in the previous sentence is a genuine problematic point.
It is more or less agreed [1] that a better mesh must be smoother than the Lagrangian mesh. In fact in the literature the rezone phase is often called the smoothing phase. One of the most cited name in this field is probably A. Winslow [115, 116] who has used the property of elliptic regularity to smooth grids. An elliptic operator is applied to the node positions leading to a better node equidistribution and to a relaxed Lagrangian mesh. The main drawback of this technique lays in its underlying metrics. Usually metrics are based on Euclidean distance and equidistribution of length, surface and angle between edges. Consequently this implies that according to this measure the "best" quadrangle is a square, the "best" triangle is an equilateral triangle, and, generaly the "best" polygon is the associated regular polygon. If the initial mesh is not an optimal mesh for this underlying metrics then the rezone strategy instantaneously starts to improve the mesh, even if no physical motion has occured yet. In other words in some part of the domain the initial and supposedly valid cells (because the user has provided a good enough mesh) are reshaped by the rezoning strategy before any physical process has started. This is not acceptable. Some cures have been developed such as zone/vertex triggers. Given constraints these triggers keep a cell/vertex Lagrangian, that is to say unmodified, by the rezoner. Quoting A.Barlow [117] "Constraints [...]

1. but it is not an issue without contention !

keep a node Lagrangian until some condition is reached e.g. element quality criterian or physical condition is reached in surrounding elements", see also [118] for an example of triggers. Such triggers are activated according to constraints and condition which are not obvious to derive. In other words it seems that there is nowadays no universal rezoner which satisfies the whole ALE community.

For another important contributor to the field of mesh generation, mesh metrics consult P. Knupp's articles, as instance [119, 120, 121, 122, 123, 124].

It is agreed that the new mesh must be at minima valid for the numerical Lagrangian method. In our case one requires that each cell remains a convex polygon. In the unlikely situation where the Lagrangian grid is tangled, before applying any rezoning technique, one applies the untangling procedure described in [125] and depicted in Fig. 2.1. This procedure computes the feasible set for a problematic node. More precisely for an invalid polygon and a bad node one determines the feasible set which is the valid space for the node to move in, that further leads to surrounding valid polygons. Although this feasible set may not always exist the authors of [125] have provided an extension of their method to deal with such a situation. Amazingly this feasible set approach can also be used as a mesh smoother [118]. In ALE INC(ubator) several smoothing techniques have been

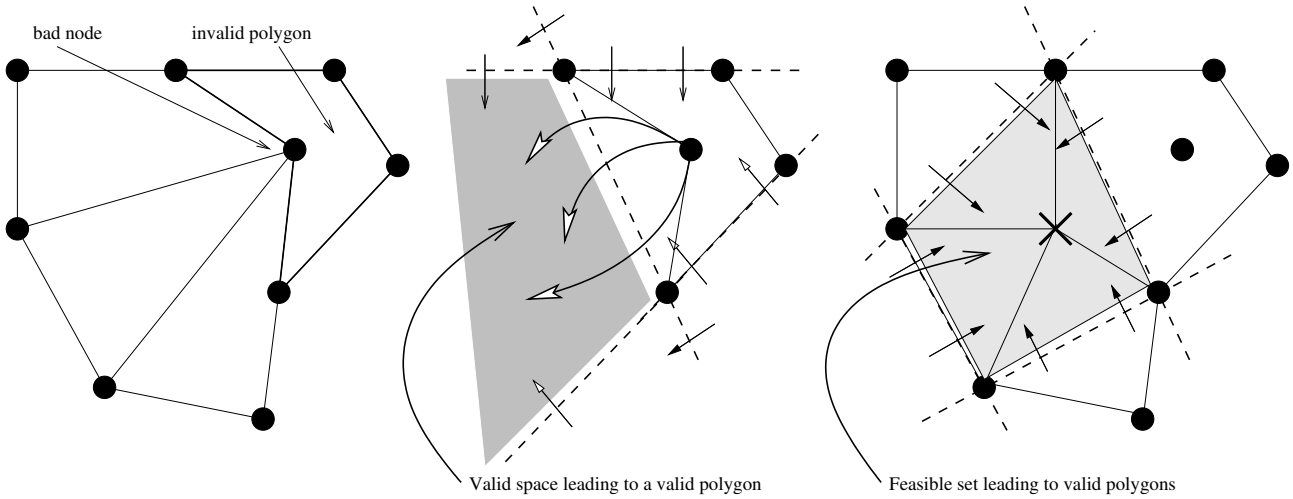


FIGURE 2.1 – *Description of the untangling technique from [125]. For an invalid polygon and a bad node one determines the feasible set which is the valid space for the node to move in that further leads to surrounding valid polygons. The middle panel shows the feasible set if one only considers to fix the invalid polygon only. The right panel shows the final feasible set leading to valid neighbor polygons.*

implemented. Winslow smoothing and the Reference Jacobian rezoning technique from [124], see the details in this paper. Our main contribution to this rezone phase is the mesh reconnection that is described in section 2.3.

One related issue with rezone and remap is the strategy to decide when a rezone and remap step is needed. This point has been little noticed and very few investigations have been carried out mainly because the rezoning step already demands several parameters to be fixed, adding two or three more parameters for the strategy does not weaken the entire process. Nevertheless we have tried to develop an automatic choice of rezone and remap strategy for our ALE code in [126]. The goal of this report is to gather several "measures" of solution quality to help the ALE code to develop its own capability to detect when and how often to rezone and remap. We skip this description but refer the reader to [126] for details.

Remap phase. The remapping phase is considered as conservative transfer (or advection) of the physical quantities from the Lagrangian mesh onto the rezoned and smoother mesh. To ease the phrasing we call the Lagrangian mesh "old" and the rezoned mesh "new", likewise for any entity defined on the Lagrangian or rezoned mesh.
An exhaustive list of contributors to this field is almost impossible as this should embrace the key words interpolation techniques, advection methods, (flux-corrected) transport methods and remapping *per se*. However some very much related works are to be found in [52, 127, 128, 129, 7, 130, 131, 132, 133, 134, 135]. In this introduction we only consider the so-called unsplit methods (as noticed by Benson [52] "despite the fact that they were never split in the first place"). By the way one urges the reader to refer to the review made by Benson [52] to have an overview of legacy remapping methods.
At first glance defining a remapping technique between an old and new mesh for a single conserved variable, say the mass, being the density multiplied by a surface, is fairly easy. First one defines a conservative representation of the old density on the old mesh ; a piecewise constant representation in a finite volume sense. Second the exact geometrical intersection between a new cell and the old mesh is computed, it consists of a set of polygons that pave the new cell without overlapping and gap. Third the new mass in the new cell is computed as the sum of all old masses present in the intersection polygons (the mass is computed as the density integrated over the intersection polygons), see Fig. 2.2 left. This straighforward method has some drawbacks :

1. it is an exact method only for a constant density function leading, in some sense, to a first-order accurate method,
2. the exact intersection of two polygonal meshes is a demanding algorithm to implement and, most of all, a relative expensive method.

This has led several authors to reconstruct the underlying function as a piecewise linear function [127, 128]. If so the remapping method is exact for linear function and, as such, is considered as a second-order accurate method. However generation of non-physical remapped quantities enforces the utilization of slope or flux limiters in the reconstruction. Such limiters may not always be trivial to properly define as instance when a vector field is to be reconstructed. Usually the limitation is independently applied to each component leading to a frame dependent limitation [2].
On the other hand to overcome the cost of the exact intersection, in the case of a fixed connectivity rezoning, some authors have proposed to use a kind of donor cell method [3]. This method is often refered to as the *swept region remapping* [127, 128] because in the 2D context this method considers the motion of each edge from its old position to its new position, see Fig. 2.2 right. This method is tremendously less expensive than the exact intersection remapping as only the region swept by the edges of a generic cell are to be computed. Between two cells sharing an edge the swept region implicitely determines a "donor" cell. This donor cell donates the integrated quantity of conservative variable over the swept region to the second cell. The swept region remapping has some drawbacks also. The mass flux can only occur between cells sharing one edge, corner cells as a consequence never interact (as instance cells 0 and 2 in Fig. 2.2-right). In other word the flux may be evaluated from an inappropriate cell (edge sharing cell instead of corner cell) [137]. Moreover the swept region may auto-intersect or may be inaccurately computed, respectively observe the red swept region and green/magenta regions on Fig. 2.2 on the right panel. These geometrical errors can however be overcome at little cost as instance with clever techniques such as the ones developed in [138].

2. This is one of the reasons why frame invariant limiters [23, 136] have been recently designed.
3. This corresponds to the forward in time upwind scheme for a transport equation at constant speed.

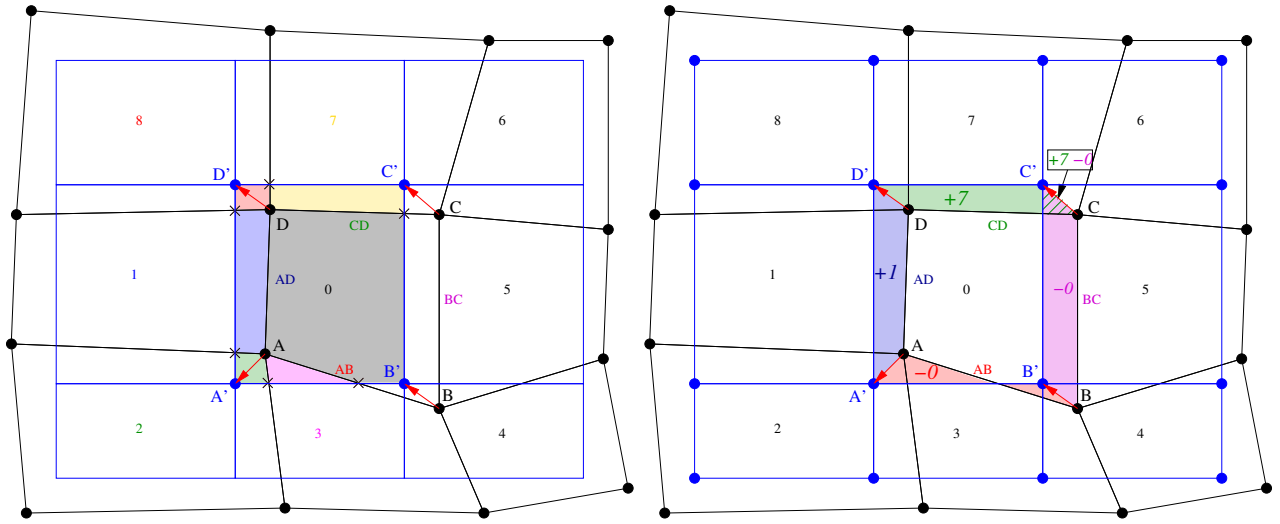


FIGURE 2.2 – *Remapping methods from an old black mesh onto a new blue mesh keeping the same connectivity. Cell* 0 *is surrounded by cells* $1, \ldots, 8$. *— Left : exact intersection based remapping. New blue cell* 0 *is paved with small pieces of old neighbor cells. Red polygon corresponds to the intersection with corner black cell* 8 *(red number), blue polygon to the intersection with left black cell* 1 *(blue number), grey polygon to the surface of old cell* 0 *remaining in new cell* 0. *— Right : swept region method. This method only considers flux between cells across a common edge. Edge* AD *moves from its old position to its new position* $A'D'$*, doing so it sweeps an area* $ADD'A'$*. The new mass is updated taking some mass from the donor cell* 1 *(hence the* $+1$ *in the picture). Sometimes swept areas overlap (dashed green and magenta triangle), or some swept area auto-intersects (swept region of edge* AB *in red), this leads to second-order errors.*

Nonetheless the remapping of a single cell-centered variable apart from second-order errors is exactly or approximately achieved. One important difficulty rises because of the staggered placement of variables to be remapped. While density and mass are located at cells (or subcells if anti-hourglass force is used), velocity is defined at points. Momentum is therefore the dual cell centered value (we will also say point centered value which is an abuse of notation)

$$\boldsymbol{Q}_p = m_p \boldsymbol{U}_p, \tag{2.1}$$

with the mass point $m_p = \sum_{c \in \mathcal{C}(p)} m_{cp}$. More precisely the momentum is defined on the dual mesh (a dual cell is defined by the subcells around a point). Consequently the momentum should be conservatively remapped from the Lagrangian dual mesh onto the rezoned dual mesh. However the mass in the dual cell has presumably changed impacted by the cell-centered mass remap phase. Properly taking into account this interleaved cell-centered mass remap and point-centered momentum remap is not obvious and usually demands some additional approximations which may or may not be properly justified. The reader is refered to [52] section 3.5 for an overview of the complication brought by staggered placement of velocity and density variables when conservative momentum remapping is desired.

Furthermore energy remapping phase still needs to be discussed. At first glance either internal to total energy can be remapped. Most of staggered hydrocodes remap internal energy like any other conserved quantity. Therefore total internal energy is conserved. Unfortunately momentum conservation does not imply kinetic energy conservation. Consequently the total energy as the sum of kinetic and internal energies decreases in time because of the numerical diffusion generated during the momentum remap. Unavoidably total energy conservation is lost with the unpleasant possibility that shock waves may weaken. Some tricks are then triggered to reduce the loss of total energy at shock fronts, see Benson [52] (section "advecting energy") like the drastic conversion

of kinetic energy losses into internal energy. In some sense these tricks are intended to "repair" the damages brought by the remapping phases. Alternative strategies have also been considered like the remap of the total energy then the deduction of the internal energy from substraction to the kinetic energy deduced from remapped momentum. However monotonicty of the resulting internal energy is not ensured which, more or less, may lead to unrealistic heating of materials [139] [4] If so retrieving physical relevant quantities may demand some 'boarderline' repair actions [5]. As a matter of fact the problem of interleaved remap phases of conserved quantities defined on different geometrical entities is not a trivial task when mass, momentum and total energy conservation is required in addition to the fact that physical relevant variables must ultimately be provided [6].

Some of our investigations on ALE falls within this framework and we present in the next section a selection of published works.

4. Alternative ways exist. In [140] the authors propose a potentially kinetic-energy-conservative algorithm for remapping nodal velocity in a staggered Lagrangian scheme, the improved algorithm is based on the minimization of a functional which may introduce oscillations in the velocity remapped field. Consequently the authors suggest to combine this approach with the low-order donor method by flux-corrected remap (FCR).

5. By 'boarderline' we explicitely emphasize the fact that sometimes for a simulation to run to completion, some shameful 'arrangements' with physcics have to be taken.

6. This somehow explains why effective cell-centered Lagrangian schemes are of particular importance in the context of ALE.

2.2 Arbitrary-Lagrangian-Eulerian (ALE)

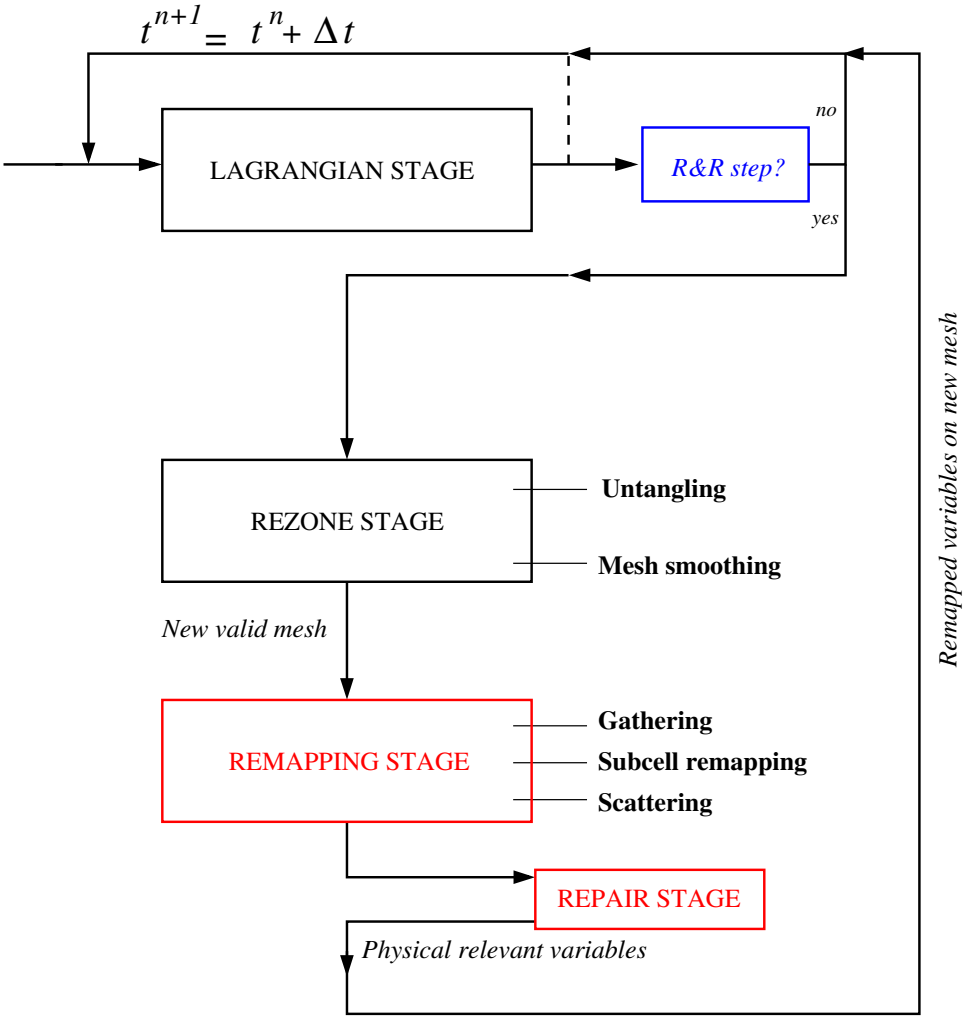


Figure 2.3 – *Description of the ALE scheme within ALE INC(ubator). A Lagrangian scheme is followed by a rezone phase which determines a smoothed mesh onto which a remapping phase conservatively projects the physical variables and ultimately provides physical relevant variables (thanks to a so-called repair technique). Our contributions mainly focus on the red boxes.*

A simplified algorithm of the ALE method implemented into ALE INC(ubator) is depicted in Fig. 2.3. In the following sections we mainly focus on our papers dealing with the "remapping stage" and the "repair stage" of this algorithm (the red bloxes in the figure).

2.2.1 Remapping

While developing the 2D ALE code ALE INC(ubator) with M.J Shashkov at the Los Alamos National Laboratory [5] we had to face the situation of remapping cell-centered density and specific internal energy and nodal velocity under the constraints of mass, momentum and total energy conservation for general polygonal mesh. These variables are the ones provided by the compatible staggered Lagrangian scheme [55, 63, 64, 12] described in the previous chapter. The staggered placement of variables complexifies the notion of conservation as the remapped cell-centered entities

must somehow be consistent with the remapped vertex-based entities.

In article [7] entitled *"A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods"*, M.J. Shashkov and I proposed a solution to this problem.
Before describing our solution to this interleaved remapping phases let us make a general comment related to the compatible staggered Lagrangian discretization that was eluded during the derivation of the scheme but is of some importance for the remapping phase of the ALE method. As already seen the primary energy variable is the cell centered specific internal energy ε_c. Moreover the conserved total energy used to derive the compatible staggered Lagrangian formulation, in other words the fact that total energy is conserved to round-off error in the whole domain, is implicitely obtained considering the subcell-centered total energy

$$E_{cp} = m_{cp}\varepsilon_c + \frac{1}{2}m_{cp}\|\boldsymbol{U}_p\|^2. \tag{2.2}$$

This derives from the total kinetic energy equation (1.18) and total internal energy equation (1.19) the sum of which defines the total energy (1.20) as a global entity over the domain (see the derivation of the compatible scheme in section 1.2.2)

$$E = \sum_c m_c\varepsilon_c + \sum_p \frac{1}{2}m_p\|\boldsymbol{U}_p\|^2.$$

The subcell-centered total energy (2.2) is revealed using the previous equation and the definition of cell mass m_c (see (1.10)) because

$$\begin{aligned} E = \sum_c \left(m_c\varepsilon_c + \sum_{p\in\mathcal{P}(c)} \frac{1}{2}m_{cp}\|\boldsymbol{U}_p\|^2 \right) &= \sum_c \left(\left(\sum_{p\in\mathcal{P}(c)} m_{cp} \right) \varepsilon_c + \sum_{p\in\mathcal{P}(c)} \frac{1}{2}m_{cp}\|\boldsymbol{U}_p\|^2 \right) \\ &= \sum_c \sum_{p\in\mathcal{P}(c)} m_{cp} \left(\varepsilon_c + \frac{1}{2}\|\boldsymbol{U}_p\|^2 \right) = \sum_c \sum_{p\in\mathcal{P}(c)} E_{cp}, \end{aligned}$$

leading to the definition of the subcell-centered total energy (2.2) given above. Remark that working on dual cells produces the same subcell-centered total energy definition because

$$\begin{aligned} E = \sum_p \left(\sum_{c\in\mathcal{C}(p)} m_{cp}\varepsilon_c + \frac{1}{2}m_p\|\boldsymbol{U}_p\|^2 \right) &= \sum_p \left(\sum_{c\in\mathcal{C}(p)} m_{cp}\varepsilon_c + \frac{1}{2} \left(\sum_{c\in\mathcal{C}(p)} m_{cp} \right) \|\boldsymbol{U}_p\|^2 \right) \\ &= \sum_p \sum_{c\in\mathcal{C}(p)} m_{cp} \left(\varepsilon_c + \frac{1}{2}\|\boldsymbol{U}_p\|^2 \right) = \sum_p \sum_{c\in\mathcal{C}(p)} E_{cp}. \end{aligned}$$

Therefore a compatible definition of total energy with the discrete staggered Lagrangian scheme reveals that total energy must be a subcell-centered entity.
In addition our discretization employs subcell masses that serve to introduce anti-hourglass force [56], see section 1.2.3. This adds an additional requirement to the remap phase — that the subcell densities (corresponding to subcell masses) have to be conservatively interpolated.

As a consequence the main goal of the work in [7] is to build subcell-centered conservative mass, momentum and total energy entities, remap based on subcells and finally recover the primary variables by association of new subcell remapped entites.

In this work, we assume that the rezone algorithm produces mesh that is "close" to the Lagrangian mesh so that a local remapping algorithm (i.e, where mass and other conserved quantities are only exchanged between neighboring cells) can be used. The swept remapping is used in practice although an exact intersection algorithm has been also implemented and tested.
Our new remapping algorithm consists of three stages.

First : *gathering stage*. We define mass, momentum, internal energy, and kinetic energy in the subcells. They are defined in such a way that the corresponding total quantities (defined as the sums over subcells) are the same as those at the end of the Lagrangian phase, ensuring that the gathering stage is conservative.

Second : *subcell remapping stage*. We use the algorithm described in [128] to remap mass, momentum, internal, and kinetic energy from the subcells of the Lagrangian mesh to the subcells of the new rezoned mesh. This algorithm is linearity-preserving and computationally efficient. It consists of a piecewise linear reconstruction and an approximate integration based on the notion of swept regions. The algorithm does not require finding the intersections of the Lagrangian mesh with the rezoned mesh, which contributes to its efficiency. The algorithm is conservative : total mass, momentum, internal and kinetic energy over subcells of the rezoned mesh are the same as mass, momentum, internal and kinetic energy over subcells of Lagrangian mesh. The total energy is also conserved, being the sum of (individually conserved) internal and kinetic energies.

Third : *scattering stage*. We recover the primary variables — subcell density, nodal velocity, and cell-centered specific internal energy — on the new rezoned mesh.
- Subcell density is recovered by using the remapped mass and volume of the subcell of the rezoned mesh. The subcell masses and the corresponding densities are then adjusted using a conservative repair procedure [128], [141], [142], [8], [9] to enforce local bounds (see also the next section for details), which may be violated during the remapping stage. This produces the final subcell density and the corresponding subcell mass that will be used in next time step.
- Next, we define the remapped nodal momenta using the remapped subcell momenta, in such a way that total momenta is conserved. New velocity components are defined by dividing by nodal mass. Then nodal velocity is repaired, resulting in the final velocity that will be used to move the point during the Lagrangian phase in the next computational cycle.
- To enforce the conservation of total energy, the discrepancy between the remapped kinetic energy in the cell and the kinetic energy that is computed from the remapped subcell masses and the final nodal velocities is contributed to the remapped internal energy in the cell. Finally, the internal energy and the corresponding specific internal energy are conservatively repaired.

Our remapping algorithm satisfies the following requirements :
- *Conservation*. The total mass, momenta and energy of the new mesh must be the same as that of the old mesh. This property, combined with the same conservation properties of the Lagrangian phase, guarantees the conservation of the overall ALE method.
- *Bound-preservation*. The remapped density, velocity components and internal energy have to be contained within physically justified bounds, which are determined from the corresponding fields in the Lagrangian solution. For example, density and internal energy have to be positive. Moreover, because we assume that the new mesh is obtained from a small displacement of the old mesh, one can require that the new values lie between bounds determined by the values of its neighbors on the old mesh, [128, 141, 8, 9].

- *Accuracy*. It is straightforward to define accuracy in the remap of density ; we will require that the remap of density is linearity-preserving. That is, if the values on the old mesh are obtained from a global linear function, then the values on the new mesh have to coincide with the values of the same linear function on the new mesh. For the remap of velocity, there are several different notions related to accuracy. For example, one widely used test of consistency is the so-called DeBar condition (see for example [52]) which can be stated as follows : if a body has a uniform velocity and spatially varying density, then the remapping process should exactly reproduce a uniform velocity. For internal energy unfortunately the situation is more complicated. We have demonstrated the accuracy of our new algorithm through the practical expedient of well-chosen test problems.
- *Reversibility*. If the new and old meshes are identical, then the remapped primary variables should endure no change. This property is closely related to the notion of being free of inversion error, see [52], where it is stated that if the new and old grids coincide, then the remapped velocity on new mesh should coincide with the velocity on the old mesh.

We have also demonstrated computationally that our new remapping method is robust and accurate for a series of test problems in 1D (Sod shock tube, Collela-Woodward blastwave, Le Blanc shock tube) and 2D (Sedov problem on quadrangular and polygonal meshes).
As instance in Fig. 2.4 one reproduces the results of the ALE scheme in its Eulerian, Lagrangian and ALE regimes for the Sedov problem on polygonal grid.

Using the remapping method developed in this paper we have constructed a full staggered ALE code working for polygonal meshes. The method combines and generalizes previous work on the Lagrangian and rezoning phases [55, 124, 125], and includes this new remapping algorithm [7]. This work has been implemented into the code ALE INC(ubator) [5].

This paper is reproduced in the following pages.

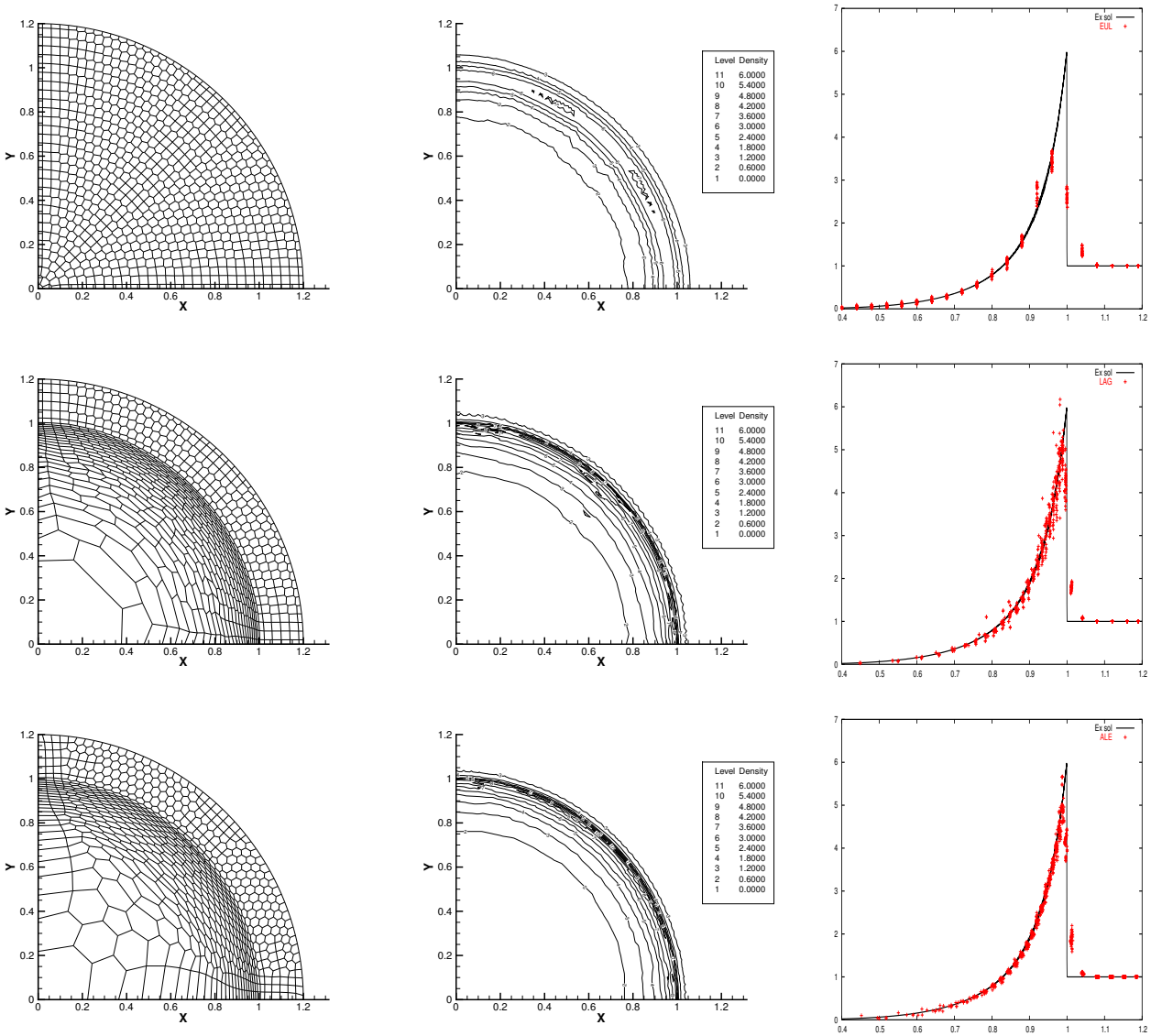


FIGURE 2.4 – *Numerical results from paper [7]. Sedov problem on polygonal Mesh — Mesh (left column), density isolines (middle column) and density as function of radius for all cells (right column) at $t = 1.0$ — Eulerian regime (top line), Lagrangian regime (middle line), ALE regime (bottom line).*

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 209 (2005) 105–138
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods

Raphaël Loubère, Mikhail J. Shashkov *

Los Alamos National Laboratory, T-7, MS B284 Los Alamos, NM, USA

Received 27 September 2004; received in revised form 3 March 2005; accepted 17 March 2005
Available online 4 May 2005

Abstract

We describe a new remapping algorithm for use in arbitrary Lagrangian–Eulerian (ALE) simulations. The new features of this remapper are designed to complement a staggered-mesh Lagrangian phase in which the cells may be general polygons (in two dimensions), and which uses subcell discretizations to control unphysical mesh distortion and hourglassing. Our new remapping algorithm consists of three stages. A *gathering stage*, in which we interpolate momentum, internal energy, and kinetic energy to the subcells in a conservative way. A *subcell remapping stage*, in which we conservatively remap mass, momentum, internal, and kinetic energy from the subcells of the Lagrangian mesh to the subcells of the new rezoned mesh. A *scattering stage*, in which we conservatively recover the primary variables: subcell density, nodal velocity, and cell-centered specific internal energy on the new rezoned mesh. We prove that our new remapping algorithm is conservative, reversible, and satisfies the DeBar consistency condition. We also demonstrate computationally that our new remapping method is robust and accurate for a series of test problems in one and two dimensions.
© 2005 Elsevier Inc. All rights reserved.

PACS: 65N05; 65N10; 65N15; 80A20

Keywords: Remapping; Conservative interpolation; ALE methods

1. Introduction and background

In numerical simulations of multidimensional fluid flow, the relationship between the motion of the computational grid and the motion of the fluid is an important issue. Two choices that are typically made represent either a Lagrangian framework, in which the mesh moves with the local fluid velocity, or an Eulerian framework, in which the fluid flows through a grid fixed in space. More generally, however, the motion of the grid can be chosen arbitrarily. The philosophy of the arbitrary Lagrangian–Eulerian methodology (ALE; cf. [14,3,4,20,15,16,26]) is to exploit this degree of freedom to improve both the accuracy and the efficiency of the simulation. The main elements of most ALE algorithms are an explicit Lagrangian phase, a rezone phase in which a new grid is defined, and a remap phase in which the Lagrange solution is transferred to the new grid [20].

Most ALE codes use a grid of fixed connectivity that, in two spatial dimensions, is formed by quadrilaterals or by a mix of quadrilaterals and triangles, the latter being considered as degenerate quadrilaterals. Ultimately, we are interested in the development of ALE methods for meshes whose connectivity may change during the calculation. In such methods, the total number of cells remains fixed, but the number of edges bounding each cell may change with time, leading to the appearance of general polygonal cells. As a first step toward this goal, here we consider ALE methods on a mesh with fixed connectivity, but allow the mesh to contain general polygonal cells. Extending the ALE methodology to this more general mesh is valuable in itself as it simplifies the setup process for computational domains with complex geometrical shapes and helps to avoid artificial mesh imprinting due to the restrictions of a purely quadrilateral mesh, [6,7].

In the rest of this introductory section, we will present notation related to a general polygonal staggered mesh, will review algorithms for the Lagrangian phase and rezone phase as presented in [8,9,17,28,33,32], and finally will describe the main ideas of our new remap procedure, which is the main topic of this paper.

1.1. Polygonal mesh

We consider a two-dimensional computational domain Ω, assumed to be a general polygon. We assume we are given a mesh on Ω whose cells, $\{c\}$, cover the domain without gaps or overlaps. Each cell may be a general polygon, and is assigned an unique index that for simplicity will also be denoted by c. The set of vertices (nodes) of the polygons is denoted by $\{n\}$, where each node has an unique index n. Then each cell can be defined by an ordered set of vertices. We denote the set of vertices of a particular cell c by $N(c)$. Further, we denote the set of cells that share a particular vertex n by $C(n)$. Note that each vertex may be shared by an arbitrary number of cells. We will subdivide each cell into a set of quadrilaterals that we will term subcells. A pair of indexes c and n uniquely defines a quadrilateral, identified as subcell cn; this subcell is constructed by connecting the geometrical center of the cell c with the middle points of cell faces having the same node n as one end point and the node itself (see Fig. 1). Hence each cell can be divided uniquely into quadrilaterals (subcells or corners).

We denote the cell and subcell volumes (in 2D Cartesian geometry these are areas) by $V(c)$ and $V(cn)$, where by construction $V(c) = \sum_{n\in N(c)} V(cn)$. A nodal volume can be defined as the sum of the volumes of subcells shared by the node n, i.e., $V(n) = \sum_{c\in C(n)} V(cn)$.

1.2. Lagrangian phase

The equations of Lagrangian gas dynamics can be written as

$$\frac{1}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\mathbf{div}\,\mathbf{u}, \qquad \rho\frac{\mathrm{d}u}{\mathrm{d}t} = -\mathbf{grad}\,p, \qquad \rho\frac{\mathrm{d}\varepsilon}{\mathrm{d}t} = -p\ \mathbf{div}\,\mathbf{u}, \tag{1.1}$$

where ρ is the density, p is the pressure, ε is the specific internal energy, and $\mathbf{u} = (u, v)$ is the velocity. The pressure is linked to density and specific internal energy via an equation of state: $p = p(\rho, \varepsilon)$. This system of Eq. (1.1) is solved by the Lagrangian phase.

* Corresponding author. Tel.: +1 505 667 4400; fax: +1 505 665 5757.
E-mail addresses: loubere@lanl.gov (R. Loubère), misha@t7.lanl.gov, shashkov@lanl.gov (M.J. Shashkov).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.03.019

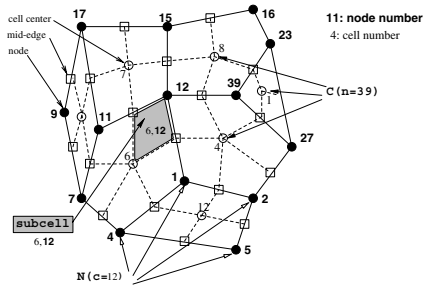


Fig. 1. Grid and notations. The ○ are the cell centers, the ● are the nodes (vertices), the □ are the mid-face (edge) points. The set of vertices for cell $c = 12$ is $N(c = 12) = \{\mathbf{5,2,1,4}\}$, and the set of cells sharing node, $n = 39$ is $C(n = \mathbf{39}) = \{1,8,4\}$. The gray subcell, $cn = 6,12$ is the quadrilateral defined by connecting the geometrical center of the cell $c = 6$ with middle points of cell faces having the same node $n = \mathbf{12}$ as one end point and the node itself.

A discretization of the gas dynamic equations for the Lagrangian phase of the ALE method for a mesh consisting of general polygons is described in [8,9], based on the philosophy of compatible hydrodynamic discretization [12]. This discretization assumes a staggered grid, where the components of the velocity vector are defined at the nodes (vertices) of the cells, $\mathbf{u}(n) = (u(n), v(n))$, and where the thermodynamic variables density $\rho(c)$ and internal energy $\varepsilon(c)$ are defined at the cell centers. In addition to nodal and cell-centered quantities, this discretization employs as additional variables the densities of the subcells, $\rho(cn)$. The preservation of subcell mass during the Lagrangian phase of the calculation introduces new forces that prevent artificial grid distortion and hourglass patterns. This enhancement of the Lagrangian algorithm was shown to be effective both for quadrilateral meshes [10], as well as for polygonal meshes [8]. The Lagrangian phase including subcell forces, is conservative; i.e., discrete forms of mass, momentum, and total energy are conserved [12]. The use of subcell masses and corresponding densities places new requirements on the remap phase of an ALE method because these subcell densities have to be remapped in addition to the usual remapping of the primary variables—nodal velocities, cell-centered densities and internal energies.

We define the subcell mass in terms of the primary cell variables as follows:

$$m(cn) = \rho(cn)V(cn). \tag{1.2}$$

Then the mass of the cell and of the node are defined

$$m(c) = \sum_{n\in N(c)} m(cn), \quad m(n) = \sum_{c\in C(n)} m(cn). \tag{1.3}$$

All of these masses are employed in the Lagrangian phase of our ALE method. Since the subcell mass, $m(cn)$ is assumed to be Lagrangian and so does not change with time, it follows that:

$$\rho(cn) = m(cn)/V(cn), \tag{1.4}$$

which serves as a definition of the subcell density for a given subcell mass. The masses of the individual cells and nodes are also Lagrangian because they are sums of the masses of the associated subcells. The mass of

the cell is used in the equation for the internal energy, while the mass of the node is used in the momentum equation. Finally, by definition we have

$$\rho(c) = \frac{m(c)}{V(c)} = \frac{\sum_{n\in N(c)} m(cn)}{\sum_{n\in N(c)} V(cn)}.$$

The total mass, M, which is conserved in the Lagrangian phase is

$$M = \sum_{cn} m(cn) = \sum_{c} m(c) = \sum_{n} m(n). \tag{1.5}$$

On the staggered mesh, momentum is most naturally defined at the nodes

$$\mu(n) = m(n)u(n), \quad v(n) = m(n)v(n), \tag{1.6}$$

or equivalently

$$u(n) = \mu(n)/m(n), \quad v(n) = v(n)/m(n). \tag{1.7}$$

Note that as a result of the remap stage we will have new momenta and masses at the nodes, so to recover velocities we will use (1.7) as the definition of velocities for given momenta and nodal mass. The total momentum components, μ_u, μ_v, which are individually conserved in the Lagrangian phase, are

$$\mu_u = \sum_{n} m(n)u(n), \quad \mu_v = \sum_{n} m(n)v(n). \tag{1.8}$$

It will be useful to define a cell-centered momenta as

$$\mu(c) = \sum_{n\in N(c)} m(cn)u(n), \quad v(c) = \sum_{n\in N(c)} m(cn)v(n). \tag{1.9}$$

Using this definition and the definition of nodal mass, the total momentum components (μ_u, μ_v)—see Eq. (1.8)—can be expressed as

$$\mu_u = \sum_{c} \mu(c), \quad \mu_v = \sum_{c} v(c). \tag{1.10}$$

Kinetic energy is also most naturally defined at the nodes

$$K(n) = m(n)\frac{|\mathbf{u}(n)|^2}{2}. \tag{1.11}$$

The internal energy is naturally defined at the cells

$$\mathscr{E}(c) = m(c)\varepsilon(c). \tag{1.12}$$

In analogy to (1.7), Eq. (1.12) can be used after the remap phase to define $\varepsilon(c)$ given $\mathscr{E}(c)$ and $m(c)$

$$\varepsilon(c) = \mathscr{E}(c)/m(c). \tag{1.13}$$

The total energy, which is also conserved in the Lagrangian phase, is

$$E = \sum_{c} \mathscr{E}(c) + \sum_{n} K(n). \tag{1.14}$$

Later we will require the concept of a cell-centered kinetic energy, which we define as follows:

$$K(c) = \sum_{n\in N(c)} m(cn)\frac{|\mathbf{u}(n)|^2}{2}. \tag{1.15}$$

Using this definition and the definition of nodal mass, the total energy, E (see formula (1.14)), can be expressed as

$$E = \sum_c (\mathscr{E}(c) + K(c)). \tag{1.16}$$

By introducing total internal and kinetic energies as

$$\mathscr{E} = \sum_c \mathscr{E}(c), \quad K = \sum_c K(c), \tag{1.17}$$

we finally can express the total energy as

$$E = \mathscr{E} + K. \tag{1.18}$$

1.3. Rezone phase

In the rezone phase, we use the reference Jacobian matrix (RJM) strategy described in [17,28]. The RJM rezone algorithm is based on a nonlinear optimization procedure that requires a valid mesh as an initial guess, and so it may be necessary to untangle the mesh (see e.g., [33,32]) prior to rezoning. The RJM rezone strategy ensures the continuing geometric quality of the computational grid, while keeping the "rezoned" grid at each time step as close as possible to the Lagrangian grid. Sets of cells and nodes of rezoned mesh will be denoted by $\{\tilde{c}\}$ and $\{\tilde{n}\}$, respectively.

When the rezoned and Lagrangian grids are sufficiently close to each other, it is possible to use a local procedure on the remapping stage, meaning that mass, energy and momentum are exchanged only between neighboring cells. Local rezoning is conceptually simpler and computationally less expensive than global rezoning. For some of the tests presented in Section 7, we will use the ALE code in the Eulerian framework, so that the rezoned mesh will always coincide with the initial mesh (see e.g., [25]).

1.4. Summary of the new remapping algorithm

To guarantee conservation in the overall ALE simulation, the remapping phase must conservatively interpolate the Lagrange solution onto the rezoned grid. The main purpose of this paper is to describe a new algorithm for remapping on a general, polygonal, staggered grid, including treatment of the density defined in the subcells. Readers interested in the history of remapping methods on staggered meshes are referred to [4,5,26,25,19,21,1,13,23].

To the best of our knowledge, there is no existing remapping method that addresses all of our requirements—remapping on a general polygonal staggered mesh with subcell densities.

We have designed a new remapping strategy consisting of the three following stages:

- First: *Gathering stage*. We define momentum, internal energy, and kinetic energy in the subcells. Recall that the mass of subcell is already defined by (1.2). Mass, momentum, internal energy and kinetic energy in the subcells are defined in such a way that the corresponding total quantities (defined as the sums over subcells) are the same as those at the end of the Lagrangian phase, ensuring that the gathering stage is conservative.
- Second: *Subcell remapping stage*. We use the algorithm described in [18] to remap mass, momentum, internal, and kinetic energy from the subcells of the Lagrangian mesh to the subcells of the new rezoned mesh. This algorithm is linearity-preserving and computationally efficient. It consists of a piecewise linear reconstruction and an approximate integration based on the notion of swept regions. The algorithm does not require finding the intersections of the Lagrangian mesh with the rezoned mesh, which contributes to its efficiency. The algorithm is conservative: total mass, momentum, internal and kinetic energy over subcells of the rezoned mesh are the same as mass, momentum, internal and kinetic energy over subcells of Lagrangian mesh. The total energy is also conserved, being the sum of (individually conserved) internal and kinetic energies. We suggest that remapping internal and kinetic energy separately is more accurate than remapping total energy, because we are not combining two quantities that can have very different magnitudes and behavior.
- Third: *Scattering stage*. We recover the primary variables—subcell density, nodal velocity, and cell-centered specific internal energy—on the new rezoned mesh.
  - Subcell density is recovered by using the remapped mass and volume of the subcell of the rezoned mesh in Eq. (1.4). The subcell masses and the corresponding densities are then adjusted using a conservative repair procedure [18,29,21] to enforce local bounds, which may be violated during the subcell remapping stage. This produces the final subcell density and the corresponding subcell mass that will be used in next time step. The new nodal masses and the cell-centered masses are defined using Eq. (1.2).
  - Next, we define the remapped nodal momenta using the remapped subcell momenta, in such a way that total momenta is conserved (see details in Sections 2 and 3). New velocity components are defined according to (1.7). Then nodal velocity is repaired, resulting in the final velocity that will be used to move the point during the Lagrangian phase in the next computational cycle.
  - To enforce the conservation of total energy, the discrepancy between the remapped kinetic energy in the cell and the kinetic energy that is computed from the remapped subcell masses and the final nodal velocities is contributed to the remapped internal energy in the cell. The new internal energy is recovered using (1.13). Finally, the internal energy and the corresponding specific internal energy are conservatively repaired.

The outline of the rest of this paper is as follows. In Section 2 we will give a precise statement of our goals for remapping on the staggered mesh and will list the desired properties of the remapping algorithm. In Section 3 we will define momentum, internal, and kinetic energy in the subcells of the Lagrangian mesh (gathering stage). The properties of the remapping of subcell quantities from the Lagrangian mesh to the rezoned mesh are briefly described in Section 4 (subcell remapping stage). The definition of the subcell density, nodal velocity and cell-centered specific internal energy on the rezoned mesh (scattering stage) is described in Section 5 and in Appendix A. In Section 6, we prove that our new remapping algorithm is conservative, reversible, and that the DeBar consistency condition for remapping of velocity [4] is satisfied. Numerical results that demonstrate the accuracy and convergence of the remapping algorithm are presented in Section 7. Finally, we conclude the paper in Section 8.

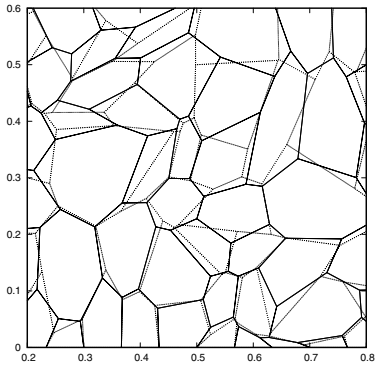


Fig. 2. Fragment of the Lagrangian (dotted lines) and the rezoned (solid lines) grids.

The remapping algorithm must satisfy the following requirements:

- *Conservation*. The total mass, momenta and energy of the new mesh must be the same as that of the old mesh

$$\tilde{M} = M, \ \tilde{\mu}_u = \mu_u, \quad \tilde{\mu}_v = \mu_v, \quad \tilde{E} = E.$$

This property, combined with the same conservation properties of the Lagrangian phase, guarantees the conservation of the overall ALE method.
- *Bound-preservation*. The remapped density, velocity components and internal energy have to be contained within physically justified bounds, which are determined from the corresponding fields in the Lagrangian solution. For example, density and internal energy have to be positive. Moreover, because we assume that the new mesh is obtained from a small displacement of the old mesh, one can require that the new value lie between bounds determined by the values of its neighbors on the old mesh, [18].
- *Accuracy*. It is straightforward to define accuracy in the remap of density; we will require that the remap of density is linearity-preserving. That is, if the values on the old mesh are obtained from a global linear function, then the values on the new mesh have to coincide with the values of the same linear function on the new mesh. For the remap of velocity, there are several different notions related to accuracy. For example, one widely used test of consistency is the so-called DeBar condition (see for example [4]) which can be stated as follows: if a body has a uniform velocity and spatially varying density, then the remapping process should exactly reproduce a uniform velocity. For internal energy, the situation is more complicated. We will demonstrate the accuracy of our new algorithm through the practical expedient of well-chosen test problems.

2. Statement of the remapping

As a result of the Lagrangian phase of a computational cycle, we have a mesh consisting of cells $\{c\}$, and nodes $\{n\}$. We will call this the Lagrangian or old mesh. We have values of density, $\rho(cn)$ in subcells, values of specific internal energy, $\varepsilon(c)$, in cells, and values of the components of velocity, $u(n)$, $v(n)$, at the nodes of the old mesh. As a result of the rezone phase, we have the rezoned or new mesh consisting of cells $\{\tilde{c}\}$, and nodes $\{\tilde{n}\}$. An example of old and new meshes is given in Fig. 2. The goal of the remapping phase is to find an accurate approximation to $\rho(\widetilde{cn})$, $\varepsilon(\tilde{c})$, $u(\tilde{n}), v(\tilde{n})$ on the new mesh. Using the primary variables we can define the total mass M, the momentum vector (μ_u, μ_v), the internal energy $\mathscr{E}$, the kinetic energy K, and the total energy E on the old mesh from Eqs. (1.5), (1.10), (1.17) and (1.18), respectively.

- *Reversibility*. If the new and old meshes are identical, then the remapped primary variables should show no change. This property is closely related to the notion of being free of inversion error, see [4], where it is stated that if the new and old grids coincide, then the remapped velocity on new mesh should coincide with the velocity on the old mesh.

3. Gathering

In the *gathering stage*, we define mass (which is already known), momentum, internal energy, and kinetic energy in the subcells: $m(cn)$, $\mu(cn)$, $v(cn)$, $\mathscr{E}(cn), K(cn)$ such that the corresponding total quantities maintain the same values as they have at the end of the Lagrangian phase

$$\begin{aligned} &M^s \stackrel{\text{def}}{=} \sum_{cn} m(cn) = M, \\ &\mu_u^s \stackrel{\text{def}}{=} \sum_{cn} \mu(cn) = \mu_u, \quad \mu_v^s \stackrel{\text{def}}{=} \sum_{cn} v(cn) = \mu_v, \\ &\mathscr{E}^s \stackrel{\text{def}}{=} \sum_{cn} \mathscr{E}(cn) = \mathscr{E}, \quad K^s \stackrel{\text{def}}{=} \sum_{cn} K(cn) = K. \end{aligned} \tag{3.1}$$

Here the superscript s emphasizes that the corresponding total quantities are defined by summation over subcells. Clearly, if we conserve the total kinetic and the total internal energy, then the total energy

$$E^s \stackrel{\text{def}}{=} \mathscr{E}^s + K^s$$

is also conserved, i.e.,

$$E^s = E. \tag{3.2}$$

As follows from Eqs. (1.10) and (1.17), the total momenta, kinetic energy and internal energy can be expressed by summation of the corresponding cell-centered quantities given by (1.9), (1.12) and (1.15). This suggests the following *design principle*: construct the subcell quantities in such a way that conservation is ensured on cell-by-cell basis. For example, the momentum components $\mu(cn)$ satisfy the following equation:

$$\sum_{n \in N(c)} \mu(cn) = \mu(c), \tag{3.3}$$

and similarly for the other quantities. Thus all requirements of conservation listed in (3.1) will be satisfied.

We note that there is no unique solution for such a construction, because there is one constraint whereas the number of unknowns is equal to the number of subcells in the given cell. For example, Eq. (1.9) suggests that the simplest way to satisfy (3.3) is to define $\mu(cn)$

$$\mu(cn) = m(cn)u(n). \tag{3.4}$$

However, this will not be accurate enough in general; e.g., in the case of a constant density, it will be exact only for a constant velocity field. In the next section we will describe a more accurate algorithm, which in the case of a constant density will be exact for any linear velocity field.

3.1. Definition of subcell momenta

We will present the procedure for defining the x-component of momentum $\mu(cn)$, noting that the definition of the other component $v(cn)$ is similar. Also, for brevity, we will refer to the velocity component u simply as velocity. The total number of nodes (and hence of subcells) of the cell c is denoted by $|N(c)|$.

We will use N instead of $|N(c)|$ as we will never need to use a local indexing for two different cells at the same time. The nodes of the cell under consideration are enumerated from 1 to N in counter-clockwise order.

The subcell momenta will be defined as

$$\mu(cn) = m(cn)u(cn),$$

where $u(cn)$—yet to be defined—has the meaning of a subcell velocity, see Fig. 3 for illustration.

The subcell velocities $u(cn)$ must be defined such that the total momentum of the cell, defined in (1.9), is conserved, i.e.,

$$\sum_{n\in N(c)} m(cn)u(cn) = \sum_{n\in N(c)} m(cn)u(n) = \mu(c). \tag{3.5}$$

As previously mentioned, a simple solution that satisfies (3.5) is to set $u(cn) = u(n)$. However because $u(cn)$ has the meaning of a velocity in subcell, setting it to the velocity in the corresponding node will not be accurate. Instead we seek a more accurate estimate of $u(cn)$ in the form

$$u(cn) = \frac{u(c) + u(n) + u_{n,n^+} + u_{n^-,n}}{4}. \tag{3.6}$$

Here $u(c)$ is not yet defined, and

$$u_{n^-,n} = \frac{1}{2}(u(n^-) + u(n)), \quad u_{n,n^+} = \frac{1}{2}(u(n) + u(n^+)) \tag{3.7}$$

are the approximations of the velocities at the mid-edge points based on the velocities of the corresponding nodes in the cell c (n^- and n^+ are the previous/next nodes with respect to n in the list of vertices of cell c, see Fig. 4). The velocity $u(c)$ has the meaning of a velocity at the cell center. Eq. (3.6) states that the velocity in the center of subcell is a simple average of velocities in the corners of the subcell. Three of these velocities, $u(n)$, $u_{n,n^+}, u_{n^-,n}$ are known quantities and one, $u(c)$, will be defined by conserving the momentum of the cell.

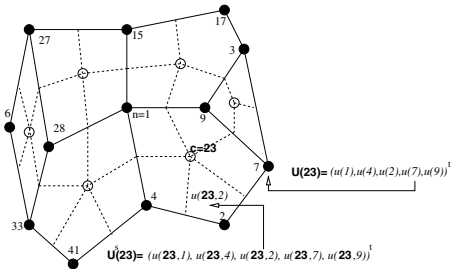


Fig. 3. Vectors $\mathbf{U}(\mathbf{c})$ and $\mathbf{U}^s(c)$ for a given mesh. The numbers are the global indexes of the nodes. $\mathbf{U}(\mathbf{23}) = (u(1), u(4), u(2), u(7), u(9))^t$ is the vector of nodal velocities, for example $u(7)$ is the velocity of node number 7. $\mathbf{U}^s(\mathbf{23}) = (u(\mathbf{23},1), u(\mathbf{23},4), u(\mathbf{23},2), u(\mathbf{23},7), u(\mathbf{23},9))^t$ is the vector of subcell velocities, for example $u(\mathbf{23},2)$ is the velocity of the subcell uniquely defined by cell **23** and node number 2. The local neighbor nodes of $n = 1$ in cell **23** are $n^- = 9$ and $n^+ = 4$, the mid-edge points being called n, $n^- = 1,9$ and $n,n^+ = 1,4$.

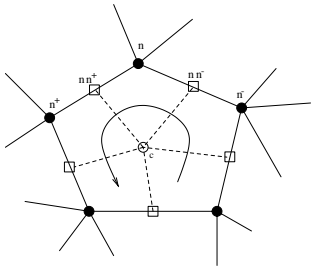


Fig. 4. Neighbor nodes of node n along the boundary of the cell c (using counterclockwise ordering) are n^- and n^+. The mid-edge points are denoted as nn^- and nn^+.

From (3.5) and (3.6) we obtain the following equation for $u(c)$:

$$\sum_{n\in N(c)} m(cn)\left(\frac{u(c) + u(n) + u_{n,n^+} + u_{n^-,n}}{4}\right) = \sum_{n\in N(c)} m(cn)u(n),$$

which is one equation with one unknown. Using the definitions of u_{n,n^+} and $u_{n^-,n}$ from (3.7), we derive a formula for $u(c)$:

$$u(c) = \frac{1}{m(c)} \sum_{n\in N(c)} m(cn)\left(2u(n) - \frac{1}{2}u(n^+) - \frac{1}{2}u(n^-)\right). \tag{3.8}$$

An equivalent form of this definition is

$$u(c) = \frac{1}{m(c)} \sum_{n\in N(c)} m(cn)u(n) - \frac{1}{m(c)} \sum_{n\in N(c)} m(cn)\frac{u(n^+) - 2u(n) + u(n^-)}{2}. \tag{3.9}$$

From this equation it is clear that if $u(n) = \mathscr{C}$ then $u(c) = \mathscr{C}$; therefore in the case of constant velocity with any arbitrary distribution of masses, our definition is exact. It is also easy to verify that if all the subcell masses are the same, then

$$u(c) = \frac{1}{|N(c)|} \sum_{n\in N(c)} u(n),$$

meaning that this formula is exact for a linear velocity field.

Substituting (3.8) into (3.6) yields

$$\begin{aligned} u(cn) &= \frac{1}{4}\left(2u(n) + \frac{u(n^+)}{2} + \frac{u(n^-)}{2}\right) + \left[\frac{1}{4}\frac{1}{m(c)} \sum_{k\in N(c)} m(ck)\left(2u(k) - \frac{1}{2}u(k^+) - \frac{1}{2}u(k^-)\right)\right] \\ &= \frac{1}{4}\left(2u(n) + \frac{u(n^+)}{2} + \frac{u(n^-)}{2}\right) + \sum_{k\in N(c)} \frac{m(ck)}{8m(c)}(4u(k) - u(k^+) - u(k^-)). \end{aligned} \tag{3.10}$$

The last term in (3.10) can be transformed as follows. First we split this term in three separate sums

$$\sum_{k\in N(c)} \frac{m(ck)}{8m(c)}(4u(k) - u(k^+) - u(k^-)) = \sum_{k\in N(c)} \frac{m(ck)}{8m(c)}4u(k) - \sum_{k\in N(c)} \frac{m(ck)}{8m(c)}u(k^+) - \sum_{k\in N(c)} \frac{m(ck)}{8m(c)}u(k^-).$$

Now by shifting the index in second and third sum and combining the resulting expressions, we get

$$\sum_{k\in N(c)} \frac{m(ck)}{8m(c)}4u(k) - \sum_{k\in N(c)} \frac{m(ck^-)}{8m(c)}u(k) - \sum_{k\in N(c)} \frac{m(ck^+)}{8m(c)}u(k) = \sum_{k\in N(c)} u(k)\left(\frac{-m(ck^-) + 4m(ck) - m(ck^+)}{8m(c)}\right). \tag{3.11}$$

Finally, using (3.11) and (3.10) we get

$$u(cn) = \frac{1}{4}\left(2u(n) + \frac{u(n^+)}{2} + \frac{u(n^-)}{2}\right) + \left[\sum_{k\in N(c)} u(k)\left(\frac{-m(ck^-) + 4m(ck) - m(ck^+)}{8m(c)}\right)\right]. \tag{3.12}$$

(In Appendix A we present a 1D analog of the derivation of this formula for the subcell velocity.)

Eq. (3.12) defines the subcell velocities $u(cn)$ in terms of the nodal velocities $u(n)$. As a result of Eq. (3.8), and the previously discussed properties of $u(c)$, Eq. (3.12) is exact in the following cases:

- A constant velocity and an arbitrary mass distribution (this property will be used later to prove the DeBar condition).
- An equal subcell mass distribution and a linear velocity.

Let us rewrite (3.12) for all $n \in N(c)$ in matrix form. To do this, we represent the velocities $u(n)$ of the vertices of one particular cell c as the elements of a vector $\mathbf{U}(c)$

$$\mathbf{U}(c) = \{u(n), \quad n \in N(c)\}^t. \tag{3.13}$$

Similarly, we represent the subcell velocities as the elements of a vector $\mathbf{U}^s(c)$

$$\mathbf{U}^s(c) = \{u(cn), \quad n \in N(c)\}^t.$$

A graphical illustration of these definitions of $\mathbf{U}(c)$, $\mathbf{U}^s(c)$ is shown in Fig. 3. As a matter of notation, we will use bold letters to denote column vectors while matrices will be denoted with a capital bold over-lined letter as $\bar{\mathbf{I}}_c$.

Now (3.12) can be rewritten in matrix form as follows:

$$\mathbf{U}^s(c) = \bar{\mathbf{I}}_c\mathbf{U}(c), \tag{3.14}$$

where the matrix $\bar{\mathbf{I}}_c$ is

$$\bar{\mathbf{I}}_c = \frac{1}{4}\cdot\begin{pmatrix} 2 & \frac{1}{2} & 0 & 0 & \cdots & 0 & \frac{1}{2} \\ \frac{1}{2} & 2 & \frac{1}{2} & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{2} & 2 & \frac{1}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{2} & 0 & 0 & 0 & \cdots & \frac{1}{2} & 2 \end{pmatrix} + \begin{pmatrix} Q_{N,1,2} & Q_{1,2,3} & Q_{2,3,4} & \cdots & Q_{N-1,N,1} \\ Q_{N,1,2} & Q_{1,2,3} & Q_{2,3,4} & \cdots & Q_{N-1,N,1} \\ Q_{N,1,2} & Q_{1,2,3} & Q_{2,3,4} & \cdots & Q_{N-1,N,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Q_{N,1,2} & Q_{1,2,3} & Q_{2,3,4} & \cdots & Q_{N-1,N,1} \end{pmatrix}. \tag{3.15}$$

Each element of the second matrix in (3.15) is formed from three consecutive subcell masses, and in terms of the global indexing

$$Q_{n^-,n,n^+} = \frac{-m(cn^-) + 4m(cn) - m(cn^+)}{8m(c)}.$$

In Appendix A we prove by construction that the matrix $\bar{\mathbf{I}}_c$ is always invertible, meaning that Eq. (3.14) prescribes a one-to-one correspondence between the nodal and the subcell velocities in a given cell. As mentioned earlier, the fact that Eq. (3.12) is exact when $u(n) = \mathscr{C}$ implies the following relations for the matrix $\bar{\mathbf{I}}_c$:

$$\bar{\mathbf{I}}_c\mathbf{C} = \mathbf{C}, \quad (\bar{\mathbf{I}}_c)^{-1}\mathbf{C} = \mathbf{C}, \tag{3.16}$$

where $\mathbf{C}$ is constant vector of length N, each of whose components equals $\mathscr{C}$. Finally, by construction we ensure that momentum is conserved in each cell, and therefore is conserved for the entire domain.

3.2. Definition of subcell kinetic energy

The subcell specific kinetic energy is denoted $k(cn)$, and we will require that the total kinetic energy in the cell c is conserved

$$\sum_{n\in N(c)} m(cn)k(cn) = \sum_{n\in N(c)} m(cn)\frac{|\mathbf{u}(n)|^2}{2} = K(c). \tag{3.17}$$

If we represent the specific kinetic energies at the vertices and in the subcells of cell c as components of vectors

$$\mathbf{k}(c) = \left\{k(n) = \frac{|\mathbf{u}(n)|^2}{2}, n \in N(c)\right\}^t, \quad \mathbf{k}^s(c) = \{k(cn), \ n \in N(c)\}^t, \tag{3.18}$$

then by definition

$$\mathbf{k}^s(c) \overset{\text{def}}{=} \bar{\mathbf{I}}_c\mathbf{k}(c). \tag{3.19}$$

and the subcell kinetic energy is finally given by

$$K(cn) = m(cn)k(cn). \tag{3.20}$$

By construction the kinetic energy in each cell is preserved, ensuring that the kinetic energy of the entire domain is preserved as well. We emphasize that

$$k(cn) \neq \frac{|\mathbf{u}(cn)|^2}{2},$$

because $k(cn)$ and $\mathbf{u}(cn)$ are defined independently. Moreover we cannot set $k(cn) = \frac{|\mathbf{u}(cn)|^2}{2}$ because this definition would not conserve the kinetic energy in the cell.

3.3. Definition of subcell internal energy

The specific internal energy ε is a cell-centered quantity. Thus the construction we developed for the subcell velocity cannot be applied. Instead, we will define the subcell internal energy in the following two steps:

(1) We will prescribe a linear reconstruction of the internal energy per unit volume in the cell c, which is denoted $(\rho\varepsilon)_c(x,y)$. This reconstruction:
- must be conservative in the cell, that is,

$$\int_c (\rho\varepsilon)_c(x,y)\,\mathrm{d}x\,\mathrm{d}y = \mathscr{E}(c) = \rho(c)\varepsilon(c)V(c) = m(c)\varepsilon(c). \tag{3.21}$$

- must be exact when $\rho(x,y)\varepsilon(x,y)$ is a linear function.

(2) We will compute the subcell internal energy by integrating $(\rho\varepsilon)_c(x,y)$ over the corresponding subcell.

In each cell c, the function $(\rho\varepsilon)_c(x,y)$ has the form

$$(\rho\varepsilon)_c(x,y) = \rho(c)\varepsilon(c) + \delta^c_x(x - x_c) + \delta^c_y(y - y_c), \tag{3.22}$$

with (x_c, y_c) being the centroid of the cell

$$x_c = \frac{1}{V(c)}\int_c x\,\mathrm{d}x\,\mathrm{d}y, \quad y_c = \frac{1}{V(c)}\int_c y\,\mathrm{d}x\,\mathrm{d}y.$$

We define the slopes δ^c_x, δ^c_y, by the Barth–Jespersen (BJ) algorithm [2]. The BJ algorithm is an algorithm for piecewise linear reconstruction of a function $f(x,y)$ given by its means $\bar{f}_c = \frac{1}{V(c)}\int_c f(x,y)\,\mathrm{d}x\,\mathrm{d}y$ over mesh cells. In cell c, function $f(x,y)$ is represented by the linear function $f_c(x,y)$. The BJ algorithm has the following properties:

- The mean of $f_c(x,y)$ over cell c is equal to the given mean value $\bar{f}_c$, that is
$$\frac{1}{V(c)}\int_c f_c(x,y)\,\mathrm{d}x\,\mathrm{d}y = \bar{f}_c.$$
- It is exact if $f(x,y)$ is a global linear function, $f(x,y) = a + bx + cy$.
- In each cell c, the linear function $f_c(x,y)$ is constructed in a such a way that its values at the cell vertices are within the bounds defined by the maximum and the minimum of the mean values over the set $C(c)$, consisting of cell c itself and its nearest neighbors. That is,
$$\min_{k\in C(c)} \bar{f}_k \leqslant f_c(x_n, y_n) \leqslant \max_{k\in C(c)} \bar{f}_k, \quad n \in N(c).$$

Details of the BJ algorithm can be found in [2] and in Appendix A of [22].

It is easy to verify that this reconstruction (3.22) is conservative because

$$\begin{aligned}\int_c (\rho\varepsilon)(x,y)\,\mathrm{d}x\,\mathrm{d}y &= \int_c \Big(\rho(c)\varepsilon(c) + \delta^c_x(x - x_c) + \delta^c_y(y - y_c)\Big)\,\mathrm{d}x\,\mathrm{d}y \\ &= V(c)\rho(c)\varepsilon(c) + \delta^c_x\underbrace{\left\{\int_c x\,\mathrm{d}x\,\mathrm{d}y - V(c)x_c\right\}}_{=0} + \delta^c_y\underbrace{\left\{\int_c y\,\mathrm{d}x\,\mathrm{d}y - V(c)y_c\right\}}_{=0} = m(c)\varepsilon(c) \\ &= \mathscr{E}(c),\end{aligned}$$

where the expressions in curly brackets are zero because of the definition of the centroid.

The subcell internal energy $\mathscr{E}(cn)$ is defined as the integral of $(\rho\varepsilon)_c(x,y)$ over the subcell cn (step two of the algorithm)

$$\mathscr{E}(cn) = \int_{cn} (\rho\varepsilon)_c(x,y)\,\mathrm{d}x\,\mathrm{d}y. \tag{3.23}$$

The internal energy over each cell is conserved because

$$\mathscr{E}(c) = \int_c (\rho\varepsilon)_c(x,y)\,\mathrm{d}x\,\mathrm{d}y = \sum_{n\in N(c)}\left(\int_{cn}(\rho\varepsilon)_c(x,y)\,\mathrm{d}x\,\mathrm{d}y\right) = \sum_{n\in N(c)}\mathscr{E}(cn),$$

and in consequence, the internal energy is conserved over the entire domain.

4. Subcell remapping

For the *subcell remapping stage*, we employ the algorithm described in [18] to remap mass, momentum, internal, and kinetic energy from the subcells of the Lagrangian mesh to the subcells of the new rezoned mesh. This algorithm produces values for the mass, momentum, internal energy, and kinetic energy in the each of the subcells of the new mesh: $m(\widetilde{cn}), \mu(\widetilde{cn}), v(\widetilde{cn}), \mathscr{E}(\widetilde{cn}), K(\widetilde{cn})$. It is conservative, i.e.,

$$\begin{aligned}&\tilde{M}^s \overset{\text{def}}{=} \sum_{\widetilde{cn}} m(\widetilde{cn}) = M^s, \\ &\tilde{\mu}^s_u \overset{\text{def}}{=} \sum_{\widetilde{cn}} \mu(\widetilde{cn}) = \mu^s_u, \quad \tilde{\mu}^s_v \overset{\text{def}}{=} \sum_{\widetilde{cn}} v(\widetilde{cn}) = \mu^s_v, \\ &\tilde{\mathscr{E}}^s \overset{\text{def}}{=} \sum_{\widetilde{cn}} \mathscr{E}(\widetilde{cn}) = \tilde{\mathscr{E}}^s, \quad \tilde{K}^s \overset{\text{def}}{=} \sum_{\widetilde{cn}} K(\widetilde{cn}) = \tilde{K}^s,\end{aligned} \tag{4.1}$$

and linearity-preserving. Clearly, if the total kinetic and the total internal energy are conserved, then the total energy

$$\tilde{E}^s \overset{\text{def}}{=} \tilde{\mathscr{E}}^s + \tilde{K}^s$$

is also conserved

$$\tilde{E}^s = E^s. \tag{4.2}$$

For future analysis we note that when the new mesh coincides with the old mesh, then the subcell remapping process does not change the subcell quantities. We want to emphasize that in this stage one could use any other accurate conservative remapping algorithm for cell-centered (cells being the subcells in this context) quantities.

5. Scattering

The third element of our algorithm is the *scattering stage*, in which we recover the primary variables—i.e., subcell density, $\rho(\widetilde{cn})$, nodal velocity, $u(\tilde{n}), v(\tilde{n})$, and cell-centered specific internal energy $\varepsilon(\tilde{c})$ – on the new mesh. At the beginning of the scattering stage, we have the following subcell quantities on the new mesh: mass $m(\widetilde{cn})$, momenta $\mu(\widetilde{cn}), v(\widetilde{cn})$, internal energy $\mathscr{E}(\widetilde{cn})$ and kinetic energy $K(\widetilde{cn})$.

The scattering stage has to maintain conservation, meaning that the primary variables on new mesh must satisfy the following conditions:

$$\tilde{M} = \sum_{\widetilde{cn}} m(\widetilde{cn}) = \sum_{\widetilde{cn}} \rho(\widetilde{cn})V(\widetilde{cn}) = \tilde{M}^s, \tag{5.1}$$

$$\tilde{\mu}_u = \sum_{\tilde{n}} m(\tilde{n})u(\tilde{n}) = \tilde{\mu}^s_u, \quad \tilde{\mu}_v = \sum_{\tilde{n}} m(\tilde{n})v(\tilde{n}) = \tilde{\mu}^s_v, \tag{5.2}$$

$$\tilde{E} = \sum_{\tilde{c}}\left(\mathscr{E}(\tilde{c}) + K(\tilde{c})\right) = \sum_{\tilde{c}}\left[m(\tilde{c})\varepsilon(\tilde{c}) + \sum_{\tilde{n}\in N(\tilde{c})} m(\widetilde{cn})\frac{|\mathbf{u}(\tilde{n})|^2}{2}\right] = \tilde{\mathscr{E}}^s + \tilde{K}^s = \tilde{E}^s, \tag{5.3}$$

where

$$m(\tilde{c}) = \sum_{\tilde{n}\in N(\tilde{c})} m(\widetilde{cn}), \quad m(\tilde{n}) = \sum_{\tilde{c}\in C(\tilde{n})} m(\widetilde{cn}). \tag{5.4}$$

5.1. Definition of subcell density

The subcell density is recovered using Eq. (1.4),

$$\rho(\widetilde{cn}) = m(\widetilde{cn})/V(\widetilde{cn}).$$

The subcell masses and densities are then corrected using a conservative repair procedure [18] to reinforce local bounds that may have been violated during the subcell remapping stage. Because we assume that the rezoned grid is close to the Lagrangian grid, we choose the bounds for $\rho(\widetilde{cn})$ as the minimal and maximal values of the subcell densities in the neighboring old subcells (i.e., before remapping). We will continue to use the same notation $\rho(\widetilde{cn}), m(\widetilde{cn})$ for the repaired quantities, and will employ the same convention for other remapped and repaired quantities later in the paper.

5.2. Definition of nodal velocity

First, we define the new subcell velocity from subcell momenta and masses

$$u(\widetilde{cn}) \overset{\text{def}}{=} \frac{\mu(\widetilde{cn})}{m(\widetilde{cn})}. \tag{5.5}$$

Next we define the nodal velocities, $u^{\tilde{c}}(\tilde{n})$, for $\tilde{n} \in N(\tilde{c})$ with respect to cell $\tilde{c}$, by inverting Eq. (3.14), applied to the new mesh

$$\mathbf{U}(\tilde{c}) = (\tilde{\mathbf{I}}_{\tilde{c}})^{-1}\mathbf{U}^{\mathbf{s}}(\tilde{c}), \tag{5.6}$$

with the formal vector notation

$$\mathbf{U}(\tilde{c}) = \left\{u^{\tilde{c}}(\tilde{n}),\ \tilde{n} \in N(\tilde{c})\right\}^t, \quad \mathbf{U}^{\mathbf{s}}(\tilde{c}) = \left\{u(\widetilde{cn}),\ \tilde{n} \in N(\tilde{c})\right\}^t.$$

We have introduced a new notation, $u^{\tilde{c}}(\tilde{n})$, because in general, Eq. (5.6) will give different results for the same node $\tilde{n}$ for different cells $\tilde{c}$. Note that the matrix $\tilde{\mathbf{I}}_{\tilde{c}}$ is constructed using the final subcell masses of the new mesh.

Finally, a unique nodal velocity at the node of the new cell can be defined

$$u(\tilde{n}) = \frac{1}{m(\tilde{n})}\sum_{\tilde{c}\in C(\tilde{n})} m(\widetilde{cn})u^{\tilde{c}}(\tilde{n}). \tag{5.7}$$

It is easy to show that momentum is conserved, i.e., $\tilde{\mu}_u = \tilde{\mu}^s_u$. In fact,

$$\tilde{\mu}_u \overset{\text{def}}{=} \sum_{\tilde{n}} m(\tilde{n})u(\tilde{n}), \tag{5.8}$$

and the definition of $u^{\tilde{c}}(\tilde{n})$ in (5.7), gives

$$\sum_{\tilde{n}} m(\tilde{n})u(\tilde{n}) = \sum_{\tilde{n}}\sum_{\tilde{c}\in C(\tilde{n})} m(\widetilde{cn})u^{\tilde{c}}(\tilde{n}). \tag{5.9}$$

By changing the order of summation in right-hand-side of the previous equation we derive

$$\sum_{\tilde{n}}\sum_{\tilde{c}\in C(\tilde{n})} m(\widetilde{cn})u^{\tilde{c}}(\tilde{n}) = \sum_{\tilde{c}}\sum_{\tilde{n}\in N(\tilde{c})} m(\widetilde{cn})u^{\tilde{c}}(\tilde{n}). \tag{5.10}$$

From the definition of $u^{\tilde{c}}(n)$ we have

$$\sum_{\tilde{n}\in N(\tilde{c})} u^{\tilde{c}}(\tilde{n})m(\widetilde{cn}) = \sum_{\tilde{n}\in N(\tilde{c})} \mu(\widetilde{cn}) = \tilde{\mu}^s_u. \tag{5.11}$$

From (5.8)–(5.11) we conclude that momentum is conserved

$$\tilde{\mu}_u = \tilde{\mu}^s_u. \tag{5.12}$$

In the final step, velocity is repaired with respect to bounds chosen as the maximal and minimal values of $u(n)$ (i.e., values before remapping) over the following stencil (see Fig. 5):

$$n \in \cup_{c\in C(n)}\{k \in N(c)\}.$$

After the repair stage, we obtain the final velocity at the nodes of the new mesh. This velocity will be used in the Lagrangian phase in the next time step. The definition of the new nodal velocity by Eq. (5.7) introduces dissipation, that is, the kinetic energy decreases. The repair process by itself conserves the total momenta, but also can change the kinetic energy. The overall change in kinetic energy will be accounted for in the definition of cell-centered specific internal energy. The final kinetic energy in the new cell is given by

$$K(\tilde{c}) = \sum_{\tilde{n}\in N(\tilde{c})} m(\widetilde{cn})\frac{|\mathbf{u}(\tilde{n})|^2}{2}. \tag{5.13}$$

5.3. Definition of cell-centered specific internal energy

The final specific internal energy has to be defined to ensure the conservation of total energy. At this stage of the scattering, we know the following quantities for each cell:

- the final kinetic energy $K(\tilde{c})$ in the cell evaluated from Eq. (5.13), in which the final velocities and the final subcell masses are used;
- the remapped subcell internal energy $\mathscr{E}(\widetilde{cn})$, and the remapped subcell kinetic energy, $K(\widetilde{cn})$.

By definition, the total energy in the cell is

$$E(\tilde{c}) = \mathscr{E}(\tilde{c}) + K(\tilde{c}), \tag{5.14}$$

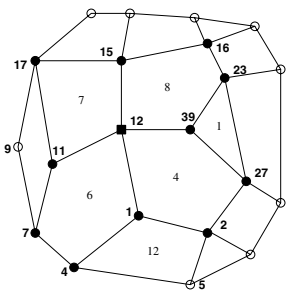


Fig. 5. The stencil for velocity repair. The stencil for node **12** (marked by solid square) consists of the union of the vertices of cells 6, 4, 8, 7, that is, **1, 2, 4, 7, 11, 12, 15, 16, 17, 23, 27, 39**, which are marked by solid circles.

where $\mathscr{E}(\tilde{c})$ is still unknown. If we define

$$E(\tilde{c}) = \sum_{\tilde{n}\in N(\tilde{c})} (\mathscr{E}(\widetilde{cn}) + K(\widetilde{cn})), \tag{5.15}$$

then the conservation of total energy is guaranteed because $\mathscr{E}(\widetilde{cn})$ and $K(\widetilde{cn})$ are obtained as a result of a conservative subcell remapping.

From Eqs. (5.15) and (5.14), we conclude that to conserve total energy, the new internal energy in the cell must be defined as follows:

$$\mathscr{E}(\tilde{c}) = \sum_{\tilde{n}\in N(\tilde{c})} \mathscr{E}(\widetilde{cn}) + \left[\left(\sum_{\tilde{n}\in N(\tilde{c})} K(\widetilde{cn})\right) - K(\tilde{c})\right]. \tag{5.16}$$

The term in the square brackets can be interpreted as the distribution of the change in kinetic energy due to the processes of defining and repairing the nodal velocities. The new specific internal energy is defined by analogy with (1.13) as

$$\varepsilon(\tilde{c}) = \mathscr{E}(\tilde{c})/m(\tilde{c}), \tag{5.17}$$

and in the final step, the specific internal energy is conservatively repaired.

6. Properties of the algorithm

6.1. Conservation

As we have proved in previous sections, mass, momentum, and total energy are all conserved at each stage: gathering, subcell remapping, and scattering. Therefore,

$$\begin{aligned}
&\widetilde{M} = \widetilde{M}^s = M^s = M,\\
&\tilde{\mu}_u = \tilde{\mu}_u^s = \mu_u^s = \mu_u,\\
&\tilde{\mu}_v = \tilde{\mu}_v^{\,s} = \mu_v^s = \mu_v,\\
&\tilde{E} = \tilde{E}^s = E^s = E.
\end{aligned}$$

That is, mass, momenta and total energy are conserved by the overall process.

6.2. Reversibility

Reversibility of the remapping means that if the new and old meshes are identical, then the primary variables will not be changed. Reversibility is a very important property that is related to the continuous dependence of the change of primary variables between the old and the new meshes. As mentioned in Section 4, there is no change in the subcell quantities during the subcell remapping stage if the new and old meshes are identical, i.e.,

$$m(\widetilde{cn}) = m(cn), \tag{6.1}$$

$$\mu(\widetilde{cn}) = \mu(cn), \quad v(\widetilde{cn}) = v(cn), \tag{6.2}$$

$$\mathscr{E}(\widetilde{cn}) = \mathscr{E}(cn), \quad K(\tilde{cn}) = K(cn). \tag{6.3}$$

From (6.1) we can immediately conclude that the subcell density has not been changed

$$\rho(\widetilde{cn}) = \frac{m(\widetilde{cn})}{V(\widetilde{cn})} = \frac{m(cn)}{V(cn)} = \rho(cn).$$

Combining (6.1) and (6.2) with the definition of $u(\widetilde{cn})$, we see that $u(\widetilde{cn}) = u(cn)$, and therefore

$$\mathbf{U}^s(\tilde{c}) = \mathbf{U}^s(c). \tag{6.4}$$

Also, (6.1), implies the matrix equality

$$\bar{\mathbf{I}}_{\tilde{c}} = \bar{\mathbf{I}}_c. \tag{6.5}$$

We recall that by definition

$$\mathbf{U}^s(c) = \bar{\mathbf{I}}_c \mathbf{U}(c). \tag{6.6}$$

Thus, combining (5.6), (6.5), (6.4) and (6.6) we derive

$$\mathbf{U}(\tilde{c}) = (\bar{\mathbf{I}}_{\tilde{c}})^{-1}\mathbf{U}^s(\tilde{c}) = (\bar{\mathbf{I}}_c)^{-1}\mathbf{U}^s(c) = (\bar{\mathbf{I}}_c)^{-1}\cdot(\bar{\mathbf{I}}_c)\mathbf{U}(c) = \mathbf{U}(c). \tag{6.7}$$

Eq. (6.7) means that

$$u^{\tilde{c}}(\tilde{n}) = u(n), \tag{6.8}$$

demonstrating that the old and new nodal velocities in node $\tilde{n}$ (from the point of view of all cells sharing this node) are the same. Now combining (5.7), (6.1) and (6.8), with the definition of $m(n)$ in (1.2), we derive

$$u(\tilde{n}) = \frac{1}{m(\tilde{n})}\sum_{\tilde{c}\in C(\tilde{n})} m(\widetilde{cn})u^{\tilde{c}}(\tilde{n}) = \frac{1}{m(n)}\sum_{c\in C(n)} m(cn)u(n) = u(n)\left(\frac{1}{m(n)}\sum_{c\in C(n)} m(cn)\right) = u(n), \tag{6.9}$$

demonstrating that the nodal velocity stays the same

$$u(\tilde{n}) = u(n).$$

We next prove that $\varepsilon(\tilde{c}) = \varepsilon(c)$. Because of (5.17), it is sufficient to prove that

$$\mathscr{E}(\tilde{c}) = \mathscr{E}(c).$$

Using (5.16) and the fact that after the subcell remapping stage, the subcell internal and kinetic energies are not changed, we derive

$$\mathscr{E}(\tilde{c}) = \sum_{n\in N(c)} \mathscr{E}(cn) + \left[\left(\sum_{n\in N(c)} K(cn)\right) - K(\tilde{c})\right]. \tag{6.10}$$

By construction

$$\sum_{n\in N(c)} \mathscr{E}(cn) = \mathscr{E}(c), \quad \sum_{n\in N(c)} K(cn) = K(c).$$

Therefore, from (6.10) we can conclude that

$$\mathscr{E}(\tilde{c}) = \mathscr{E}(c) + [K(c) - K(\tilde{c})]. \tag{6.11}$$

Finally, the expression in square brackets in (6.11) is zero, because of the definition of $K(\tilde{c})$ in (5.13), and because $u(\tilde{n}) = u(n)$. Thus we have proved that $\mathscr{E}(\tilde{c}) = \mathscr{E}(c)$, and so

$$\varepsilon(\tilde{c}) = \varepsilon(c).$$

To summarize, we have demonstrated that all of the primary quantities before repair do not change if the new mesh and the old mesh are the same. Further, the repair process does not change anything because all variables are in bounds by definition, if the meshes are identical.

6.3. DeBar consistency condition

As mentioned in Section 3.1, when the nodal velocity on the old mesh is constant $\{u(n) = \mathscr{C}\}$ the subcell velocity on the old mesh is also constant, $\{u(cn) = \mathscr{C}\}$. For a constant subcell velocity $\{u(cn) = \mathscr{C}\}$, it is true by definition that the subcell momentum on the old mesh is $\{\mu(cn) = \mathscr{C}\cdot m(cn)\}$ and the remapped subcell momentum on the new mesh is $\{\mu(\widetilde{cn}) = \mathscr{C}\cdot m(\widetilde{cn})\}$. Now using (5.5) we get $\{u(\widetilde{cn}) = \mathscr{C}\}$. From the property of the matrix $\bar{\mathbf{I}}_{\tilde{c}}$ in (3.16) (which holds for both the old and new meshes) and Eq. (5.6) we conclude that $\{u^{\tilde{c}}(\tilde{n}) = \mathscr{C}\}$. Finally, from Eq. (5.7), and because $\{u^{\tilde{c}}(\tilde{n}) = \mathscr{C}\}$, we derive $u(\tilde{n}) = \mathscr{C}$. Thus we have proved that the DeBar consistency condition is satisfied: if a body has an uniform velocity and a spatially varying density, then the remap procedure exactly reproduces this uniform velocity.

7. Numerical results

In this section we will investigate numerically the performance of our new method. All problem are solved in Cartesian coordinates (x, y).

Our remapping method is unique in the sense that it is intended for a staggered mesh of general polygons using a subcell discretization of the density. As previously mentioned, we are not aware of any other method that can treat such a remapping problem. However, we are still interested in comparing our new remapping method with other known methods for remapping on a staggered mesh. To make such comparisons, we need to identify specific situations where both our new method and other existing methods can be used. One such situation is the case of a 1D staggered discretization, where there are no polygons and there is no hourglass phenomenon. Therefore, in Section 7.1.1, we consider several well-known 1D problems (i.e., where the solution depends only on x): Sod's problem, [30,31]; the blast wave problem of Woodward and Colella, [34,25]; and the LeBlanc shock tube problem, [4,25]. On this set of problems we will compare our new method with three other methods: the Half-Interval-Shift (HIS) method, [3,4]; the Nodal Momentum Remap (NMR) method, [25]; and the Method of Moments (MM), [19,4]. All three methods, HIS,NMR, and MM employ only a cell-centered discretization for density (no subcells) and differ individually in how velocity is remapped.

The 1D HIS method employs a remap of two cell-centered momenta, that are "momenta shifted" from the vertices of the corresponding cell, remapped, and then combined to recover unique velocities at the vertices. The 1D MM employs a cell-centered remap of cell-centered momentum, (1.9), and discrete derivative, $\delta u/\delta x \sim \partial u/\partial x$, and then uses these to recover a unique velocity at the node. The NMR method uses a dual mesh with vertices in the centers of the original cells to directly remap nodal momentum. We will not discuss advantages and disadvantages of these methods, but refer the interested reader to [4]. In our implementations of all these methods, after a unique velocity at each node is recovered, a cell-centered specific internal energy is defined as described in Section 5.3. In all implementations, repair is performed in the same way as described in Section 5. Also, in our 1D implementation of these three methods, the Lagrangian phase is the same for all methods and is consistent with the methodology described in [12] when all subcell densities are equal. In 1D there can be no hourglass deformation, and so no hourglass treatment is necessary. To make the implementation of HIS, MM, and NMR, methods comparable to our method, we also have arranged the order of computations similarly to our method. In all three methods the flux limiter remapper is used with the specific choice of the Barth–Jespersen limiter [2]. We apply all methods in an Eulerian framework, which is described as "Eulerian as Lagrange Plus Remap", [25].

In Section 7.1.2, we will use the same set of 1D test problems to investigate numerically the convergence properties of our new algorithm, always in the Eulerian framework. In Section 7.2, we will use

the well-known 2D Sedov blast wave problem to demonstrate the performance of our new method on both logically rectangular and polygonal meshes. This problem can be run in pure Lagrangian regime as well, and we will use Lagrangian results as a reference. For this problem we will present results for both the Eulerian framework and also for an ALE method using the RJM rezone strategy, [17,28].

7.1. One-dimensional tests

For all 1D test problems we use our new method implemented in a 2D code, but run on an initially square mesh with only two cells in y direction. The length of the computational domain in the y direction, $y_{\max}$, depends on number of cells in x direction (in the x direction initial mesh is always uniform). In our description of the test problems, we will specify only the length of computational domain and the number of cells in the x direction. It is interesting that our numerical experiments produce almost identical results (at the resolution presented) for the MM, the HIS and the NMR methods. For this reason we present only results obtained by NMR method.

7.1.1. Comparison with other methods

7.1.1.1. Sod problem. The Sod problem is a Riemann shock tube with a relatively small discontinuity, and so is very mild test. Its solution consists of a left moving rarefaction, a contact discontinuity and a right moving shock; the exact solution is illustrated in Fig. 6 by the solid line.

In our numerical experiments, the computational domain is $1 \geqslant x \geqslant 0$. The discontinuity is initially at 0.5. The domain is filled with an ideal gas with $\gamma = 1.4$. The density/pressure values on the left side of the discontinuity are 1.0/1.0, while those on the right side are 0.125/0.1. In Fig. 6, we present numerical results for the density at the final time $t = 0.25$ for a run with $N_x = 200$ computational cells. The results obtained by our new method and by the NMR method are very close, but the resolution of the contact discontinuity is slightly better for our method.

7.1.1.2. Woodward–Colella blast wave problem. The computational domain for this problem has length one, with reflecting walls at the both ends. The gas is an ideal gas with $\gamma = 1.4$. At $t = 0.$, the gas is at rest with an uniform density equal to 1.0. The initial pressure is 1000.0 in the leftmost tenth of the domain, 100.0 in the rightmost tenth, and 0.01 everywhere else. The final problem time is $t = 0.038$. Initially, two shocks and two contacts develop at the initial discontinuities and propagate toward one another, while two rarefactions develop, propagate toward the walls, and reflect off them. As time progresses, these six initial waves interact and create additional contact discontinuities. There is no analytical solution for this problem and typically a solution obtained by purely Lagrangian method with very high resolution ($N_x = 3600$ cells in our case) is considered as the reference "truth" (the solid line in Fig. 7). As has been mentioned in [25], the Lagrangian solution has a flaw, a spurious overshoot at $x \approx 0.765$. In Fig. 7 we present numerical results obtained by NMR and our new method for $N_x = 1200$.

A discussion of the results obtained by the NMR method can be found in [25]. It appears that, at least for the density, our method gives better results for this problem. We note however that the difference in the results is accentuated by the use of the Barth–Jespersen limiter. When the minmod limiter is used in both methods, the results are much closer. In presenting these results, we note that in our 2D code, we use the Barth–Jespersen limiter most typically.

7.1.1.3. LeBlanc shock tube problem. In this extreme shock tube problem, the initial discontinuity separates a region of very high energy and density from one of low energy and density. This is a much more severe test than the Sod Problem. The computational domain is $9 \geqslant x \geqslant 0$ and is filled with an ideal gas with $\gamma = 5/3$. The gas is initially at rest. The initial discontinuity is at $x = 0.3$: $(\rho,\varepsilon) = (1, 0.1)$ for $x < 3$ and

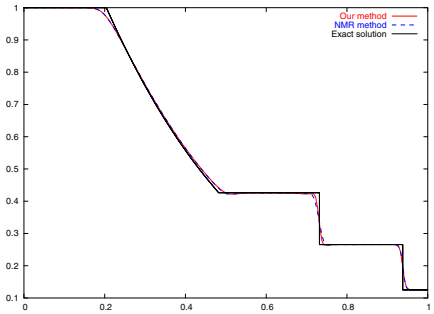


Fig. 6. Sod test problem. Comparison of our new method and the NRM method: density at $t = 0.25$, $N_x = 200$.

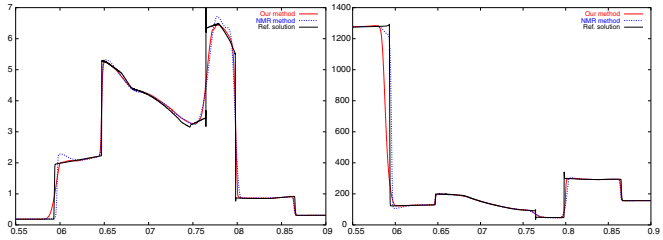


Fig. 7. Woodward–Colella blast wave problem. Comparison of the NMR method and the new method: density (zoom)—left, and specific internal energy (zoom)—right at $t = 0.038$, $N_x = 1200$.

(0.001×10^{-7}) for $x > 3$. The solution consists of a rarefaction moving to the left, and a contact discontinuity and a strong shock moving to the right—solid line in Fig. 8. At the final time of $t = 6.0$, the shock wave is located at $x = 7.975$. In Fig. 8 we present numerical results obtained by NMR and our new method for $N_x = 1400$. In comparison with the NMR method, our new method gives a more accurate position of the contact discontinuity, but shows a larger, relatively narrow, overshoot at the contact. The position of the shock is slightly more accurate for the NMR method.

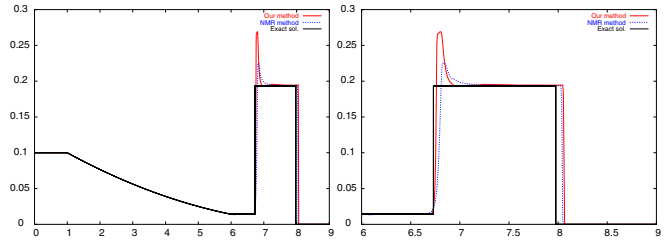


Fig. 8. LeBlanc shock tube problem. Comparison of the NMR method and the new method—specific internal energy at $t = 6.0$, $N_x = 1440$: entire computational domain—left, zoom—right.

The numerical results presented in this subsection demonstrate that, on these 1D problems, our new method shows comparable performance to other known remapping methods on a staggered mesh, i.e., the nodal momentum remap method, the half-interval-shift method, and the method of moments.

7.1.2. Convergence tests

In this subsection we investigate numerically the convergence of our new method for the 1D test problems described in the previous section. Recall that all these problems are run in the Eulerian framework.

7.1.2.1. Sod problem. In Fig. 9 we present the exact solution and numerical results for the density for resolutions $N_x = 50, 100, 200$. In Table 1 we present the L_1 errors for density and corresponding estimates for the convergence rate, which is close to 2.

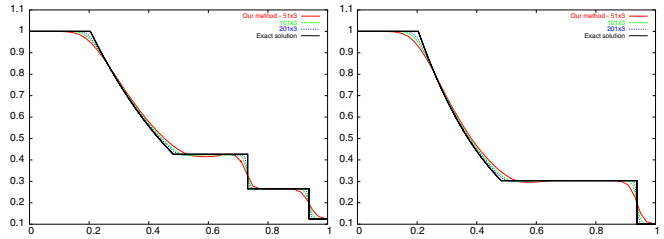


Fig. 9. Convergence for Sod problem: density and pressure at $t = 0.25$ for $N_x = 50, 100, 200$.

Table 1
Sod problem

N_x	L_1 Error	Convergence rate
50	5.63E−4	–
100	1.52E−4	1.88
200	4.22E−5	1.87
400	1.17E−5	1.85
800	3.23E−6	1.86
1600	8.77E−7	1.88

Errors and convergence rate.

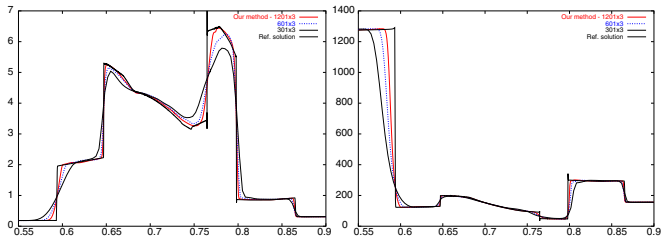


Fig. 10. Convergence for Woodward–Colella blast wave problem. Density (zoom)—left, and specific internal energy (zoom)—right at $t = 0.038$, $N_x = 300, 600, 1200$.

7.1.2.2. Woodward–Colella blast wave problem. In Fig. 10 we graphically demonstrate the convergence rate for the Woodward–Colella blast wave problem on the set of meshes with resolutions $N_x = 300, 600, 1200$. We present numerical results both for density and for specific internal energy. Because there is no analytical solution for this problem we do not present a table with convergence rates.

7.1.2.3. LeBlanc shock tube problem. In Figs. 11 and 12 we present numerical results and the exact solution for the specific internal energy and pressure for the LeBlanc shock tube problem. The convergence rate is analyzed in Table 2. Table 2 demonstrates approximately first-order convergence for the specific internal energy. We note here that the initial spatially uniform mesh for LeBlanc problem creates 10^3 jump in the masses of the cells adjacent to initial discontinuity, which implies a loss of accuracy in the Lagrangian stage at the beginning of calculation. This explains the observed low order of convergence in comparison with the Sod problem.

In summary, the numerical results presented for the Sod problem, the Woodward–Colella blast wave problem, and the LeBlanc shock tube problem, indicate a convergence rate between first and second order.

7.2. Two-dimensional tests

In this subsection we present numerical results for the Sedov blast wave problem, [27], which describes the evolution of a blast wave in a point symmetric explosion; it is an example of a diverging

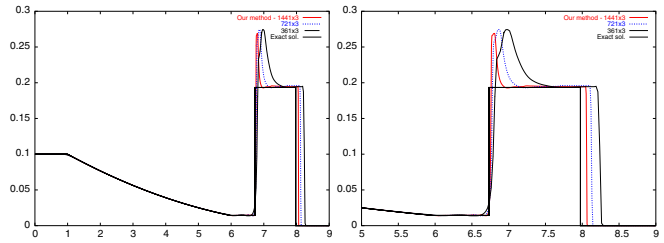


Fig. 11. Specific internal energy for the LeBlanc shock tube problem at $t = 6.0$, $N_x = 360, 720, 1440$: entire computational domain—left, zoom—right.

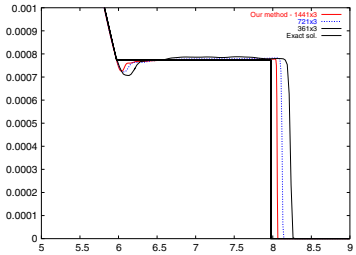


Fig. 12. Zoom for the pressure for the LeBlanc shock tube problem at $t = 6.0$, $N_x = 360, 720, 1440$. The pressure profile for the entire domain is not presented because details of shock resolution (and shock itself) are not visible due to the strong rarefaction wave.

Table 2
LeBlanc problem

N_x	L_1 Error	Convergence rate
180	7.39E−2	–
360	3.38E−2	1.13
720	1.60E−2	1.08
1440	7.79E−3	1.04
2880	3.84E−3	1.02

Errors and convergence rate.

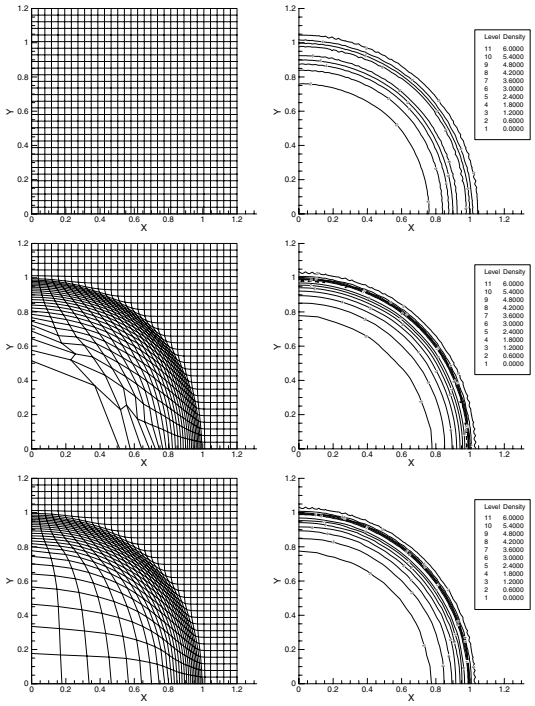


Fig. 13. Sedov problem—quadrilateral mesh. Mesh (left), and density isolines (right) at $t = 1.0$—Eulerian regime (top), Lagrangian regime (middle), ALE regime (bottom).

shock wave. We consider the cylindrically symmetric Sedov problem, in Cartesian coordinates (x, y). The total energy of the explosion is concentrated at the origin and has magnitude $E_{\text{total}} = 0.244816$ (similar to [9]). The material is an ideal gas with $\gamma = 1.4$ and initially is at rest with an initial density

equal to 1. At time $t = 1.0$ the exact solution is a cylindrically symmetric diverging shock whose front is at radius, $r = \sqrt{x^2 + y^2} = 1$ and has a peak density of 6.0 (the solid line in Fig. 14). In our numerical experiments E_{total} is concentrated in one cell located at the origin (that is, containing the vertex $(x, y) = (0, 0)$). The specific internal energy of this cell, c is defined as $a(c) = E_{\text{total}}/V(c)$. Therefore the initial pressure is $p = (\gamma - 1)\rho\varepsilon = 0.4E_{\text{total}}/V(c)$. For this problem we compare results obtained by our 2D code in three different frameworks: a purely Eulerian, a purely Lagrangian, and an ALE framework. Simulations in all three frameworks have been carried out on both quadrilateral and polygonal meshes. In the ALE calculation, the rezoning/remapping is performed once every 10 Lagrangian steps. The CFL number is chosen to be equal to 0.25 for all simulations. For each simulation we show both the initial and the final mesh, with 11 density isolines equally distributed in magnitude between 0.0 and 6.0. (Figs. 13 and 15). Each isoline has a label that refers to a density value in the legend scale. Also we show a 1D plot of density as a function of the radius, r, and a corresponding plot of the exact solution (Figs. 14 and 16). The 1D plots demonstrate how well the numerical solution preserves cylindrical symmetry.

7.2.1. Quadrilateral meshes

For this set of simulations, the computational domain is a square $(x, y) \in [0 : 1.2] \times [0 : 1.2]$ whose initial mesh consists of 31×31 square cells (top-left mesh in Fig. 13). The two top panels in Fig. 13 and the left panel in Fig. 14 shows the results of purely Eulerian computations. The symmetry of the solution is preserved quite well but the density peak is diminished ($\rho_{\max} = 3.55$ instead of 6) and the shock wave is spread over several cells. The two panels in the middle of Fig. 13 and the central panel in Fig. 14 shows the results of purely Lagrangian computations. The peak density magnitude, 4.9, is much closer to the correct value than is the Eulerian computational value. Also, the symmetry is better preserved in the Lagrangian calculation, especially near the peak. However, the Lagrangian mesh has a very low geometrical quality near the axis. The two bottom panels in Fig. 13 and right panel in Fig. 14 shows results of the ALE computations. The symmetry of the solution is even better than was found in the Lagrangian calculations and the peak density is 4.75 which is little bit smaller than in the Lagrangian calculations. The geometrical quality of the mesh is significantly improved in comparison with the Lagrangian case. In the top part of Table 3 we present the peak density values and also the number of time steps needed to reach the final time of $t = 1.0$ for the Eulerian, Lagrangian and ALE computations. It is interesting to note that the ALE computation takes the least number of time steps.

The ratio between the CPU time spent for the Eulerian regime versus the Lagrangian regime is ~10, between the ALE regime and the Lagrangian one is ~2. We remark that these timing comparisons are strongly dependent on the details of implementation and are presented to the reader as very "rough" estimates.

7.2.2. Polygonal meshes

The computational domain is one quarter of a circular disk with radius of $r_{\max} = 1.2$. A polygonal mesh is constructed in the computational domain using a Voronoi diagrams (see for example, [24]) for the set of point defined as follows:

$$x_{i,j} = r_j \sin(\theta_{i,j}) y_{i,j} = r_j \sin(\theta_{i,j}); \qquad j = 1, \ldots, J; \quad i = 1, \ldots I(j).$$

where

$$r_j = r_{\max} \cdot \frac{j-1}{J}, \quad I(j) = \mathbf{round}((j-1)\frac{\pi}{2}), \quad \theta_{i,j} = \frac{i-1}{I(j)} \cdot \frac{\pi}{2}, \quad J = 31.$$

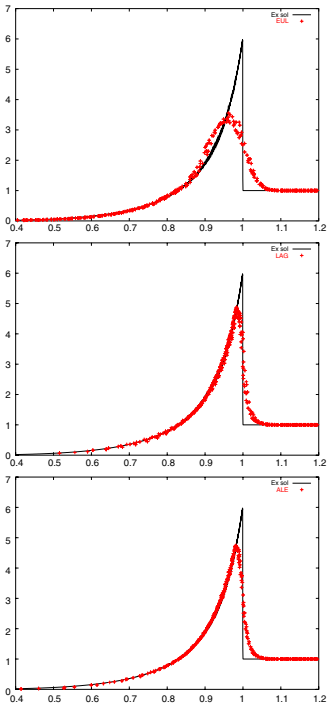


Fig. 14. Sedov problem—quadrilateral mesh. Density at $t = 1.0$ as a function of the radius (solid line exact solution)—Eulerian regime (top), Lagrangian regime (middle), ALE regime (bottom).

and function **round**(x) returns the closest integer to x. According to these formulas, on each circle of radius r_j points are distributed so that the distance between adjacent points along the circle is approximately equal to $\Delta r = r_{\max}/(J - 1)$. The total number of points is 775. There is exactly one Voronoi cell corresponding to

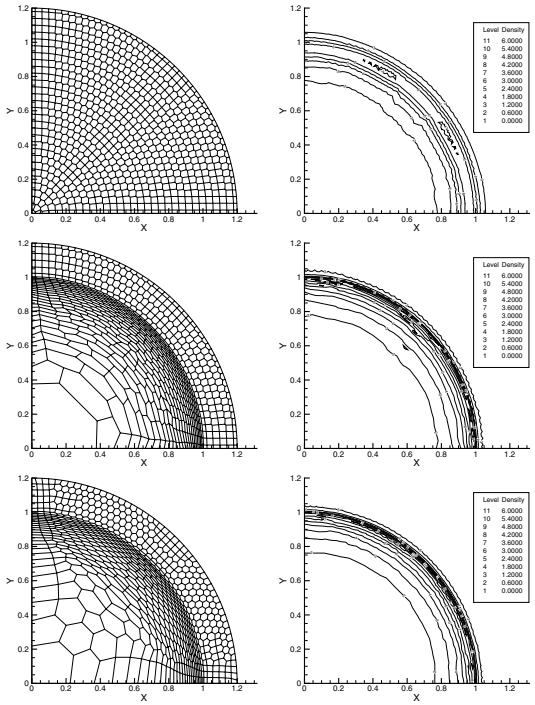


Fig. 15. Sedov problem—polygonal mesh. Mesh (left), and density isolines (right) at $t = 1.0$—Eulerian regime (top), Lagrangian regime (middle), ALE regime (bottom).

each point. The mesh consists of a mixture of convex polygons: quadrilaterals, pentagons and hexagons, and the total number of vertices is 1325; the mesh is shown in Fig. 15 (top-left panel). The resulting polygonal mesh has approximately the same resolution as the quadrilateral mesh presented in Fig. 13. Numerical

Table 3
Sedov problem

	# of time steps	Peak density	Mesh type
Eulerian	477	3.55	Quad
Lagrangian	375	4.90	Quad
ALE-10	338	4.75	Quad
Eulerian	1567	3.69	Poly
Lagrangian	603	6.20	Poly
ALE-10	408	5.70	Poly

Number of time steps needed to reach final time $t = 1.0$ and peak density values.

results for the initially polygonal mesh are arranged in a similar way as was done for our study of the quadrilateral meshes and are presented in Figs. 15 and 16, and bottom of Table 3. Qualitatively, the relative performance of purely Eulerian, purely Lagrangian, and ALE methods on polygonal meshes is the same as for quadrilateral meshes. The results of the purely Eulerian and purely Lagrangian calculations on the polygonal mesh exhibit less symmetry than the corresponding calculations on the quadrilateral meshes. However, the polygonal mesh behaves better near the axes even for purely Lagrangian calculations. In this case the ratio between the CPU time spent for the Eulerian versus the Lagrangian regimes is ~20 and between the ALE and Lagrangian regime ~2.

8. Conclusion

In this paper we have constructed a full ALE method for use on a staggered polygonal mesh. The method combines and generalizes previous work on the Lagrangian and rezoning phases, and includes a new remapping algorithm.

In the Lagrangian phase of the ALE method we use compatible methods to derive the discretizations [8,9]. We assume a staggered grid where velocity is defined at the nodes, and where density and internal energy are defined at cell centers. In addition to nodal and cell-centered quantities, our discretization employs subcell masses that serve to introduce special forces that prevent artificial grid distortion and hourglass-type motions, [10]. This adds an additional requirement to the remap phase—that the subcell densities (corresponding to subcell masses) have to be conservatively interpolated in addition to nodal velocities and cell-centered densities and internal energy.

In the remap phase, we assume that the rezone algorithm produces mesh that is "close" to Lagrangian mesh so that a local remapping algorithm (i.e., where mass and other conserved quantities are only exchanged between neighboring cells) can be used.

Our new remapping algorithm consists of three stages.

- A *gathering stage*, where we define momentum, internal energy, and kinetic energy in the subcells in a conservative way such that the corresponding total quantities in the cell are the same as at the end of the Lagrangian phase.
- A *subcell remapping stage*, where we conservatively remap mass, momentum, internal, and kinetic energy from the subcells of the Lagrangian mesh to the subcells of the new rezoned mesh.
- A *scattering stage*, where we conservatively recover the primary variables: subcell density, nodal velocity, and cell-centered specific internal energy on the new rezoned mesh.

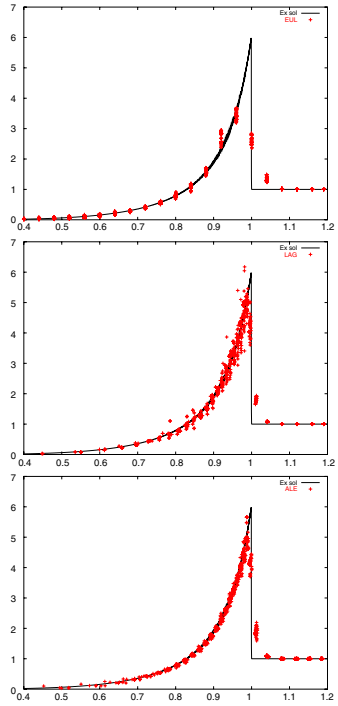


Fig. 16. Sedov problem—polygonal mesh. Density at $t = 1.0$ as a function of the radius (solid line exact solution)—Eulerian regime (top), Lagrangian regime (middle), ALE regime (bottom).

We have proved that our new remapping algorithm is conservative, reversible, and satisfies the DeBar consistency condition.

We have also demonstrated computationally that our new remapping method is robust and accurate for a series of test problems in one and two dimensions.

Acknowledgements

The authors thank L. Margolin, B. Rider, S. Li, B. Wendroff, R. Anderson, R. Pember, D. Benson, and T. Dey for fruitful discussions. Special thanks go to K. Lipnikov for developing the procedure for the analytical inversion of matrices described in Appendix A.

This work was performed under the auspices of the US Department of Energy at Los Alamos National Laboratory, under contract W-7405-ENG-36. The authors acknowledge the partial support of the DOE/ASCR Program in the Applied Mathematical Sciences and the Laboratory Directed Research and Development program (LDRD). The authors also acknowledge the partial support of DOE's Advanced Simulation and Computing (ASC) program.

Appendix A. Details of the velocity gathering

Invertibility: Here we present a constructive proof of the invertibility of $\bar{\mathbf{l}}_c$ (see Eq. (3.15)). The analytical inversion can be performed by taking into account the specific form of the matrix $\bar{\mathbf{l}}_c$, and then rewriting $\bar{\mathbf{l}}_c$ into the form

$$\bar{\mathbf{l}}_c = \bar{\mathbf{S}} + \mathbf{w}\mathbf{r}^t, \tag{A.1}$$

where

$$\bar{\mathbf{S}} = \frac{1}{4}\begin{pmatrix} 2 & \frac{1}{2} & 0 & 0 & \cdots & 0 & \frac{1}{2} \\ \frac{1}{2} & 2 & \frac{1}{2} & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{2} & 2 & \frac{1}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{2} & 0 & 0 & 0 & \cdots & \frac{1}{2} & 2 \end{pmatrix}, \tag{A.2}$$

and

$$\mathbf{w} = (1, 1, \ldots, 1)^t, \quad \mathbf{r} = (Q_{N,1,2}, Q_{1,2,3}, \ldots, Q_{N-1,N,1})^t. \tag{A.3}$$

Lemma. $\bar{\mathbf{l}}_c$ is invertible and $(\bar{\mathbf{l}}_c)^{-1}$ is given by the following formula:

$$(\bar{\mathbf{l}}_c)^{-1} = \bar{\mathbf{S}}^{-1} - \mathbf{w}\mathbf{r}^t\bar{\mathbf{S}}^{-1}. \tag{A.4}$$

Proof. Let us first remark that

$$\bar{\mathbf{S}}\mathbf{w} = \frac{3}{4}\mathbf{w}, \tag{A.5}$$

and

$$\mathbf{r}^t\mathbf{w} = (Q_{N,1,2}, Q_{1,2,3}, \ldots, Q_{N-1,N,1}).(1, 1, \ldots, 1)^t = \sum_{n=1}^{N} Q_{n^-,n,n^+} \tag{A.6}$$

$$= \frac{1}{8m(c)}\sum_{n=1}^{N}(-m(cn^-) + 4m(cn) - m(cn^+)) \tag{A.7}$$

$$= \frac{1}{8m(c)}\sum_{n=1}^{N} 2m(cn) = \frac{2m(c)}{8m(c)} = \frac{1}{4}. \tag{A.8}$$

Now we verify that $\bar{\mathbf{l}}_c(\bar{\mathbf{l}}_c)^{-1} = \bar{\mathbf{E}}$, where $\bar{\mathbf{E}}$ is the identity matrix

$$\bar{\mathbf{l}}_c(\bar{\mathbf{l}}_c)^{-1} = (\bar{\mathbf{S}} + \mathbf{w}\mathbf{r}^t)\left(\bar{\mathbf{S}}^{-1} - \mathbf{w}\mathbf{r}^t\bar{\mathbf{S}}^{-1}\right) = \bar{\mathbf{E}} - \underbrace{(\bar{\mathbf{S}}\mathbf{w})}_{=3/4\mathbf{w}}\mathbf{r}^t\bar{\mathbf{S}}^{-1} + \mathbf{w}\mathbf{r}^t\bar{\mathbf{S}}^{-1} - \mathbf{w}\underbrace{(\mathbf{r}^t\mathbf{w})}_{=1/4}\mathbf{r}^t\bar{\mathbf{S}}^{-1}$$

$$= \bar{\mathbf{E}} + \underbrace{\left(-\frac{3}{4} + 1 - \frac{1}{4}\right)}_{=0}\mathbf{w}\mathbf{r}^t\bar{\mathbf{S}}^{-1} = \bar{\mathbf{E}}.$$

Therefore $\bar{\mathbf{l}}_c$ is invertible and $(\bar{\mathbf{l}}_c)^{-1}$ is given by Eq. (A.4). □

The matrix $\bar{\mathbf{S}}$ can be exactly inverted for every size (recall that the dimension of $\bar{\mathbf{S}}$ is 3 if cell c is a triangle, 4 for a quadrilateral cell, etc). So once inverse matrices have been stored for every reasonable integer, then $(\bar{\mathbf{l}}_c)^{-1}$ can be easily computed using Eq. (A.4).

1D Analog: The meaning of formula (3.12) becomes more clear in 1D, where it is more natural to return to standard notations, Fig. 17. For nodes, we will use integer indexes, $i, i+1$ and so on. For cells, we will use half indexes, such that cell with end nodes $i, i+1$ has index $i+1/2$. Subcells , cn, now will be denoted like $i+1/2, i$. Also in 1D we will use of subscripts. For instance, nodal velocities are denoted by u_i, u_{i+1}, cell-centered velocities are $u_{i+1/2}$, and subcell velocities are $u_{i,i+1/2}$ and similar for other quantities.

In the 1D boundary of cell $c = i + 1/2$ consists of two nodes $n^- = i$, $n^+ = i + 1$. The subcell velocities are defined as follows (analog of formula (3.7))

$$u_{i+\frac{1}{2},i} = \frac{u_i + u_{i+\frac{1}{2}}}{2}, \quad u_{i+\frac{1}{2},i+1} = \frac{u_{i+\frac{1}{2}} + u_{i+1}}{2}. \tag{A.9}$$

This is a natural definition because subcell velocity is associated with the center of the subcell and it is natural to define it as an average of the velocities of the subcell end points.

The 1D analog of formula (3.5) is

$$m_{i+\frac{1}{2},i}u_{i+\frac{1}{2},i} + m_{i+\frac{1}{2},i+1}u_{i+\frac{1}{2},i+1} = m_{i+\frac{1}{2},i}u_i + m_{i+\frac{1}{2},i+1}u_{i+1}. \tag{A.10}$$

The formulas (A.9) and (A.10) give the following definition of cell-centered velocity (analog of (3.8))

$$u_{i+\frac{1}{2}} = \frac{m_{i+\frac{1}{2},i}u_i + m_{i+\frac{1}{2},i+1}u_{i+1}}{m_{i+\frac{1}{2},i} + m_{i+\frac{1}{2},i+1}}, \tag{A.11}$$

which is just the mass average of nodal velocities. One also can consider this formula as result of linear interpolation of momentum between nodes. Finally, using definition (A.9) and formula (A.11) we obtain the following expressions for subcell velocities (analog of formula (3.12)):

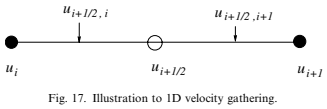


Fig. 17. Illustration to 1D velocity gathering.

$$u_{i+\frac{1}{2},j} = \frac{1}{2}u_i + \left[\frac{1}{2}\frac{1}{m_{i+\frac{1}{2}}}\left(m_{i+\frac{1}{2},j}u_i + m_{i+\frac{1}{2},i+1}u_{i+1}\right)\right],$$

$$u_{i+\frac{1}{2},i+1} = \frac{1}{2}u_{i+1} + \left[\frac{1}{2}\frac{1}{m_{i+\frac{1}{2}}}\left(m_{i+\frac{1}{2},j}u_i + m_{i+\frac{1}{2},i+1}u_{i+1}\right)\right].$$

In 1D the matrix $\bar{\mathbf{l}}_c$, (3.15), is

$$\bar{\mathbf{l}}_c = \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} + \frac{1}{2m_{i+\frac{1}{2}}}\begin{pmatrix}m_{i+\frac{1}{2},j} & m_{i+\frac{1}{2},i+1}\\ m_{i+\frac{1}{2},j} & m_{i+\frac{1}{2},i+1}\end{pmatrix} = \frac{1}{2m_{i+\frac{1}{2}}}\begin{pmatrix}2m_{i+\frac{1}{2},j} + m_{i+\frac{1}{2},i+1} & m_{i+\frac{1}{2},i+1}\\ m_{i+\frac{1}{2},j} & m_{i+\frac{1}{2},j} + 2m_{i+\frac{1}{2},i+1}\end{pmatrix}.$$

The determinant of this matrix is equal to 1/2 and therefore it is invertible.

References

[1] A.A. Amsden, H.M. Ruppel, C.W. Hirt, SALE: A Simplified ALE computer program for fluid flow at all speeds, Los Alamos National Laboratory Report LA-8095, 1980.
[2] T.J. Barth, Numerical methods for gasdynamics systems on unstructured grids, in: D. Kroner, M. Ohlberger, C. Rohde (Eds.), An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws, Freiburg/Littenweiler, October, 20–24, 1997, Lecture Notes in Computational Science and Engineering, Springer, Berlin, 1997, pp. 195–285.
[3] D.J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs, Comput. Meth. Appl. Mech. Eng. 72 (1989) 305–350.
[4] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Meth. Appl. Mech. Eng. 99 (1992) 235–394.
[5] D.J. Benson, Momentum advection on a staggered grid, J. Comput. Phys. 100 (1992) 143–162.
[6] D.E. Burton, Multidimensional discretization of conservation laws for unstructured polyhedral grids, Report UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994.
[7] D.E. Burton, Consistent finite-volume discretization of hydrodynamics conservation laws for unstructured grids, Report UCRL-JC-118788, Lawrence Livermore National Laboratory.
[8] J. Campbell, M. Shashkov, A compatible Lagrangian hydrodynamics algorithm for unstructured grids, Selcuk J. Appl. Math. 4 (2003) 53–70, report version can be found at www.cnls.lanl.gov/~shashkov.
[9] J. Campbell, M. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys. 172 (2001) 739–765.
[10] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comput. Phys. 142 (1998) 521–561.
[12] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1998) 227–262.
[13] R.B. Demuth, L.G. Margolin, B.D. Nichols, T.F. Adams, B.W. Smith, SHALE : a computer program for solid dynamics, Report Los Alamos National Laboratory Report LA-10236, 1985.
[14] C.W. Hirt, A.A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (1974) 227–253, and reprinted in 135 (1997) 203–216.
[15] D.S. Kershaw, M.K. Prasad, M.J. Shaw, J.L. Milovich, 3D unstructured mesh ale hydrodynamics with the upwind discontinuous finite element method, Comput. Meth. Appl. Mech. Eng. 158 (1998) 81–116.
[16] P. Kjellgren, J. Hyvarien, An arbitrary Lagrangian–Eulerian finite element method, Comput. Mech. 21 (1998) 81–90.
[17] P. Knupp, L.G. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 176 (2002) 93–112.
[18] M. Kucharik, M. Shashkov, B. Wendroff, An efficient linearity- and bound-preserving remapping method, J. Comput. Phys. 188 (2003) 462–471.
[19] L.G. Margolin and C.W. Beason, Remapping on the staggered mesh, Report UCRL-99682 of Lawrence Livermore National Laboratory, 1988. Available from: <www.llnl.gov/tid/lof/documents/pdf/208550.pdf>.
[20] L.G. Margolin, Introduction to an arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 135 (1997) 198–202.

[21] L.G Margolin, M. Shashkov, Remapping, recovery and repair on staggered grid, Comput. Meth. Appl. Mech. Eng. 193 (2004) 4139–4155.
[22] L.G. Margolin, M. Shashkov, Second-order sign preserving remapping on general grids, LA-UR-02-525. Available from: <www.cnls.lanl.gov/~shashkov/>.
[23] J.M. McGlaun, S.L. Thompson, M.G. Elrick, CTH: A three dimensional shock wave physics code, Int. J. Impact Eng. 10 (1990) 351–360.
[24] A. Okabe, B. Boots et al., Spatial Tessellations. Concepts and Applications of Voronoi Diagrams, 2nd ed., John Wiley, Chichester, England, 2000.
[25] R.B. Pember, R.W. Anderson, A comparison of staggered-mesh Lagrange plus remap and cell-centered direct Eulerian Godunov schemes for Eulerian shock hydrodynamics, Preprint UCRL-JC-139820, Lawrence Livermore National Laboratory, 2000. Available from: <www.llnl.gov/tid/lof/documents/pdf/238843.pdf>.
[26] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Comput. Meth. Appl. Mech. Eng. 187 (2000) 591–619.
[27] L.I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, New York, 1959.
[28] M. Shashkov, P. Knupp, Optimization-based reference-matrix rezone strategies for arbitrary Lagrangian–Eulerian methods on unstructured grids, Selcuk J. Appl. Math. 3 (2002) 81–99.
[29] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, J. Comput. Phys. 198 (2004) 265–277.
[30] G.A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1978) 1–31.
[31] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction, Springer, Berlin, 1999.
[32] P. Vachal, R. Garimella, M.J. Shashkov, R. Loubere, Untangling of meshes in ALE simulations, in: Proceedings of the 7th US National Congress on Computational Mechanics, Albuquerque, July 2003, Available from: <www.cnls.lanl.gov/~shashkov>.
[33] P. Vachal, R. Garimella, M. Shashkov, Untangling of 2D meshes in ALE simulations, J. Comput. Phys. 196 (2004) 627–644.
[34] P.R. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.

2.2.2 Repair

With B. Wendroff and M. Staley we have investigated new repair methods in a paper entitled *The Repair Paradigm : New Algorithms and Applications to Compressible Flow* in [8]. The goal in this paper is to improve upon and apply the repair idea introduced in [142, 141, 128].

A repair method can be viewed as a way to correct values on a discrete mesh by redistributing the conserved quantity so that conservation and a maximum principle are preserved. The maximum principle states that new values should obey certain upper and lower bounds obtained from old values. In this way, not only are non-physical quantities eliminated, but oscillations are reduced (albeit not necessarily eliminated). We therefore seek repair algorithms that can be applied to CFD problems, advection problems, or other situations where values of a discrete variable must be placed in bounds without violating a conservation law and without introducing significant errors in the dynamics. As stated in [141] (Section 8, page 275), repair is a mass redistribution nonlinear filter. Notice that this technique is vaguely apparented to other methods for the correction of nonphysical data, such as Flux Corrected Transport, which are discussed in [143].

As we have seen a critical part of Lagrangian-based methods and ALE methods for Computational Fluid Dynamics (CFD) is the ability to remap or interpolate data from one computational mesh to another. Remapping is also essential for pure Lagrangian methods, because they can lead to tangled grids that must then be untangled with a concomitant remap step. Even if the basic scheme produces only physically meaningful quantities, a remapping method can create out-of-bounds quantities such as negative densities or pressures. In some CFD codes, the offending values are simply set to a small positive number when this occurs, at which point mass or total energy is no longer conserved. In most instances the error thereby created is negligible, but we have shown that in at least one example the error is significant. It is possible, by taking great care with the remapping in the CFD context, to maintain positive mass density. This is done by first extending the given mean densities in each original cell to the whole domain so that the new distribution is everywhere positive, and then computing new mean values by exact integration over the cells of the new grid. Total energy can be remapped in this way, but then there is no guarantee that internal energy will be positive. Furthermore, in three dimensions, exact integration is computationally intensive. Another context in which non-physical data can occur is in divergence-free advection of a concentration that must retain values between zero and one. High quality advection schemes, some of which are based on remapping ideas ([144], [134]), unavoidably have this fault ([143]). In the case of advection of a concentration, repair keeps the newly computed concentration in a cell between the maximum and minimum concentrations in neighboring old cells, thus guaranteeing at least that the new concentration is between zero and one.

In this paper we have reviewed and applied several conservative repair methods that can be used in situations where variables must stay between predefined bounds while respecting conservation. Such situations occur often, in hydrodynamics for example, when the density or the specific internal energy becomes negative due to remapping. Such unphysical situations must be cured, but replacing negative values by small positive numbers is not acceptable from the point of view of conservation. The methods developed in this paper are

Local order-dependent repair. This is perhaps the most obvious local repair algorithm. The underlying idea is to expand the neighborhood of a cell i which needs repair, until enough room is found in this neighborhood. Suppose cell i has a negative density, but the minimum bound is 0. This repair algorithm expands the neighborhood of cell i until enough mass can be found and removed from the neighborhood to fill cell i and produce a repaired density equal to the minimum bound, that is, to 0. Then, the next cell is checked and repaired if necessary. A similar concept is applied to repair an over-bound value. This repair algorithm is order-

dependent, meaning that the final result depends on the order in which cells are visited. This unphysical order-dependence is unacceptable in many practical situations.

Global order-independent repair. A simple, global order-independent repair algorithm clips out-of-bounds values to their bounds, counts the total discrepancy this produces in the quantity to be conserved, and spreads the discrepancy over the entire mesh. This method is order-independent because any cell that has to be repaired is immediately brought to its nearest bound and contributes to a total discrepancy which is not accounted for until all individual cells have been repaired. Clearly, this algorithm is conservative and order-independent. It is also symmetry-preserving, in the sense that equivalent cells (cells which have the same mass and bounds) are treated in the same manner. The drawback is that such a repair process can violate the physics in computational fluid dynamics. In other words, such a repair process on hydrodynamics problems can severely perturb the physics of the phenomena one is trying to study ; it may violate the causality. However for a pure advection problem, the global repair algorithm can be appreciated for its simplicity and its ability to parallelize.

Local order-independent repair. We have also developed an iterative, order-independent repair algorithm that addresses the disadvantages of the previous algorithm. This is a two-stage algorithm : one stage repairs all values that are above their upper bound, and the other stage repairs all values that are below their lower bound. Upper bounds can be fixed before lower bounds, or vice versa. The order affects the result, but given a choice of order, the algorithm produces the same result regardless of the order in which cells are examined. This algorithm converges, is conservative and order-independent, and can preserve a 1D symmetry with the modifications outlines above. However, on parallel machines this algorithm (as well as the local order-dependent method) is slow. This is largely due to the neighborhood expansion needed by both algorithms.

Mixed local/global order-independent repair. The global repair algorithm needs very little communication and can be used very efficiently in a parallel framework. We consider an amalgamation of the local algorithm, which gives more physically meaningful results, and the global algorithm, which is more parallelizable. The mixed local/global order-independent repair algorithm is based on the assumption that most of the out-of-bounds cells can be fixed locally (using only the immediate neighborhood) because they are due to very small disturbances, and that, as a corollary, only a few cells need to find room/mass far away from their location. The idea of this algorithm is to repair as many cells as possible with the local order-independent algorithm, and then if some of the cells are still out-of-bounds, to repair them with the global repair algorithm.

1. **Local** treatment : for all out-of-bounds cells, try to repair with the Local order-independent symmetry-preserving algorithm, but without expanding any immediate neighborhoods. If the current cell still has excess mass, then leave it out of bounds. Iterate this process in order to converge to a situation where either every cell is repaired, or the remaining unrepaired cells cannot be repaired using their closest neighborhood. Our experimentation indicated that few iterations are needed.
2. **Global** treatment : for any remaining out-of-bounds cells, perform the global repair. This step finally fixes the remaining out-of-bounds cells, the number of which is presumably small, and which should be out of bounds only by small amounts

The mixed local/global repair algorithm is conservative, because each of its steps is conservative. Moreover, both the local and global treatments are order-independent and symmetry

preserving, and there is no particular difficulty with parallelization because there is no indefinite neighborhood expansion. If any cells are still out-of-bounds after the first step, the global repair fixes them. The earlier argument stating that this method can violate causality still holds, but the effect is far less pronounced because very few cells will remain to be fixed after the initial local treatment, and the amounts by which they need to be fixed will be less. Therefore, the causality violation is negligible.

Of the methods presented in this paper, we believe that the mixed local/global repair scheme best meets the requirements of locality and efficiency. This method was applied to an advection example and to test cases in an ALE hydrodynamics framework, where the use of a conservative repair algorithm allowed us to :

- preserve the accuracy of the underlying method, as in the Sod Riemann problem ;
- stabilize and improve bad profiles, as in the Le Blanc Riemann problem ;
- maintain physical and reasonable results, as in the blast wave interaction problem and the Sedov problem.

In Fig. 2.5 one represents the results obtained on the Sedov problem in ALE regime (rezone and remap every ten cycles) at times $t = 0.1$ and $t = 1.0$, the density variable and the mesh are displayed. At time $t = 1.0$ one provides the cell-based density value as a function of the radius (right panel). When no repair is performed in this test case, the code stops due to the creation of negative internal energy after the fourth remapping. On the other hand, the use of a repair method fixes the parasitic negative values and allows us to observe good results. The maximum density with the Mixed Local/Global repair method is 5.62.

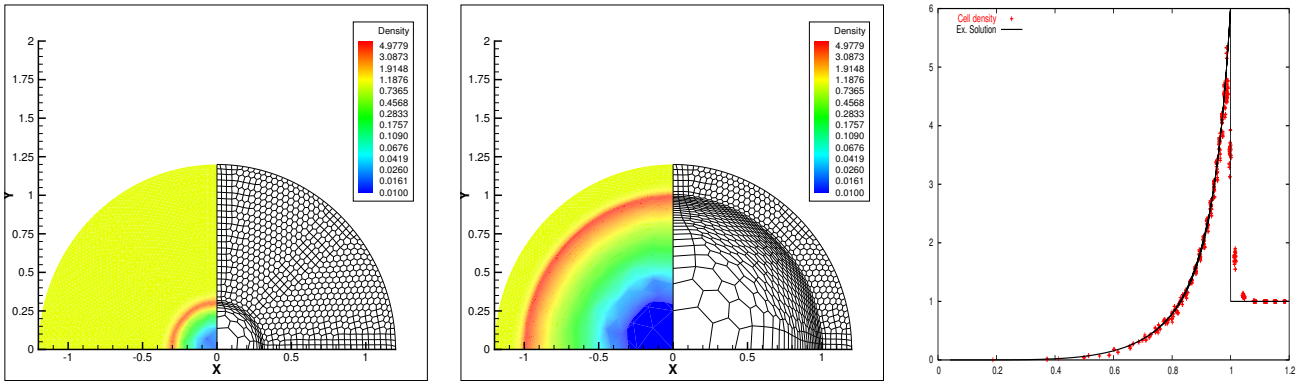


FIGURE 2.5 – *Numerical results from paper [8]. Sedov problem on polygonal Mesh —Sedov blast wave on a polygonal mesh (1325 nodes and 775 cells). ALE–1o regime — Left-Middle : mesh and density contours (exponential scale) at $t = 0.1$ and $t = 1.0$. Right : density at $t = 1.0$ (cell-based value as a function of the radius)*

This paper is reproduced in the following pages.

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 211 (2006) 385–404
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

The repair paradigm: New algorithms and applications to compressible flow

Raphaël Loubère *, Martin Staley, Burton Wendroff

Los Alamos National Laboratory, Theoretical Division T-7, MS B284, Los Alamos, NM 87545, USA

Received 14 October 2004; received in revised form 5 April 2005; accepted 20 May 2005

Abstract

The repair paradigm leads to several algorithms for redistributing mass, momentum and energy, while adhering to local maximum principles, as an adjunct to the remapping step in certain compressible flow codes that use remapping, such as Arbitrary–Lagrangian–Eulerian codes, or for just redistributing mass in advection codes. In the case of advection of a concentration, repair keeps the newly computed concentration in a cell between the maximum and minimum concentrations in neighboring old cells, thus guaranteeing at least that the new concentration is between zero and one. For compressible flow, density, velocity and internal energy are similarly constrained while maintaining conservation of mass, momentum and total energy. In this way, positive density and internal energy are achieved as a side effect. We propose a new algorithm, combining both local and global repair, that maintains causality and is efficient in a parallel computational setting. The local/global algorithm is independent of the order in which the distribution is performed, and it maintains 1D symmetry. This is applied to advection in two dimensions, and to, among others, the LeBlanc problem, the Sedov problem, and an interacting 2D blast wave problem. The latter is done with a Lagrangian code for which rezoning, remapping and repair are essential.
© 2005 Elsevier Inc. All rights reserved.

Keywords: Repair; Mass redistribution; Conservative reconstruction; Remapping; Advection; Hydrodynamics

1. Introduction

A critical part of Lagrangian-based methods for computational fluid dynamics (CFD) is the ability to remap or interpolate data from one computational mesh to another. This is the case for the popular ALE schemes that perform Lagrangian steps followed by remaps to fixed grids. Remapping is also essential for pure Lagrangian methods, because they can lead to tangled grids that must then be untangled with a concomitant remap step. Even if the basic scheme produces only physically meaningful quantities, a remapping method can create out-of-bounds quantities such as negative densities or pressures. In some CFD codes, the offending values are simply set to a small positive number when this occurs, at which point mass or total energy is no longer conserved. In most instances the error thereby created is negligible, but we shall show that in at least one example the error is significant.

It is possible, by taking great care with the remapping in the CFD context, to maintain positive mass density. This is done by first extending the given mean densities in each original cell to the whole domain so that the new distribution is everywhere positive, and then computing new mean values by exact integration over the cells of the new grid. Total energy can be remapped in this way, but then there is no guarantee that internal energy will be positive. Furthermore, in more than one dimension, exact integration is computationally intensive.

Another context in which non-physical data can occur is in divergence-free advection of a concentration that must retain values between zero and one. High quality advection schemes, some of which are based on remapping ideas [1,2], unavoidably have this fault [3].

The goal in this paper is to improve upon and apply the repair idea introduced in [4,5]. A repair method can be viewed as a way to correct values on a discrete mesh by redistributing the conserved quantity so that conservation and a maximum principle are preserved. The maximum principle is that new values should obey certain upper and lower bounds obtained from old values. In this way, not only are non-physical quantities eliminated, but oscillations are reduced (albeit not necessarily eliminated). We therefore seek repair algorithms that can be applied to CFD problems, advection problems, or other situations where values of a discrete variable must be placed in bounds without violating a conservation law and without introducing significant errors in the dynamics.

As stated in [4] (Section 8, p275), repair is a mass redistribution nonlinear filter. Other methods for the correction of nonphysical data, such as flux corrected transport, are discussed in [3].

The rest of this paper is arranged as follows. We first present notation, goals and expected properties of repair methods. We then review a local repair method [5] which repairs out-of-bounds values and distributes the resulting mass discrepancies locally. This method can produce different results depending on the order in which cells are visited, and it is therefore called order-dependent. Next we review a simple global repair process [4] which repairs out-of-bounds values and distributes the resulting mass discrepancy across the entire grid. The next two sections introduce order-independent local methods, and we conclude with a discussion of repair methods in advection and hydrodynamics contexts, where numerical tests are performed to show the effects of such methods.

2. Notation, goals and properties

Repair methods can be used for many kinds of variables, including density, velocity, energy, pressure, and concentration, but we will henceforth call our variable to be repaired a density ρ, or equivalently, a mass m. If we denote old cells by c and new cells by $\tilde{c}$, then the quantity to be conserved is the total mass $m = \sum_{\tilde{c}} m(\tilde{c}) = \sum_c \rho(c)V(c)$, where $m(c)$, $\rho(c)$, and $V(c)$ denote the mass, density, and volume, respectively, of cell c.

Consider an old mesh $\mathscr{M}$ with cell-averaged densities (called old densities), and a new mesh $\widetilde{\mathscr{M}}$ with remapped cell-averaged densities (called new densities). We assume for simplicity that the connectivity is the same for the old and new grids.

In the case of advection the meshes would coincide, but typically the new mesh is a small perturbation of the old one. Define the *bound* neighborhood $N(c)$ of a cell c as a patch of surrounding cells,

* Corresponding author. Tel.: +1 505 667 1407.
E-mail addresses: loubere@lanl.gov (R. Loubère), mstaley@lanl.gov (M. Staley), bbw@lanl.gov (B. Wendroff).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.05.010

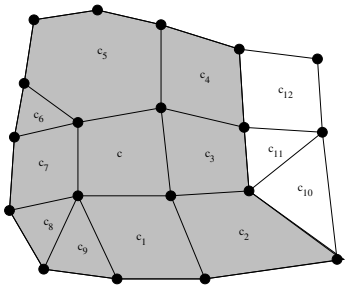


Fig. 1. An example of a neighborhood. Any cell in contact with c is an element of the set: $N(c) = \{c, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$. In this example, c_{10}, c_{11}, and c_{12} are not part of the neighborhood of cell c.

as in Fig. 1. The notion of neighborhood is essential for any repair method. However it is a user choice how to define the notion of neighborhood. In our case the neighborhood of a given cell c is composed of all cells in contact with c, through either a node or an edge. (Alternatively, we could have defined the neighborhood as the set of all cells in contact with c through an edge, for example.) Formally,

$$N(c) = \{c', \quad c' \bigcap c \neq \varnothing\}. \tag{1}$$

$|N(c)|$ denotes the number of elements in the set $N(c)$. For example, $|N(c)| = 10$ in Fig. 1. The neighborhood can be expanded with an iterative process defined as follow ($N^1(c) = N(c)$):

$$\forall v > 1, \quad N^v(c) = \bigcup_{c' \in N^{v-1}(c)} N(c'). \tag{2}$$

Using this neighborhood definition, we can define maximum and minimum density bounds as $\rho_+(c) = \max_{s \in N(c)} \rho(s)$ and $\rho_-(c) = \min_{s \in N(c)} \rho(s)$. (There are, of course, other reasonable ways to define density bounds.) No matter how these bounds are defined, a feasibility condition is required in order for repair to work at all.

Feasibility. The total mass m must not exceed (respectively be below) the total upper bound mass (respectively the total lower bound mass), that is, the total mass if each new cell were at its upper (respectively lower) bound.

Let's illustrate the feasibility condition on a simple example: if the mass in each cell, of a 1D mesh having K cells, is m, and each upper bound mass is u and $m > u$, then the total mass is $M = K \cdot m$ and the upper bound mass is $U = K \cdot u$; hence $M > U$ and it is not feasible to repair every cell. M units of mass (which we must keep in order to maintain mass conservation) simply cannot fit into U units of mass.

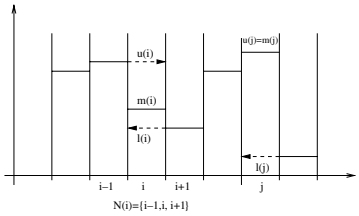


Fig. 2. Illustration of upper and lower bounds in 1D. We examine the cell masses before remapping takes place. The neighborhood of cell i consists of cells $i-1$, i, and $i+1$. Then, $u(i) = m(i-1)$ and $l(i) = m(i+1)$, while $u(j) = m(j)$ and $l(j) = m(j+1)$. After remapping, $m(i)$ can be out-of-bounds. For example, if $m(i) > u(i)$ then cell i needs repair.

Repair is not possible if this condition is violated. However, any repair procedure of the type that truncates out-of-bounds values and redistributes the discrepancy does work if it is satisfied.

If a remapping process produces negative densities $\rho(\tilde{c})$, or more generally produces out-of-bounds densities, then a repair step must be done to make these densities obey their bounds. The properties to be fulfilled by a repair method are:

Conservation:

$$\sum_{\tilde{c}} m(\tilde{c}) = \sum_c \rho(c)V(c) = \sum_{\tilde{c}} \rho(\tilde{c})V(\tilde{c}) = \sum_{\tilde{c}} m(\tilde{c}). \tag{3}$$

Maximum principle:

$$\forall c, \quad \rho_-(c) \leqslant \rho(\tilde{c}) \leqslant \rho_+(c). \tag{4}$$

The original grid plays no role in the repair process, other than to provide bounds, so for brevity we henceforth use i rather than $\tilde{c}$ to represent new cell indices. Let us write $u(i)$ and $l(i)$ for the upper and lower mass bounds, respectively, of cell i. Then,

$$u(i) = V(i)\rho_+(i) = V(i) \max_{s \in N(i)} \rho(s), \tag{5}$$

$$l(i) = V(i)\rho_-(i) = V(i) \min_{s \in N(i)} \rho(s). \tag{6}$$

These have dimensions of mass, while the mass of cell i is $m(i) = V(i)\rho(i)$.

In Fig. 2 we illustrate $u(i)$ and $l(i)$ in 1D. The neighborhood of a cell i is given by i and the two adjacent cells: $i-1$ and $i+1$. Therefore, we have $u(i) = \max(m(i-1), m(i), m(i+1))$ and $l(i) = \min(m(i-1), m(i), m(i+1))$.

3. Local order-dependent repair

This is perhaps the most obvious local repair algorithm. The underlying idea is to expand the neighborhood of a cell i which needs repair, until enough room is found in this neighborhood. Suppose cell i has a negative density, but the minimum bound is 0. This repair algorithm expands the neighborhood of cell i

until enough mass can be found and removed from the neighborhood to fill cell i and produce a repaired density equal to the minimum bound, that is, to 0. Then, the next cell is checked and repaired if necessary. A similar concept is applied to repair an over-bound value.

3.1. Algorithm

Suppose cell i, with upper bound $u(i)$, must be repaired because $m(i) > u(i)$, that is, because cell i's mass is above its upper bound. (A value that is below its lower bound would be treated similarly.) The first step is to compute the value needed to repair cell i:

$$m_{\mathtt{need}}(i) = m(i) - u(i). \tag{7}$$

This value is "needed" in the sense that if we repair $m(i)$ by reducing it to $u(i)$, then the loss $m_{\mathtt{need}}(i)$ must be added elsewhere in order to maintain conservation.

Next, we choose a set of neighboring cells $N(i)$, called the search neighborhood, and we check $N(i)$ to see if enough room is available to accommodate $m_{\mathtt{need}}(i)$:

$$m^{\mathtt{total}}_{\mathtt{avail}}(i) = \sum_{j\in N(i)} \max(u(j) - m(j), 0) \tag{8}$$

$$= \sum_{j\in N(i)} m_{\mathtt{avail}}(j), \tag{9}$$

where $m_{\mathtt{avail}}(j) \equiv \max(u(j) - m(j), 0)$ is the amount by which cell j's value, $m(j)$, can be increased without exceeding its upper bound, $u(j)$. If enough room is available in the neighborhood, that is, if $m^{\mathtt{total}}_{\mathtt{avail}}(i) \geqslant m_{\mathtt{need}}(i)$, then the repair of cell i succeeds, and we make the following adjustments:

$$m(i) \leftarrow u(i), \tag{10}$$

$$\forall j \in N(i), \quad j \neq i, \quad m(j) \leftarrow m(j) + m_{\mathtt{need}}(i)\ \frac{m_{\mathtt{avail}}(j)}{m^{\mathtt{total}}_{\mathtt{avail}}(i)}. \tag{11}$$

That is, we clip $m(i)$ to its upper bound, then add its lost mass proportionally to all acceptor cells (cells whose masses are below their upper bounds, so that they can accept more mass) in its neighborhood.

If the search neighborhood $N(i)$ does not have enough room to accept cell i's excess mass, we extend the neighborhood until enough mass is available. It was shown in [4] that a sufficient condition for such an extension process to successfully terminate for repair of densities is that the (old) cells of the *bound* neighborhood cover the new cell i.

For each cell i, the local repair algorithm performs the procedure outlined above if the cell's value is above its upper bound, or performs a similar procedure if the cell's value is below its lower bound.

3.2. Properties and issues

This repair algorithm is order-dependent, meaning that the final result depends on the order in which cells are visited. This is because out-of-bounds cells are repaired as soon as they are found, and once it has been repaired, a cell's other properties, such as its ability to accept excess mass from elsewhere, change. Moreover, in repairing a cell the properties of some of its neighbors change as well, as mass is transferred between them and the cell being repaired.

This unphysical order-dependence is unacceptable in many practical situations, so we now focus our attention on developing order-independent repair algorithms.

4. Global order-independent repair

A simple, global order-independent repair algorithm clips out-of-bounds values to their bounds, counts the total discrepancy this produces in the quantity to be conserved, and spreads the discrepancy over the entire mesh. This method is order-independent because any cell that has to be repaired is immediately brought to its nearest bound and contributes to a total discrepancy which is not accounted for until all individual cells have been repaired.

4.1. Algorithm

The global repair algorithm begins by repairing every cell that needs repair, while keeping track of the discrepancy, that is, the change in total mass due to performing the repairs. Let each new cell have computed mass $m(i)$, upper bound $u(i)$ and lower bound $l(i)$ as defined in (5) and (6), and a neighborhood $N(i)$ as defined in (1). The upper and lower bounds are fixed numbers, while the values m and the discrepancy Δ evolve as repair progresses, as follows.

$$\Delta = \sum_i \max(0, m(i) - u(i)) - \sum_i \max(0, l(i) - m(i)), \tag{12}$$

$$m(i) \leftarrow \begin{cases} u(i) & \text{if } m(i) > u(i), \\ m(i) & \text{if } l(i) \leqslant m(i) \leqslant u(i), \\ l(i) & \text{if } m(i) < l(i). \end{cases} \tag{13}$$

Note that values above their upper bounds make positive contributions to Δ, while values below their lower bounds make negative contributions to Δ.

Each m is now within its bounds. If the total discrepancy, Δ, is zero, we stop. (This unlikely scenario would mean that the total above-bound mass precisely equaled the total below-bound mass.) Otherwise, we must add a total of Δ to the values of cells in the mesh that can accept it. Note that this works regardless of Δ's algebraic sign. A positive Δ means the bulk of out-of-bounds masses were above their upper bounds, in which case the procedure decreased those masses to their upper bounds, and we must add the (positive) Δ elsewhere to make up for the loss. A negative Δ means the bulk of out-of-bounds masses were below their lower bounds, in which case the procedure increased those masses to their lower bounds, and we must add the (negative) Δ elsewhere to get rid of the excess.

To adjust for the discrepancy Δ, we begin by computing the following:

$$k(i) = \begin{cases} u(i) - m(i), & \text{if } \Delta > 0, \\ m(i) - l(i), & \text{if } \Delta < 0. \end{cases} \tag{14}$$

For each cell i, this is the amount by which the cell's value is allowed to increase (if $\Delta > 0$) or decrease (if $\Delta < 0$) without going out of bounds. By construction, $k(i)$ is always positive or zero, because $m(i)$ has already been placed between its bounds. Whether $k(i)$ is interpreted as an allowable increase or an allowable decrease is determined by the algebraic sign of Δ.

Finally, this repair procedure cancels the discrepancy Δ as follows:

$$\forall i, \quad m(i) \leftarrow m(i) + \Delta\frac{k(i)}{K}, \tag{15}$$

where $K = \sum_i k(i)$ is the total allowable increase (if $\Delta > 0$) or decrease (if $\Delta < 0$) over all cells. Again by construction, $K > 0$, and we see that for applicable cells i, $m(i)$ increases if $\Delta > 0$ and decreases if $\Delta < 0$. (Of course, for some cells $k(i) = 0$, and $m(i)$ for those cells is unchanged in this step.) Using the fraction $k(i)/K$ means cells are adjusted in proportion to how much room they have.

The above procedure can be illustrated with the following example. Consider a mesh of four cells, for which the first cell's mass is 3 units above its allowable maximum, the second cell's mass is within bounds and 5 units below its maximum, the third cell's mass is within bounds and 6 units below its maximum, and the fourth cell's mass is at its maximum. The first cell is fixed by decreasing its mass by 3 units. To conserve total mass, we must add 3 units elsewhere. The second cell can take 5 units, the third cell 6 units, and the fourth cell 0 units. According to (15) cell 2's mass is increased by $3 \times 5/11$, and cell 3's mass is increased by $3 \times 6/11$.

Cell 3	Cell 4
u = 6	u = 1
m = 0	m = 1
l = 0	l = 0
Cell 1	**Cell 2**
u = 0	u = 5
m = 3	m = 0
l = 0	l = 0

$\Longrightarrow$

Cell 3	Cell 4
u = 6	u = 1
m = 0 + 3 ←6/11	m = 1
l = 0	l = 0
Cell 1	**Cell 2**
u = 0	u = 5
m = 0	m = 0 + 3 ←5/11
l = 0	l = 0

4.2. Properties and issues

Clearly, this algorithm is conservative and order-independent. It is also symmetry-preserving, in the sense that equivalent cells (cells which have the same mass and bounds) are treated in the same manner. Moreover, the algorithm is well suited for parallelization.

The drawback is that such a repair process can violate the physics in computational fluid dynamics. Consider its behavior on a simple example, the Sod Riemann problem in 2D, in which a 1D rarefaction wave, a contact discontinuity, and a shock wave begin to differentiate from each other. Such waves are separated by plateaus. Suppose an over-bound value of energy in the post-shocked region has to be repaired. With the present algorithm, this excess energy is spread uniformly over all acceptor cells – all cells that can accept some mass – even if those cells are in the rarefaction wave or are far from the shock wave. So, energy that should have been distributed close to the shocked region appears far away. By repeating such a repair process on every time step, energy is gradually removed from its physical position and scattered elsewhere, and the result is a poor approximation of shock speed, density/energy plateaus, etc. In other words, such a repair process on hydrodynamics problems can severely perturb the physics of the phenomena one is trying to study.

Such an argument can actually be applied to every repair method we present. Because a redistribution of an out-of-bounds cell's value is always involved, its excess energy appears elsewhere instantaneously. However, with local algorithms, in which out-of-bounds values are redistributed in a neighborhood of the cell in question, we have not experienced such modification of the physics. Expanding a cell's neighborhood only as far as necessary provides the smallest neighborhood into which mass must be redistributed, while with the global repair algorithm a cell's "neighborhood" is the entire domain.

For a pure advection problem, the global repair algorithm can be appreciated for its simplicity and its ability to parallelize.

5. Local order-independent repair

We have developed a new, iterative, order-independent repair algorithm that addresses the disadvantages of the previous algorithm. This algorithm is known to converge, as we will show in the next section.

5.1. Algorithm

This is a two-stage algorithm: one stage repairs all values that are above their upper bound, and the other stage repairs all values that are below their lower bound. Upper bounds can be fixed before lower bounds, or vice versa. The order affects the result, but given a choice of order, the algorithm produces the same result regardless of the order in which *cells* are examined. Repair of the upper bounds proceeds as follows.

Upper-bounds repair. If $m(i) \leqslant u(i)$ for every cell i, then no upper-bounds repair is needed. Otherwise:

(1) $\forall i$, let $\delta(i) = 0$. As the algorithm proceeds, $\delta(i)$ can accumulate portions of nearby cells that are above their upper bounds, if there is room for those portions in cell i.
(2) While $\exists i$ with $m(i) > u(i)$, iterate the following.
(3) $\forall i$ with $m(i) > u(i)$, do the following. First, let $e(i) = m(i) - u(i)$. This is cell i's excess mass, which we must distribute to nearby cells. Find the smallest neighborhood $N(i)$ whose acceptor cells j (cells whose values are below their maximum bounds) can accept, in total, at least $e(i)$ units of mass: $a \equiv \sum_j (u(j) - m(j)) \geqslant e(i)$. For each of these acceptor cells j, let $\delta(j) \leftarrow \delta(j) + e(i) \cdot (u(j) - m(j))/a$. That is, distribute cell i's excess mass into the δ's of neighboring acceptor cells j, in proportion to what each of these acceptor cells can receive. Note that we haven't yet modified m anywhere.
(4) $\forall i$, if $m(i) > u(i)$ then repair the cell: set $m(i) \leftarrow u(i)$. Otherwise, check $\delta(i)$ to see if cell i was an acceptor for some other cell's excess mass. If $\delta(i) > 0$ then it was, and we now "accept" that mass: set $m(i) \leftarrow m(i) + \delta(i)$. However, doing this might put $m(i)$ above its upper bound, in which case we set a flag indicating that another iteration is necessary.
(5) Set $\delta(i) = 0$, because $\delta(i)$ has now been accounted for.
(6) Iterate (go to step 3) if necessary.

We omit the procedure for repairing lower bounds, which is similar.

In step 4 we remarked that the modification $m(i) \leftarrow m(i) + \delta(i)$ might put $m(i)$ above its upper bound. Recall that the individual contributions to any cell's δ, in the form $e(i) \cdot (u(j) - m(j))/a$ as in step 3, are by construction too small to put the cell's value above its upper bound. However, an acceptor cell can accept such contributions from many nearby out-of-bounds cells, and the sum of those contributions, accumulated in δ and finally added to $m(i)$ in step 4, can possibly put $m(i)$ above its upper bound. If this happens, then this $m(i)$ will be fixed in the next iteration.

Each iteration fixes all values that were above their upper bounds at the beginning of the iteration, but as we have just seen, doing so can put other values above their upper bounds. Although this can happen, the algorithm is guaranteed to converge. Once a cell is repaired, it can never receive a nonzero δ, and can therefore never be broken. Because each iteration fixes at least one cell, and never breaks a cell that has already been fixed, the algorithm must converge (in at most N iterations, if N is the number of cells) if repair is feasible at all.

This algorithm is conservative, and it achieves order-independence by distributing excess mass into δ and only taking δ into account later. However, in the above form the algorithm does not preserve a 1D symmetry. Consider the following symmetric example, in which cells 1, 2, and 3 are each 1 unit above their maximum, and cells 4, 5, and 6 are within their bounds. Under the new algorithm, the upper-bounds repair proceeds as follows:

Cell 4	Cell 5	Cell 6
$\mathtt{u} = 0.2$	$\mathtt{u} = 0.2$	$\mathtt{u} = 0.2$
$\mathtt{m} = -1$	$\mathtt{m} = -1$	$\mathtt{m} = -1$
$\mathtt{l} = -1$	$\mathtt{l} = -1$	$\mathtt{l} = -1$
Cell 1	Cell 2	Cell 3
$\mathtt{u} = 0$	$\mathtt{u} = 0$	$\mathtt{u} = 0$
$\mathtt{m} = 1$	$\mathtt{m} = 1$	$\mathtt{m} = 1$
$\mathtt{l} = 0$	$\mathtt{l} = 0$	$\mathtt{l} = 0$

$\Longrightarrow$

Cell 4	Cell 5	Cell 6
$\delta = 1/2 + 1/3$	$\delta = 1/2 + 1/3 + 1/2$	$\delta = 1/3 + 1/2$
Cell 1	Cell 2	Cell 3
$e = 1$	$e = 1$	$e = 1$

Cell 4 is one of Cell 1's *two* acceptor neighbors, and one of Cell 2's *three* acceptor neighbors, so it receives δ portions of 1/2 and 1/3 from those cells, respectively. Similarly, cell 5 receives portions from cells 1, 2, and 3, while cell 6 receives portions from cells 2 and 3. After the first iteration, we get values as follows:

Cell 4	Cell 5	Cell 6
$m \leftarrow m + \delta = -1/6$	$m \leftarrow m + \delta = 1/3$	$m \leftarrow m + \delta = -1/6$
Cell 1	Cell 2	Cell 3
$m \leftarrow u = 0$	$m \leftarrow u = 0$	$m \leftarrow u = 0$

which are in-bounds but are not symmetric.

5.2. Symmetry preservation

The loss of symmetry in this algorithm occurs because cells that should be equivalent, because they have the same values and same bounds, are treated differently according to whether or not they are near the boundary. In the above example, cells 1, 2, and 3 have the same values and same bounds, but cells 1 and 3, due to their proximity to the boundary, have only two neighboring acceptor cells, while cell 2 has three neighboring acceptor cells.

To preserve symmetry, we can double-count boundary cells in an appropriate way. Equivalently, we can introduce ghost cells with the property that a modification made to a ghost cell is later transferred to its corresponding real cell. Fig. 3 illustrates this process.

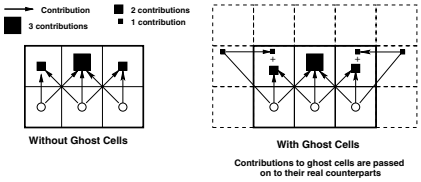


Fig. 3. Preservation of basic symmetry. In the absence of ghost cells, the repair algorithm transfers more mass from broken cells (open circles) to nearby internal cells (big black squares) than to nearby boundary cells (medium black squares). If we introduce ghost cells, then the acceptor cells on the boundary receive mass from broken cells both directly, and indirectly through their ghost copy (small black squares).

Consider the previous example, now with ghost cells (denoted by primes). Initially, we have:

Cell 4'	Cell 4'	Cell 5'	Cell 6'	Cell 6'
Cell 4'	**Cell 4** $\mathtt{u} = 0.2$ $\mathtt{m} = -1$ $\mathtt{l} = -1$	**Cell 5** $\mathtt{u} = 0.2$ $\mathtt{m} = -1$ $\mathtt{l} = -1$	**Cell 6** $\mathtt{u} = 0.2$ $\mathtt{m} = -1$ $\mathtt{l} = -1$	Cell 6'
Cell 1'	**Cell 1** $\mathtt{u} = 0$ $\mathtt{m} = 1$ $\mathtt{l} = 0$	**Cell 2** $\mathtt{u} = 0$ $\mathtt{m} = 1$ $\mathtt{l} = 0$	**Cell 3** $\mathtt{u} = 0$ $\mathtt{m} = 1$ $\mathtt{l} = 0$	Cell 3'
Cell 1'	Cell 1'	Cell 2'	Cell 3'	Cell 3'

Cells 1, 2, and 3 now account for the ghost cells when they distribute portions of their excess mass:

Cell 4'	Cell 4'	Cell 5'	Cell 6'	Cell 6'
Cell 4' $\delta' = 1/3 \longrightarrow$	**Cell 4** $\delta = \delta'$ $+1/3$ $+1/3$	**Cell 5** $\delta = 1/3$ $+1/3$ $+1/3$	**Cell 6** $\delta = \delta'$ $+1/3$ $+1/3$	Cell 6' $\longleftarrow \delta' = 1/3$
Cell 1'	**Cell 1** $e = 1$	**Cell 2** $e = 1$	**Cell 3** $e = 1$	Cell 3'
Cell 1'	Cell 1'	Cell 2'	Cell 3'	Cell 3'

Each broken cell now has three acceptor cells. Consider cell 1. Its nonzero δ is spread over cells 4′, 4, and 5. Because cell 4′ is a ghost cell, its δ, which we call δ', is ultimately transferred to the real cell 4. So, cell 4 receives one contribution from cell 1, one contribution from cell 2, and one contribution from its ghost version, cell 4′. The final values show preservation of symmetry:

Cell 4	Cell 5	Cell 6
$\mathtt{m} \leftarrow \mathtt{m} + \delta = 0$	$\mathtt{m} \leftarrow \mathtt{m} + \delta = 0$	$\mathtt{m} \leftarrow \mathtt{m} + \delta = 0$
Cell 1	Cell 2	Cell 3
$\mathtt{m} \leftarrow \mathtt{u} = 0$	$\mathtt{m} \leftarrow \mathtt{u} = 0$	$\mathtt{m} \leftarrow \mathtt{u} = 0$

In practice, because the neighborhood size might not be fixed, it may be desirable to use an appropriate bookkeeping scheme near the boundaries, in lieu of using ghost cells. For example, we can reflect indices at the boundaries, i.e., treat the grid as a torus.

5.3. Properties and issues

This algorithm converges, is conservative and order-independent, and can preserve a 1D symmetry with the modifications outlines above. However, on parallel machines this algorithm (as well as the local order-dependent method) is slow. This is largely due to the neighborhood expansion needed by both algorithms. To define how far we need to look for mass/room, the algorithms expand the neighborhood of a cell, but there is no way to know, *a priori*, how many expansion steps are needed. On a parallel machine, this leads to excessive communication between processors if the expansion process reaches a border between processors. If ghost cells are used, a possible solution is to create thicker layers of ghost cells (as thick as we anticipate will be needed) along processor boundaries. Another solution is described next.

6. A mixed local/global order-independent repair

The global repair algorithm that was described earlier needs very little communication and can be used very efficiently in a parallel framework. Once every cell has been repaired for every processor, a single communication is performed to give the total discrepancy Δ, and the final update of the masses is made on each processor without additional communication. However, the global algorithm can violate causality unless most of the cells are within or close to their bounds. Therefore, we consider here an amalgamation of the local algorithm, which gives more physically meaningful results, and the global algorithm, which is more parallelizable.

6.1. Algorithm

The mixed local/global order-independent repair algorithm is based on the assumption that most of the out-of-bounds cells can be fixed locally (using only the immediate neighborhood) because they are due to very small distributions, and that, as a corollary, only a few cells need to find room/mass far away from their location.

The idea of this algorithm is to repair as many cells as possible with the local order-independent algorithm, and then if some of the cells are still out-of-bounds, to repair them with the global repair algorithm. That is, this method consists of the following two steps:

(1) *Local treatment*. For all out-of-bounds cells, try to repair with the Local order-independent symmetry-preserving algorithm, but without expanding any immediate neighborhoods. For example, if cell i has Δ units of excess mass, then check if some of its mass can be spread out in the $|N(i)|$ *closest* cells (recall that $|N(i)|$ is the number of cells in $N(i)$). If $0 < n \leqslant |N(i)|$ cells can accept A unit of mass in total, then give $\Delta a(j)/A$ to each of these cells, where $a(j)$ is the mass that can be accepted by cell j. If cell i still has excess mass, then leave it out of bounds. Iterate this process in order to converge to a situation where either every cell is repaired, or the remaining unrepaired cells cannot be repaired using their closest neighborhood. As in the local order-independent repair method, we try to correct the upper bounds first and then the lower bounds, or vice versa.
(2) *Global treatment*. For any remaining out-of-bounds cells, perform the global repair.

Our experimentation indicated that few iterations are needed to perform the local treatment. The global treatment finally fixes the remaining out-of-bounds cells, the number of which is presumably small, and which should be out of bounds only by small amounts.

6.2. Properties and issues

The mixed local/global repair algorithm is conservative, because each of its steps is conservative. Moreover, both the local and global treatments are order-independent and symmetry preserving (by using the method described in 5.2), and there is no particular difficulty with parallelization because there is no indefinite neighborhood expansion.

The local neighborhood used in the first step is user-dependent and can consist of 0, 1, or more layers of the neighborhood as defined in (1). In our simulations, we used the 1-layer neighborhood.

If any cells are still out-of-bounds after the first step, the global repair fixes them. The earlier argument stating that this method can violate causality still holds, but the effect is far less pronounced because very few cells will remain to be fixed after the initial local treatment, and the amounts by which they need to be fixed will be less. Therefore, the causality violation could be negligible. In the next section, we present an example of advection where the data show that most of the cells are indeed fixed by the local treatment.

7. Application to advection

As in [4], we compute a pure advection problem in 2D with the following equation:

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} = 0, \tag{16}$$

where ρ is the density and u the velocity.

Three revolutions of a cone are computed on a 100×100 quadrilateral mesh on the domain $\Omega = [0,2] \times [0,2]$ with a CENO type numerical scheme. In the continuum case, the cone is not disturbed after the rotations. In the discrete case, we can measure the impact of a given numerical scheme, including diffusion and oscillations, because after three revolutions the cone should be as close as possible to the original one. The original cone has a peak at 5 and a minimum value of 0 (see Fig. 4).

The density field ρ as a function of the radius r is defined as follows:

$$\rho(r) = \begin{cases} 20(0.2 - r) + 1, & \text{if } r \leqslant 0.2, \\ 1, & \text{if } 0.2 \leqslant r \leqslant 0.4, \\ 0, & \text{else.} \end{cases} \tag{17}$$

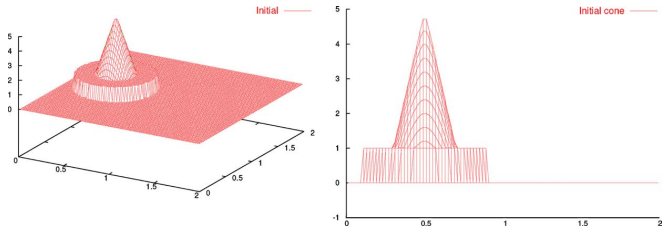


Fig. 4. Initial cone for the advection problem. This is the exact cone after three rotations in the continuum case. Left: 2D view. Right: 1D view plane x, ρ.

The center of the cone is shifted to $(0.5, 1)$, the boundary conditions are treated as walls (normal velocity equal to 0), and the velocity field is given by $u(x, y) = y - 1$, $v(x, y) = -x + 1$.

The intent at the moment is not to write a good advection scheme, but to show that a repair method can help *any* advection scheme to perform better. Indeed, in this case our advection scheme itself performs poorly on the input data because it creates unphysical oscillations that require repair.

In Fig. 5 we present the third revolution when no repair algorithm is used (top-left), and when the global (top-right), local order-independent (bottom-left), and mixed local–global (bottom right) repair methods are used. The figure is a 1D view along the x-axis. The local order-*dependent* repair method is not used because its order dependence is too serious an issue.

This problem is quite difficult for any repair method because so many cells must be repaired per iteration.

No repair. Some parasitic oscillations which generate negative values are present when no repair method is used. The maximum and minimum densities are 4.46 and −0.405, whereas the exact values are 5.0 and 0.0.

Global repair. No parasitic oscillations can be seen. The time spent to solve the problem is about 6 times larger than the time spent to solve the problem without any repair.

Order-independent and symmetry-preserving local repair. The time ratio is 7 and the maximum and minimum values are 4.12 and 0.0. No oscillations can be seen, and the shape of the cone is respected.

Mixed local/global repair. During this simulation, there are an average of 2000–2500 out-of-bounds cells per cycle. This is (∼20–25% of the total number of cells.) After the local step, the percent of cells still out-of-bounds is between 0.05% and 2%, meaning that most of the cells have been fixed locally. This was one of the assumptions to motivate the development of such a repair algorithm. The few remaining out-of-bounds cells are finally fixed by the global repair step. The time ratio is 7, and the minimum/maximum values are 0.0 and 4.16.

In Table 1 we gather the results of these methods: the time spent and the minimum/maximum density values.

This advection problem shows the ability of our repair methods to fix unphysical oscillations without destroying the shape of the cone, and to maintain reasonable accuracy at the maximum value of the cone even if the peak has been clipped. Recall that we specifically designed our advection scheme to produce very poor results; almost every cell needs repair every 4 time steps, and therefore, the repair stage constitutes a

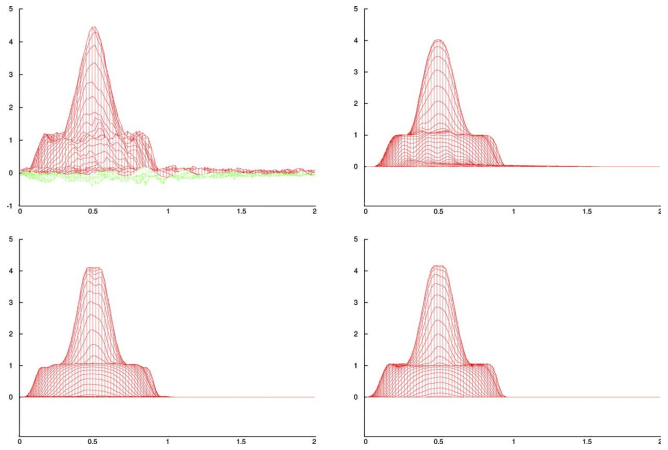


Fig. 5. Revolutions of a cone with different repair algorithms. Plane (x, ρ). Top left: without repair. Top right: with global repair. Bottom left: with order-independent local repair. Bottom right: with mixed local/global repair.

Table 1
Revolutions of a cone with four repair algorithms

Methods	Time ratio δ_T	Min/max values
No repair	1	−0.405/4.46
Global	6	0.0/4.03
Order-independent	7	0.0/4.12
Mixed local/global	7	0.0/4.16
Exact	–	0.0/5.0

Minimum and maximum values and δ_T, the ratio between the time spent to solve the problem with and without the repair method.

large percentage of the total run-time of the code. In a real application this will not be the case, and the time ratio of 1:7 will decrease dramatically. Our ratios reflect worst-case scenarios.

8. Application to hydrodynamics

We now apply the repair idea to some compressible flow problems in which remapping is required, either because the scheme is an Eulerian one of Lagrange-remap type, or because it is a Lagrangian scheme

needing rezoning. There are many ways to do the remapping, but, without going into details, we have chosen to use limited piecewise linear reconstruction on the original cells, and integration over the intersection of the old and new cells. In one dimension this integration is exact, while in two dimensions we use a quadrature as described in [6,5]. The variables remapped are mass, momentum, and total energy. Because a quadrature is used, the remapped density can be negative. Because the internal energy is obtained by subtracting the new kinetic energy from the total energy, this can negative, and frequently is.

For repair, we first repair density to satisfy its maximum principle. [5] provides a simple and easily satisfied condition, relating the bounds and the mesh, that implies the feasibility condition stated in the Introduction. We next repair the velocity components (defined as the ratio of momentum to repaired density) to satisfy their own maximum principle, maintaining conservation of momentum. A sufficient condition for the success of this step is stated in [4], and although that condition cannot be verified in this application, we have never observed a failure of velocity repair. The final step is to repair internal energy, defined as the remapped total energy minus the kinetic energy as obtained from the repaired density and velocity. It is shown in [4] that if the new total kinetic energy does not exceed the old kinetic energy, then internal energy can be repaired without violating local lower bounds, in particular, without being negative. Lower bound energy repair has not failed in our examples. In theory, local upper bounds on internal energy cannot *always* be satisfied. However, we attempt to satisfy them when possible.

8.1. Staggered grids

Staggered polygonal grids are sometimes used in ALE codes. In this scenario, fluid variables live in different places: density and specific internal energy at cell centers, and velocity at nodes. Density can also be given in subcells, where a subcell is a quadrilateral defined by joining a cell's center, one of its nodes, and the centers of the cell's edges that are linked to the node. With staggered grids, the repair algorithms require no special treatment because they make no assumption about the meaning (staggered or otherwise) of an underlying grid. Dealing with a mesh (for energy repair), a subcell mesh (for density repair), or a median mesh (for velocity repair), is irrelevant to the algorithms themselves.

8.2. Enforcement of boundary conditions

Different types of boundary conditions are used in Lagrangian numerical schemes. These include piston (nonzero velocity), wall (zero velocity), vacuum (zero pressure), and compression/expansion (nonzero pressure). Such boundary conditions (BCs) are implemented by enforcing the velocity of all the boundary nodes or by enforcing the pressure of all the boundary cells. For example, assume a nonzero boundary velocity is enforced during an ALE calculation. After the rezone and remap parts occur, velocities on the boundary nodes will have changed and will probably violate the boundary conditions. If this happens, a convenient way to re-enforce the BC is to repair the nodal velocities using upper and lower bounds equal to the required boundary velocity. Then, no momentum is lost (because the repair is conservative) and the required velocity at the boundary is satisfied.

8.3. Numerical results

We now present some numerical tests in a compressible hydrodynamics framework where the use of a repair method can improve the results.

To produce the following results we used an ALE code called ALE INC(ubator) that is designed for general polygonal grids [7]. This code is split into a Lagrangian scheme, a rezoning phase, and a remapping phase. In the remapping phase, we avoid the computation of polygon intersections because it can be expensive in 2D, and unaffordable or simply infeasible in 3D. Mostly because of this approximation, the

remapping can produce out-of-bounds values, and therefore a repair process must be used to ensure at least the positivity of density and energy. For these numerical experiments, a full repair is performed. That is, density, velocity and specific internal energy are all corrected.

The code can be used in the Eulerian regime (as Lagrange + Remap), in the Lagrangian regime, or in the ALE regime (with Rezone and Remap phases). Moreover, it can be run in 1D or 2D on general polygonal grids.

Sod Riemann problem. This is a very simple 1D Riemann problem for an ideal gas with $\gamma = 1.4$. The data are given by a left state $(\rho, u, v, p)_L = (1, 0, 0, 1)$ and a right state $(\rho, u, v, p)_R = (0.125, 0, 0, 0.1)$. The discontinuity is located at $X = 0.5$ on a domain $[0{:}1] \times [0{:}y_{\max}]$ where $y_{\max}$ is defined so that the initial cells are squares. Most numerical schemes produce decent results on this problem. The first run of the code is made without conservative repair; if a density or an energy is negative, we simply clip it to zero. The second run is performed with a conservative repair method for density, velocity, and specific internal energy.

The results given with and without conservative repair are plotted in Fig. 6 for a perfect quadrilateral 101×11 mesh at time $t = 0.25$ in the Eulerian regime. Clearly, the results without conservative repair fit the exact solution. Using a repair method does not break the behavior and the results are almost identical. Also, in both cases we checked that 1D symmetry is preserved.

Le Blanc Riemann problem. This is a very strong 1D shock tube; the jump in pressure being 10^9 and the jump in density 10^3. The data are given by a left state $(\rho, u, v, p)_L = (1, 0, 0, \frac{2}{3}10^{-1})$ and a right state $(\rho, u, v, p)_R = (10^{-3}, 0, 0, \frac{2}{3}10^{-10})$ with $\gamma = 5/3$ and a discontinuity located at $X = 3$ on $[0{:}9] \times [0{:}y_{\max}]$ where $y_{\max}$ is defined so that the initial cells are squares.

Most numerical schemes produce an overshoot after the contact discontinuity in specific internal energy, and a bad approximation of the shock speed. The results given with and without conservative repair are plotted in Fig. 7.

A perfect quadrilateral mesh is used with our ALE code, which is run in the Eulerian regime (as Lagrange + Remap). Without a conservative repair method, and for a 601×3 mesh, the conservation in total energy is slightly violated $[(E_{\text{final}} - E_{\text{initial}})/E_{\text{initial}}) \simeq 10^{-6}]$. This violation occurs because negative internal energies are created, and must be cut to 0 for the code to run properly. With any conservative repair method, on the other hand, conservation is preserved to machine error. So, in this problem, the use of a conservative repair method preserves the positivity, reduces the overshoot, and stabilizes the profile in the

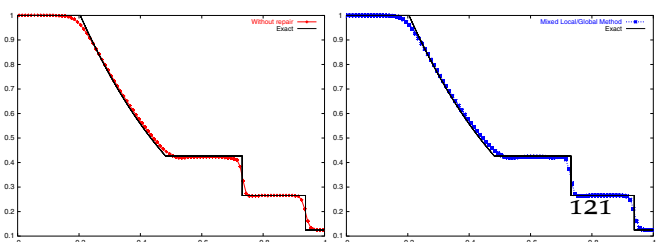


Fig. 6. Sod shock tube – density at $t = 0.25$ without (left) and with (right) a conservative repair method on a quadrilateral 101×11 mesh – Eulerian regime versus the exact solution.

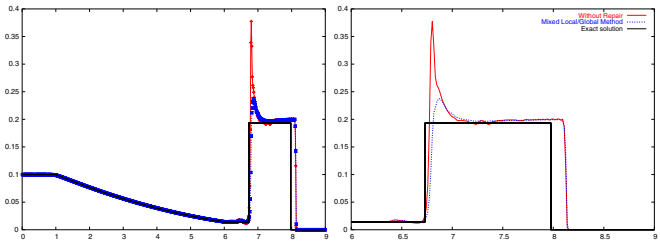


Fig. 7. Le Blanc shock tube. Specific internal energy at $t = 6.0$ with and without conservative repair, on a 601×3 mesh. Entire domain (left) and zoom (right).

plateau as shown in the zoom in Fig. 7. In this run the mixed local/global repair method is used, but such general behavior can be seen with all of our repair methods.

Interaction of blast waves. On $\Omega = [-1{:}1] \times [0{:}1]$ we initiate two half-disks of radius 0.1 centered at $X_1 = -(0.4{,}0)$ and $X_2 = (0.4{,}0)$. The perfect equation of state is used with $\gamma = 1.4$. The density is constant and equal to 1, and the velocity is zero everywhere. The internal energies are $\epsilon_1 = 30$, $\epsilon_2 = 60$ in the disks and 0 elsewhere in the domain. This high energy generates two non-symmetric cylindrical blast waves which interact. At the same time they reflect onto the walls. The final time is $t = 0.7$ and the code ran in its ALE–10 regime (rezone and remap every 10 Lagrangian cycles) on a quadrilateral mesh of 4950 nodes and 4802 cells, which has been adapted to fit the disks.

The first remappings create negative specific internal energies. Therefore, without special treatment, the code cannot go further. On the other hand, with a conservative repair method the negative energies are removed, and the conservation of mass, momentum and total energy are preserved. Our code, used with the mixed local/global repair method, produces the meshes and densities plotted in Fig. 8.

Sedov blast wave. The computational domain is one quarter of a circular disk with a radius of $r_{\max} = 1.2$. A polygonal mesh is constructed in the computational domain using a Voronoi diagram for the set of points defined as follows:

$$x_{i,j} = r_j \sin(\theta_{i,j})\ y_{i,j} = r_j \sin(\theta_{i,j}); \qquad j = 1, \ldots, J; \quad i = 1, \ldots, I(j),$$

where

$$r_j = r_{\max} \cdot \frac{j-1}{J}, \quad I(j) = \text{round}\left((j-1)\frac{\pi}{2}\right), \quad \theta_{i,j} = \frac{i-1}{I(j)} \cdot \frac{\pi}{2}, \quad J = 31$$

and the function round(x) returns the closest integer to x. According to these formulas, on each circle of radius r_j the points are distributed so that the distance between adjacent points along the circle is approximately equal to $\Delta r = r_{\max}/(J-1)$. The total number of points is 775, and there is exactly one Voronoi cell corresponding to each point. The mesh consists of a mixture of convex quadrilaterals, pentagons and hexagons, with a total of 1325 vertices; see Fig. 9. The disk is filled at $t = 0$ with an ideal gas ($\gamma = 1.4$) at rest whose density is uniformly equal to 1. The specific internal energy is zero except in the pentagonal cell c in contact with the origin, where $\epsilon(c) = E/m(c) = E/V(c)$. $V(c)$ is the volume of cell c and E is the total energy

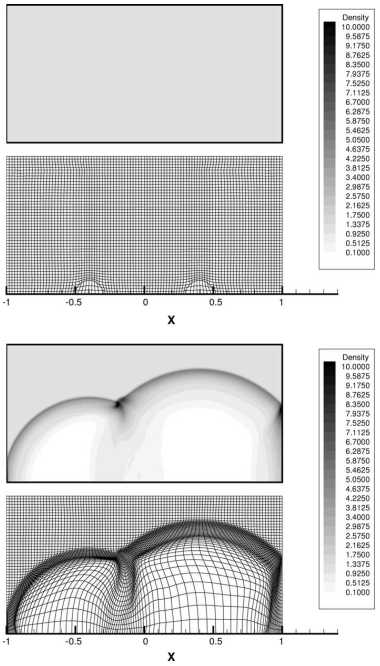


Fig. 8. Interaction of cylindrical blast waves on a quadrilateral mesh. ALE–10 regime. Mesh and density contours (exponential scale) at $t = 0.1$ and $t = 0.7$.

in the system, chosen as 0.244816. This choice of E leads to a diverging shock wave that, at $t = 1.0$, should be at radius 1.0. The peak in density should be equal to 6.

The code is used in its ALE–10 regime. When no repair is performed in this test case, the code stops due to the creation of negative internal energy after the fourth remapping. On the other hand, the use of a repair method fixes the parasitic negative values and allows us to observe good results. The maximum density with the mixed local/global repair method is 5.62.

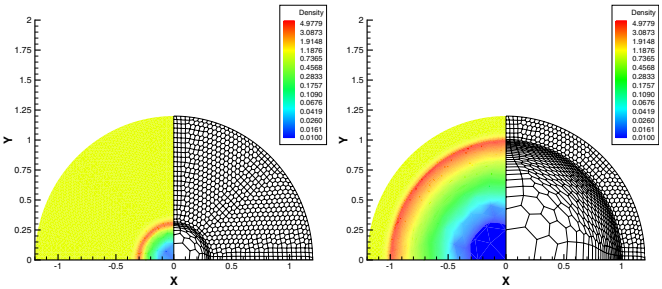


Fig. 9. Sedov blast wave on a polygonal mesh (1325 nodes and 775 cells). ALE–10 regime. Mesh and density contours (exponential scale) at $t = 0.1$ and $t = 1.0$.

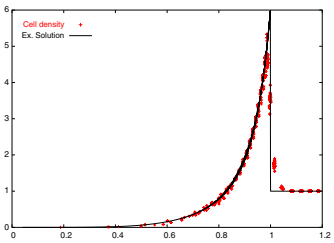


Fig. 10. Sedov blast wave on a polygonal mesh (1325 nodes and 775 cells). ALE–10 regime. Density at $t = 1.0$ (cell-based value as a function of the radius).

In Fig. 9 the density contours are shown in exponential scale, and in Fig. 10 the cell density is shown as a function of the cell radius at $t = 1.0$, versus the exact solution.

9. Conclusion

In this paper we have reviewed and applied several conservative repair methods that can be used in situations where variables must stay between predefined bounds while respecting conservation. Such situations occur often, in hydrodynamics for example, when the density or the specific internal energy becomes negative due to remapping. Such unphysical situations must be cured, but setting negative values to small positive numbers is not acceptable from the point of view of conservation.

Of the methods presented here, we believe that the mixed local/global repair scheme best meets the requirements of locality and efficiency. This method was applied to an advection example and to test cases in an ALE hydrodynamics framework, where the use of a conservative repair algorithm allowed us to:

- preserve the accuracy of the underlying method, as in the Sod Riemann problem;
- stabilize and improve bad profiles, as in the Le Blanc Riemann problem;
- maintain physical and reasonable results, as in the blast wave interaction problem and the Sedov problem.

Acknowledgements

The authors thank M. Shashkov for fruitful discussions. This work was performed under the auspices of the US Department of Energy at Los Alamos National Laboratory, under contract W-7405-ENG-36. The authors acknowledge the partial support of the DOE/ASCR Program in the Applied Mathematical Sciences and the Laboratory Directed Research and Development program (LDRD). The authors also acknowledge the partial support of DOE's Advanced Simulation and Computing (ASC) program.

References

[1] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, Journal of Computational Physics 87 (1990) 171–200.
[2] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, Journal of Computational Physics 160 (2000) 318–335.
[3] A. Oliveira, A.B. Fortunato, Toward an oscillation-free, mass conservative, Eulerian–Lagrangian transport model, Journal of Computational Physics 183 (2002) 142–164.
[4] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, Journal of Computational Physics 198 (2004) 265–277.
[5] M. Kucharich, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping method, Journal of Computational Physics 188 (2003) 462–471.
[6] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grid, Journal of Computational Physics 184 (2003) 266–298.
[7] R. Loubere, M.J. Shashkov. A subcell remapping method on staggered polygonal grids for Arbitrary–Lagrangian–Eulerian methods. Journal of Computational Physics (2004) (accepted).

In paper [9] entitled *Convergence and Sensitivity Analysis of Repair Algorithms in 1D* B. Després made most of the the convergence analysis of some simple Repair algorithms and I joint his effort to numerically retrieve his convergence and sensitivity results. In this paper we limit the theoretical study to transport equation in 1D and to some simple remapping strategy in 1D. In order to prove the main convergence result we introduced a box of size p in which the distribution of the excess of mass is performed. Numerical results are proposed for the Lax-Wendroff (LW) scheme plus Repair and the DownWind (DW) scheme plus Repair. The LW intends to be representative of high-order prediction schemes. The DW scheme intends to be representative of highly anti-dissipative prediction schemes. In a specific section we study the gas dynamics equations in 1D with a Lagrange+Remap code. This code is built on two components : a staggered Lagrangian scheme and a Remap strategy which may need repair. The Sod and Le Blanc shock tubes and the blastwave of Colella-Woodward are tested. It seems that the size of the box (parameter p) is not necessarily an important parameter if one uses a high-order prediction scheme as the Lax-Wendroff scheme for non oscillating computations. But with a more anti- dissipative prediction scheme as the Downwind scheme, the results can vary with p. When p is too large the numerical solution may not be correct. It gives some indication that it is much preferable to restrict ourselves to local Repair (i.e with a local redistribution of the mass).

An important feature of the Repair paradigm is the simplicity and versatility in any dimension. Moreover the repair process is independent of the kind of mesh used ; cell-centered values or nodal values can be repaired the same way. We only need the notion of neighborhood to define the bounds and to redistribute the amount of conservative variable. Therefore any repair algorithm is suitable for staggered formulation where physical variables are not defined at the same location.

This paper is reproduced in the following pages.

Convergence and Sensitivity Analysis of Repair Algorithms in 1D

Bruno Després†
†*Commissariat à l'Energie Atomique CEA, BP12, 91680, Bruyères-Le-Châtel, France*
bruno.despres@cea.fr

Raphaël Loubère*
Laboratoire Math. pour l'Industrie et la Physique (MIP) Bureau 214 University Toulouse 31062 Toulouse Cedex
loubere@mip.ups-tlse.fr

Abstract

We prove the convergence of some repair algorithms for linear advection in dimension one. The convergence depends on the size of the box where the distribution of the mass excess is performed. Various numerical examples illustrate the theoretical results. Applications to gas dynamics in dimension one is also discussed.

Key words : Finite Volume Schemes, Repair algorithms, TVD schemes, Gas Dynamics.

1 Introduction

The aim of this work is the convergence analysis of some very simple Repair Algorithms. To our knowledge such a proof has never been given, even for simple problems. The Repair algorithms we consider are very much in the spirit of what was proposed by M. Shashkov and B. Wendroff in [11]. The underlying idea of Repair methods is related to the fact that conservative remapping methods (present in Arbitrary-Lagrangian-Eulerian (ALE) framework for example) may not necessarily preserve a maximum principle: *no new extrema should be generated during the remapping process*. If so a Repair method can remove these new maxima (resp. minima) by distributing (resp. taking) the amount of the associated conservative variable to (resp. from) the neighborhood.
In this paper we limit the theoretical study to transport equation in 1D and to some simple remapping strategy in 1D. However we hope that some of the conclusions of the study can give insights even for multidimensional repair algorithms for systems of PDE's, such as compressible gas dynamics.
The outline of this paper is as follows. In the second section, we describe the repair algorithm for the linear transport. In section three we adapt the repair algorithm for the remapping process. The numerical results on 1D linear transport are gathered in the fourth section, whereas the fifth section presents some numerical results for 1D gas dynamics. Finally the conclusion and the plan for future developments is presented in the last section.

2 Repair algorithm for linear transport

Let us consider the equation of transport in 1D, with constant velocity

$$\partial_t u + a\partial_x u = 0, \quad a > 0. \tag{1}$$

The initial condition is $u(0,x) = u_0(x)$ where $u_0 \in L^1(\mathbf{R}) \cap BV(\mathbf{R})$. Let us consider a mesh, that is uniform even if our result extends to non uniform mesh. The mesh size is denoted by Δx. The time step is denoted by Δt. Thus u_j^n stands for the numerical solution in cell j at time step n.

The repair strategy amounts to: 1) compute a prediction of the numerical solution at time step $n+1$ using a "reasonable" and "local" scheme, as instance this scheme can be a high order non monotone scheme or a highly anti-dissipative scheme, 2) check if the new value satisfies a local maximum principle, 3) if the new value does not satisfy the local maximum principle, then repair it. Repairing means changing the value of the unknown for 3) to be fulfilled. A difficulty is that one wants the total mass to be preserved. So we need to describe in details how to redistribute the mass of the repaired quantity, such as the total mass is preserved.

This family of algorithms can be local if one redistributes the mass in a local box around the cell that needs to be repaired, or global if one redistributes the mass in the entire domain. It has been a debate since the early time of the Repair Algorithms to decide whether local repairing is better or not than global repairing. On one hand, global repairing is a more simple algorithm than local repairing. On the other hand our analysis implies that we can not prove the convergence of a global repair process. This is a theoretical indication that global repair process can be dangerous in some cases. Moreover a global repair process breaks the causality by instantaneously spreading mass all over the domain. The numerical experiments show it is indeed the case, in particular when the prediction step or the underlying physics is oscillating.

Let us describe in detail the repair algorithm that we analyze in this work.

2.1 The prediction scheme

First, one computes the new value of the unknown using the finite volume and conservative scheme

$$\boxed{\frac{\overline{u}_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n + c_{j+\frac12}^n - u_{j-1}^n - c_{j-\frac12}^n}{\Delta x} = 0,} \tag{2}$$

where u_j^n is the mean value of $u(x)$ in cell j: $[x_{x-\frac12}; x_{x+\frac12}]$.
The Courant or CFL number is less than one ($a > 0$)

$$\nu = a\frac{\Delta t}{\Delta x} \leq 1. \tag{3}$$

The scheme is *a priori* different from the upwind scheme. Thus all the difference between the upwind scheme and the scheme used is embedded in the definition of the correction flux $ac_{j+\frac12}^n$ for all j. This correction flux can be either a linear or a non linear function of (u_j^n). The only assumption is

$$\exists C_1 > 0,\ \exists k \in \mathbf{N}, \quad |c_{j+\frac12}^n| \leq C_1 \sum_{j-k\leq q\leq j+k} |u_q - u_{q-1}|. \tag{4}$$

The hypothesis (4) essentially means that the flux is defined as the upwind flux plus a correction. Of course the correction is zero if the numerical profile is flat (that is if $u_q - u_{q-1} \equiv 0$ in a neighborhood of cell j) : (4) is compatible with such a principle. The hypothesis is true for the Lax-Wendroff scheme as instance, moreover all non linear TVD algorithms satisfy (4).

2.2 Correction step

The spirit of this repair algorithm is to compare $\overline{u}_j^{n+1}$ with

$$M_j^n = \max(u_j^n, u_{j-1}^n) \text{ and } m_j^n = \min(u_j^n, u_{j-1}^n), \tag{5}$$

that is one checks if $m_j^n \leq \overline{u}_j^{n+1} \leq M_j^n$ is true or not. Suppose $\overline{u}_j^{n+1} > M_j^n$, then one has to modify the value of $\overline{u}_j^{n+1}$ and redistribute the mass "around". In the convergence analysis of the method, we discovered that it is better at the theoretical level not to redistribute the mass globally but locally at least in a box of size $p \in \mathbf{N}$ around the current cell. This is why we have introduced a new step in the repair algorithm first proposed in [11] to be able to ensure that the redistribution of mass can be made in the box of size p. Since p is a parameter of the method, one recovers the global repair by setting $p \approx +\infty$.

So let us define boxes of size p. Each box is the collection of cells j such that $rp \leq j \leq (r+1)p-1$ where $r \in \mathbf{Z}$. The mathematical definition of these boxes B_r is

$$B_r = \{j;\ rp \leq j \leq (r+1)p-1\}, \quad r \in \mathbf{Z}. \tag{6}$$

However it is also possible to use boxes of different sizes, provided the size is smaller than the predefined maximal box's size p. It is also possible to use moving boxes, that is the starting point of each box is different from one time step to the other. For the simplicity of the mathematical exposure we use only (6).

To make the correction we first need to compute

$$b_r^M = \left(\sum_{j\in B_r} \left(u_j^n - \nu(u_j^n - u_{j-1}^n) - M_j^n\right)\right) - \nu(c_{(r+1)p-\frac12}^n - c_{rp-\frac12}^n), \tag{7}$$

$$b_r^m = \left(\sum_{j\in B_r} \left(u_j^n - \nu(u_j^n - u_{j-1}^n) - m_j^n\right)\right) - \nu(c_{(r+1)p-\frac12}^n - c_{rp-\frac12}^n). \tag{8}$$

Most presumably $b_r^M \leq 0$ (resp. $b_r^m \geq 0$), since this is the result of a comparison between $\sum_{j\in B_r}\left(u_j^n - \nu(u_j^n - u_{j-1}^n) - M_j^n\right) \leq 0$ (resp. $\sum_{j\in B_r}\left(u_j^n - \nu(u_j^n - u_{j-1}^n) - m_j^n\right) \geq 0$) and $\nu(c_{(r+1)p-\frac12}^n - c_{rp-\frac12}^n)$. Moreover if p is large enough and ν small enough (i.e. the time step is small) then $b_r^M \leq 0$ (resp. $b_r^m \leq 0$) is probably stating true. The correction is here to ensure that $b_r^M \leq 0$ and $b_r^m \geq 0$ are always satisfied. The idea being that if these inequalities are not satisfied then we multiply the value of the fluxes by a small number such that $b_r^M \leq 0$ and $b_r^m \geq 0$ are fulfilled.

So we define

$$d_{j-\frac12}^n = \mu_{j-\frac12}^n c_{j-\frac12}^n \text{ where } \mu_{j-\frac12}^n \in [0,1]. \tag{9}$$

The coefficient $\mu_{j-\frac12}^n$ has to be computed to give a corrected value of the flux. The constraint $\mu_{j-\frac12}^n \in [0,1]$ appears natural from the consistency point of view. We expect that the definition of these $\mu_{j-\frac12}^n$ will be the closest as possible to 1, so that the corrected flux $d_{j-\frac12}^n$ is almost equal to the flux of the prediction scheme. We need to check

$$\tilde{b}_r^M - \nu(\mu_{(r+1)p-\frac12}^n c_{(r+1)p-\frac12}^n - \mu_{rp-\frac12}^n c_{rp-\frac12}^n) \leq 0, \tag{10}$$

$$\tilde{b}_r^m - \nu(\mu_{(r+1)p-\frac12}^n c_{(r+1)p-\frac12}^n - \mu_{rp-\frac12}^n c_{rp-\frac12}^n) \geq 0, \tag{11}$$

where by definition $\tilde{b}_r^M = \sum_{j\in B_r}\left(u_j^n - \nu(u_j^n - u_{j-1}^n) - M_j^n\right)$ is not positive and $\tilde{b}_r^m = \sum_{j\in B_r}\left(u_j^n - \nu(u_j^n - u_{j-1}^n) - m_j^n\right)$ is not negative. One feasible strategy can be derived as:

Analysis of (10)

if $c^n_{(r+1)p-\frac{1}{2}} \geq 0$ **and** $c^n_{rp-\frac{1}{2}} \geq 0$ **:** Then (10) is true once $\nu(\mu^n_{rp-\frac{1}{2}} c^n_{rp-\frac{1}{2}}) \leq -\tilde{b}^M_r$. Thus we define $\varphi^{1,-}_r, \varphi^{1,+}_r$ such that

$$\mu^n_{rp-\frac{1}{2}} \leq \varphi^{1,-}_r = \frac{-\tilde{b}^M_r}{c^n_{rp-\frac{1}{2}}} \text{ and } \varphi^{1,+}_r = +\infty. \tag{12}$$

if $c^n_{(r+1)p-\frac{1}{2}} < 0$ **and** $c^n_{rp-\frac{1}{2}} < 0$ **:** Then (10) is true once $\nu(-\mu^n_{(r+1)p-\frac{1}{2}} c^n_{(r+1)p-\frac{1}{2}}) \leq -\tilde{b}^M_r$. Thus we define $\varphi^{2,-}_r, \varphi^{2,+}_r$ such that

$$\varphi^{2,-}_r = +\infty \text{ and } \mu^n_{(r+1)p-\frac{1}{2}} \leq \varphi^{2,+}_r = \frac{-\tilde{b}^M_r}{-c^n_{(r+1)p-\frac{1}{2}}}. \tag{13}$$

if $c^n_{(r+1)p-\frac{1}{2}} \geq 0$ **and** $c^n_{rp-\frac{1}{2}} < 0$ **:** Then (10) is true without condition. Thus $\varphi^{3,-}_r = \varphi^{3,+}_r = +\infty$

if $c^n_{(r+1)p-\frac{1}{2}} < 0$ **and** $c^n_{rp-\frac{1}{2}} \geq 0$ **:** Then it is not possible to simplify the inequality (10). Thus we impose $\varphi^{4,-}_r = \varphi^{4,+}_r$ and

$$\mu^n_{(r+1)p-\frac{1}{2}},\ \mu^n_{rp-\frac{1}{2}} \leq \varphi^{4,-}_r = \frac{-\tilde{b}^M_r}{c^n_{rp-\frac{1}{2}} - c^n_{(r+1)p-\frac{1}{2}}} \tag{14}$$

Analysis of (11)

if $c^n_{(r+1)p-\frac{1}{2}} \geq 0$ **and** $c^n_{rp-\frac{1}{2}} \geq 0$ **:** Then (11) is true once $\nu(\mu^n_{(r+1)p-\frac{1}{2}} c^n_{(r+1)p-\frac{1}{2}}) \leq \tilde{b}^m_r$. Thus we impose that

$$\psi^{1,-}_r = +\infty \text{ and } \mu^n_{(r+1)p-\frac{1}{2}} \leq \psi^{1,+}_r = \frac{\tilde{b}^m_r}{c^n_{(r+1)p-\frac{1}{2}}}. \tag{15}$$

if $c^n_{(r+1)p-\frac{1}{2}} < 0$ **and** $c^n_{rp-\frac{1}{2}} < 0$ **:** Then (11) is true once $\nu(-\mu^n_{rp-\frac{1}{2}} c^n_{rp-\frac{1}{2}}) \leq \tilde{b}^m_r$. Thus we impose that

$$\mu^n_{rp-\frac{1}{2}} \leq \psi^{2,-}_r = \frac{\tilde{b}^m_r}{-c^n_{rp-\frac{1}{2}}} \text{ and } \psi^{2,+}_r = +\infty. \tag{16}$$

if $c^n_{(r+1)p-\frac{1}{2}} \geq 0$ **and** $c^n_{rp-\frac{1}{2}} < 0$ **:** Then it is not possible to simplify the inequality (11). Thus we impose $\psi^{3,-}_r = \psi^{3,+}_r$ that

$$\mu^n_{(r+1)p-\frac{1}{2}},\ \mu^n_{rp-\frac{1}{2}} \leq \psi^{3,-}_r = \frac{-\tilde{b}^m_r}{-c^n_{(r+1)p-\frac{1}{2}} + c^n_{rp-\frac{1}{2}}} \tag{17}$$

if $c^n_{(r+1)p-\frac{1}{2}} < 0$ **and** $c^n_{rp-\frac{1}{2}} \geq 0$ **:** Then (11) is true without condition. Thus $\psi^{4,-}_r = \psi^{4,+}_r = +\infty$

Let us consider each of the cases considered in inequalities (12) to (17). We gather the restrictions it imposes for all $\mu^n_{rp-\frac{1}{2}}$. The mathematical definition of the correction algorithm is the following.

Definition 2.1 Let us define the corrected fluxes at the boundaries of the boxes

$$d^n_{(r+1)p-\frac{1}{2}} = \min\left(1, \min_l \varphi^{l,+}_r, \min_l \psi^{l,+}_r, \min_l \varphi^{l,-}_{r+1}, \min_l \psi^{l,-}_{r+1}\right) \times c^n_{(r+1)p-\frac{1}{2}}. \tag{18}$$

Inside the boxes we do not correct, that is

$$d^n_{j-\frac{1}{2}} = c^n_{j-\frac{1}{2}}, \quad \forall j \neq rp. \tag{19}$$

The next step consists in the computation of the new prediction $\hat{u}^{n+1}_j$ with the corrected flux:

$$\frac{\hat{u}^{n+1}_j - u^n_j}{\Delta t} + a\frac{u^n_j + d^n_{j+\frac{1}{2}} - u^n_{j-1} - d^n_{j-\frac{1}{2}}}{\Delta x} = 0. \tag{20}$$

Lemma 2.2 One has the inequalities after the correction step

$$\left(\sum_{j\in B_r} \left(u^n_j - \nu(u^n_j - u^n_{j-1}) - M^n_j\right)\right) - \nu\left(d^n_{(r+1)p-\frac{1}{2}} - d^n_{rp-\frac{1}{2}}\right) \leq 0 \tag{21}$$

$$\left(\sum_{j\in B_r} \left(u^n_j - \nu(u^n_j - u^n_{j-1}) - m^n_j\right)\right) - \nu\left(d^n_{(r+1)p-\frac{1}{2}} - d^n_{rp-\frac{1}{2}}\right) \geq 0. \tag{22}$$

The proof is performed by considering all cases (12-17) separately.

Inequalities (21-22) will be crutial in the analysis of the repairing procedure, essentialy inequality (31) in subsection 2.4.

2.3 Repairing

As already mentioned, to repair a value means successively to compare with the local maximum and minimum, to truncate if needed, then to redistribute the excess of mass on all surrounding cells. Using mathematical notations, one gets

$$\begin{cases} \text{if } \hat{u}^{n+1}_j > M^n_j & \text{then } \mathbf{u}^{n+1}_j = M^n_j & \text{and } \Delta m^n_j = \hat{u}^{n+1}_j - M^n_j > 0, \\ \text{if } \hat{u}^{n+1}_j < m^n_j & \text{then } \mathbf{u}^{n+1}_j = m^n_j & \text{and } \Delta m^n_j = \hat{u}^{n+1}_j - m^n_j < 0, \\ \text{else} & \text{then } \mathbf{u}^{n+1}_j = \hat{u}^{n+1}_j & \text{and } \Delta m^n_j = 0. \end{cases} \tag{23}$$

The total mass in box B_r of the new unknown $\mathbf{u}^{n+1}_j$ may as well be different from the correct mass, so one defines the default of mass as:

$$\Delta M^n_r = \sum_{rp\leq j\leq (r+1)p-1} \Delta m^n_j. \tag{24}$$

This default of mass may be positive or negative. So we need to redistribute it on the box to get at least a conservative algorithm. Following [11], we consider

$$\begin{cases} \text{if } \Delta M^n_r > 0 & \text{then } u^{n+1}_j = \mathbf{u}^{n+1}_j + \lambda^n_r (M^n_j - \mathbf{u}^{n+1}_j), \\ \text{if } \Delta M^n_r < 0 & \text{then } u^{n+1}_j = \mathbf{u}^{n+1}_j + \lambda^n_r (m^n_j - \mathbf{u}^{n+1}_j), \\ \text{if } \Delta M^n_r = 0 & \text{then } u^{n+1}_j = \mathbf{u}^{n+1}_j, \end{cases} \tag{25}$$

where the coefficient λ^n_r is set to

$$\begin{cases} \text{if } \Delta M^n_r > 0 & \lambda^n_r = \frac{\Delta M^n_r}{\sum_{rp\leq j\leq (r+1)p-1} (M^n_j - \mathbf{u}^{n+1}_j)}, \\ \text{if } \Delta M^n_r < 0 & \lambda^n_r = \frac{\Delta M^n_r}{\sum_{rp\leq j\leq (r+1)p-1} (m^n_j - \mathbf{u}^{n+1}_j)}, \\ \text{if } \Delta M^n_r = 0 & \lambda^n_r = 0. \end{cases} \tag{26}$$

The repair algorithm that we analyze in this paper consists of equations (2) to (26).

2.4 Properties

Before proving our main convergence theorem, we state the stability lemma

Lemma 2.3 Whatever the value of $p \in \mathbf{N}^\star$ is, for all time step n, the repair algorithm is such that the total mass is preserved (conservation)

$$\sum_j u^{n+1}_j = \sum_j u^n_j, \tag{27}$$

the maxima and minima are respected (maximum principle)

$$m^n_j \leq u^{n+1}_j \leq M^n_j, \quad \forall j, \tag{28}$$

and the total variation is diminishing (TVD)

$$\sum_j |u^{n+1}_j - u^{n+1}_{j-1}| \leq \sum_j |u^n_j - u^n_{j-1}|. \tag{29}$$

Remark: One notices that the scheme is TVD because of (29). **Proof** First one has

$$\sum_{j\in B_r} \hat{u}^{n+1}_j = \left(\sum_{j\in B_r} (u^n_j - \nu(u^n_j - u^n_{j-1}))\right) - \nu\left(d^n_{(r+1)p-\frac{1}{2}} - d^n_{rp-\frac{1}{2}}\right).$$

Since by construction $\mathbf{u}^{n+1}_j = \hat{u}^{n+1}_j - \Delta m^n_j$, then we get the relation

$$\Delta M^n_r + \sum_{j\in B_r} \mathbf{u}^{n+1}_j = \left(\sum_{j\in B_r} (u^n_j - \nu(u^n_j - u^n_{j-1}))\right) - \nu\left(d^n_{(r+1)p-\frac{1}{2}} - d^n_{rp-\frac{1}{2}}\right). \tag{30}$$

Assume for instance that $\Delta M^n_r > 0$ (the other cases are easily deduced by mimicking this one). Then the next stage of the algorithm consists in the computation of λ^n_r. The key point is to prove that $0 \leq \lambda^n_r \leq 1$ and the property follows. One has

$$\begin{aligned} &\Delta M^n_r - \textstyle\sum_{j\in B_r} \left(M^n_j - \mathbf{u}^{n+1}_j\right) \\ &= \left(\textstyle\sum_{j\in B_r} (u^n_j - \nu(u^n_j - u^n_{j-1}) - M^n_j)\right) - \nu\left(d^n_{(r+1)p-\frac{1}{2}} - d^n_{rp-\frac{1}{2}}\right), \end{aligned}$$

and clearly

$$\Delta M^n_r - \sum_{j\in B_r} \left(M^n_j - \mathbf{u}^{n+1}_j\right) \leq 0, \tag{31}$$

due to the correction step (21)-(22). Then $\Delta M^n_r \leq \sum_{j\in B_r} \left(M^n_j - \mathbf{u}^{n+1}_j\right)$ that is

$$0 \leq \lambda^n_r = \frac{\Delta M^n_r}{\sum_{j\in B_r} M^n_j - \mathbf{u}^{n+1}_j} \leq 1.$$

And finally

$$m^n_j \leq u^{n+1}_j = (1-\lambda^n_r)\mathbf{u}^{n+1}_j + \lambda^n_r M^n_j \leq M^n_j,$$

since $m_j^n \le \mathbf{u}_j^{n+1} \le M_j^n$ by construction. Thus (28) is proved for $\Delta M_r^n > 0$. (The other case is proved using the same method.) It implies (29).
It remains to prove (27). The definition of λ_r^n associated with equation (30) yields to

$$\begin{aligned}\sum_{j\in B_r} u_j^{n+1} &= \sum_{j\in B_r} \mathbf{u}_j^{n+1} + \lambda_r^n \sum_{j\in B_r}\left(M_j^n - \mathbf{u}_j^{n+1}\right)\\ &= \sum_{j\in B_r} \mathbf{u}_j^{n+1} + \Delta M_r^n = \sum_{j\in B_r}(u_j^n - \nu(u_j^n - u_{j-1}^n)) - \nu(d^n_{(r+1)p-\frac12} - d^n_{rp-\frac12}).\end{aligned}$$

Summing up with respect to r gives (27) and completes the proof. ■

2.5 Finite Volume Form

The next step is to identify the algorithm (2) to (26) as a standard finite volume scheme as stated in (35). For the simplicity of notations, assume that $\Delta M_r^n > 0$. Due to the definition of the scheme (2) to (26), one has

$$\begin{aligned}u_j^{n+1} &= \mathbf{u}_j^{n+1} + \lambda_r^n(M_j^n - \mathbf{u}_j^{n+1}) = \hat{u}_j^n - \Delta m_j^n + \lambda_r^n(M_j^n - \mathbf{u}_j^{n+1})\\ &= \left(u_j^n - \nu(u_j^n - u_{j-1}^n)\right) - \nu(d^n_{j+\frac12} - d^n_{j-\frac12}) + \left(-\Delta m_j^n + \lambda_r^n(M_j^n - \mathbf{u}_j^{n+1})\right).\end{aligned}$$

In order to be able to rewrite the scheme as a finite volume scheme, we need to construct some fluxes $e^n_{j+\frac12}$ such that for all j

$$\Delta t(e^n_{j+\frac12} - e^n_{j-\frac12}) = -\Delta m_j^n + \lambda_r^n(M_j^n - \mathbf{u}_j^{n+1}). \tag{32}$$

The solution can be constructed as

$$\begin{cases} e^n_{rp-\frac12} = 0, & \forall r \in \mathbf{Z},\\ e^n_{j+\frac12} = \frac{1}{\Delta t}\sum_{rp\le k\le j}\left(-\Delta m_k^n + \lambda_r^n(M_k^n - \mathbf{u}_k^{n+1})\right), & rp < j.\end{cases} \tag{33}$$

This formula is correct because (24)-(26) implies that (33) is correct for $j = (r+1)p-1$: $e^n_{(r+1)p-\frac12} = 0$. If one (or more) of the $\Delta M_r^n < 0$ is negative, the result remains the same.

LEMMA 2.4 The L^1 norm of $(e^n_{j+\frac12})$ is bounded

$$||e^n||_1 = \Delta x \sum_j |e^n_{j+\frac12}| \le 2C_1(2k+1)p||u||_{BV}, \tag{34}$$

where $||u||_{BV}$ is the BV norm of the solution.

Proof Due to the definition (33) we deduce

$$||e^n||_1 \le \frac{p}{\Delta t}\Delta x \sum_j |\Delta m_j^n|.$$

But one always has

$$\begin{aligned}|\Delta m_j^n| \le |\hat{u}_j^{n+1} - (u_j - \nu(u_j^n - u_{j-1}^n))| &\le \nu(|d^n_{j+\frac12}| + |d^n_{j-\frac12}|)\\ &\le \nu(|c^n_{j+\frac12}| + |c^n_{j-\frac12}|),\end{aligned}$$

moreover due to the hypothesis (4) we finally get

$$\frac{\Delta x}{\Delta t}\sum_j |\Delta m_j^n| \le 2C_1(2k+1)\sum_j |u_j^n - u_{j-1}^n| \le 2C_1(2k+1)||u||_{BV}.$$

This ends the proof. ■

Then we are able to write the repair algorithm as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n - u_j^{n-1}}{\Delta t} = (s_j^n - s_{j-1}^n), \tag{35}$$

where s_j^n is defined by

$$s_j^n = -\frac{1}{\Delta x} a d^n_{j+\frac12} + e^n_{j+\frac12}. \tag{36}$$

The L^1 norm of s^n is bounded by

$$||s^n||_1 \le C_1(2k+1)||u||_{BV} + ||e^n||_1 \le C_1(2k+1)(2p+1)||u||_{BV}. \tag{37}$$

From (35) we get

$$\frac{u^{n+1} - u^n}{\Delta t} + \frac{a}{\Delta x}(I - T)u^n = (I - T)s^n, \tag{38}$$

where T is the translation operator to the right. This is equivalent to say that

$$u^n - (I + \nu(T - I))^n u^0 = \Delta t \sum_{p=0}^{n-1}(I + \nu(T - I))^{n-p-1}(I - T)s^p. \tag{39}$$

The main convergence result is given by the

THEOREM 2.5 Assume that $\nu < 1$. The difference between the upwind scheme and the repair algorithm (as defined by (2) to (26), or (35-36)) tends to zero with Δx in the sense that there exists a constant $C_2 > 0$ which does not depend on k, p, C_1, such that

$$||u^n - (I + \nu(T - I))^n u^0||_1 \le C_2 C_1(2k+1)(2p+1)||u||_{BV}\sqrt{(n\Delta t)\Delta x}. \tag{40}$$

Proof Equation (38-39) shows that the scheme exactly fits in the framework developed in [7] and applied in [8] for the convergence analysis of non linear schemes for linear advection. The key estimate proved in [7] is that

$$||(I + \nu(T - I))^{q+1}(I - T)||_1 \le C\frac{1}{\nu(1-\nu)}\frac{1}{\sqrt{q}}. \tag{41}$$

The constant C is a universal one. A simplified proof is provided in appendix for the case $\nu = \frac12$. This estimate means that the right hand side of (39) is made of terms that are small with respect to $n - p$. The final stage of the proof is straightforward. One has

$$||u^n - (I + \nu(T - I))^n u^0||_1 \le \Delta t \max||s^p||_1 \times \sum_{p=0}^{n-1}||(I + \nu(T - I))^{n-p-1}(I - T)||_1$$

$$\le C\frac{1}{\nu(1-\nu)}\Delta t \max||s^p||_1 \sum_{q=1}^{n}\frac{1}{\sqrt{q}} \le C\frac{1}{\nu(1-\nu)}\Delta t \max||s^p||_1 \int_0^n \frac{dy}{\sqrt{y}}$$

$$\le C\frac{1}{\nu(1-\nu)}\Delta t C_1(2k+1)(2p+1)||u||_{BV}\frac{\sqrt{n}}{2}$$

$$\le \frac{C}{2a\sqrt{1-\nu}}C_1(2k+1)(2p+1)||u||_{BV}\sqrt{T\Delta x}.$$

Defining $C_2 = \frac{C}{2a\sqrt{1-\nu}}$ ends the proof. ■

Remark: Since the numerical solution of the upwind scheme converges to the exact solution meaning

$$(I + \nu(T - I))^n u^0 \to u(n\Delta t) \quad \text{in } L^1(\mathbf{R}) \text{ as } \Delta x \to 0,$$

then the inequality of theorem 2.5 is a convergence result. Of course this inequality does not explain that the repair algorithm is better than the upwind scheme. Actually Repair is equivalent to the upwind scheme is one choose $c^n_{j+\frac12} = 0$, for all j and n. However the result shows that repair can not diverge if $p^2\Delta x \to 0$.

Remark: On the other hand the error estimate blows up if p is too large. At least we need $p^2\Delta x \to 0$ to get a vanishing error on the right hand side of the estimate. This estimate is the reason why we have incorporated the correction step in the repair algorithm.

Remark: One may wonder the reason of the $\frac{1}{a\sqrt{1-\nu}}$ contribution in the definition of the constant C_2. Indeed if $a \to 0$ or $\nu \to 1$, then C_2 can go to infinity which makes the estimate of convergence meaningless. First of all, for a given computation $C_2 < \infty$. Second of all, it is possible to use sharper estimates to get rid of the $\frac{1}{a}$. This is done in [7] for the convergence analysis of TVD schemes for instance. Finally one may argue that a similar argument should be possible at the theoretical level to get rid of the $\frac{1}{\sqrt{1-\nu}}$. Nevertheless real computations use $\nu < 1$ which is another reason to use this hypothesis. At the numerical level, we never saw any dependence of the rate of convergence with respect to this parameter. Thus we consider this as an artifact of the analysis.

3 Repair algorithm for remapping

In this section we show how to extend the previous result to take into account some very simple remapping algorithms. The main idea is that remapping is very close

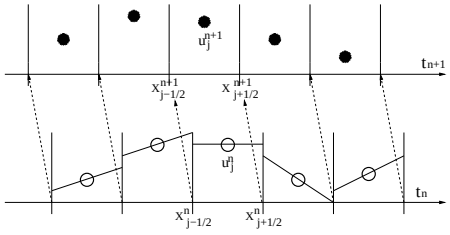


Figure 1: Remapping stage. The old and new densities are the bullets. The *high order* reconstruction stage are represented with the slopes.

to transport. So let us consider a uniform mesh. (The uniformity of the mesh is not absolutely necessary however this hypothesis simplifies a lot the proof.).

At time step n, the value of the unknown is u_j^n in the cell j. The boundaries of cell j are $x^n_{j-\frac12}$ and $x^n_{j+\frac12} = x^n_{j-\frac12} + \Delta x$, where Δx is the mesh size. Then the mesh changes. We assume the simplest change: a uniform translation to the left, then

$$x^{n+1}_{j+\frac12} = x^n_{j+\frac12} - a\Delta t, \quad \forall j, \tag{42}$$

where $a > 0$ is a kind of mesh velocity and Δt is an equivalent time step. We assume that the mesh does not move too fast, that is we assume the CFL condition

$$\nu = a\frac{\Delta t}{\Delta x} \le 1.$$

A standard remapping algorithm is divided in two stages : first reconstruct a *high order* profile using a MUSCL like algorithm, then project on the new mesh. Let us detail these operations. An illustration is given in figure 1.

The reconstruction amounts to the definition of the slopes in function of the old values. So we define

$$v_j^n(x) = u_j^n + w_j^n(x), \tag{43}$$

where w_j^n is the slope function which is added to the average value at step n. A natural hypothesis is that the average value of $w_j^n(x)$ is zero

$$\int_{x^n_{j-\frac12}}^{x^n_{j+\frac12}} w_j^n(x) = 0, \tag{44}$$

and that $w_j^n(x)$ is a local reconstruction, that is

$$|w_j^n(x)| \le C_3 \sum_{j-k\le q\le j+k} |u_q^n - u_{q-1}^n|. \tag{45}$$

Then we project the reconstructed function onto the new mesh. Thus

$$u_j^{n+1} = \int_{x_{j-\frac{1}{2}}^{n+1}}^{x_{j+\frac{1}{2}}^{n+1}} v^n(x)dx. \tag{46}$$

That is

$$u_j^{n+1} = \int_{x_{j-\frac{1}{2}}^{n+1}}^{x_{j-\frac{1}{2}}^{n}} v_{j-1}^n(x)dx + \int_{x_{j-\frac{1}{2}}^{n}}^{x_{j+\frac{1}{2}}^{n+1}} v_j^n(x)dx.$$

Replacing $v_j^n(x)$ and $v_{j-1}^n(x)$ using (43), one gets

$$u_j^{n+1} = \nu u_{j-1}^n + (1-\nu)u_j^n + \int_{x_{j-\frac{1}{2}}^{n+1}}^{x_{j-\frac{1}{2}}^{n}} w_{j-1}^n(x)dx + \int_{x_{j-\frac{1}{2}}^{n}}^{x_{j+\frac{1}{2}}^{n+1}} w_j^n(x)dx.$$

Let us define

$$c_{j-\frac{1}{2}}^n = -\frac{1}{x_{j-\frac{1}{2}}^n - x_{j-\frac{1}{2}}^{n+1}} \int_{x_{j-\frac{1}{2}}^{n+1}}^{x_{j-\frac{1}{2}}^{n}} w_{j-1}^n(x)dx = -\frac{1}{a\Delta t} \int_{x_{j-\frac{1}{2}}^{n+1}}^{x_{j-\frac{1}{2}}^{n+1}+a\Delta t} w_{j-1}^n(x)dx. \tag{47}$$

Hypothesis (45) turns into

$$|c_{j+\frac{1}{2}}^n| \leq C_3 \sum_{j-k\leq q\leq j+k} |u_q^n - u_{q-1}^n|. \tag{48}$$

Moreover using (44) — the fact that the mean value of the correction is zero — yields to the definition of the new value $\overline{u}_j^{n+1}$

$$\frac{\overline{u}_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_j^n + c_{j+\frac{1}{2}}^n - u_{j-1}^n - c_{j-\frac{1}{2}}^n}{\Delta x} = 0. \tag{49}$$

With these notations the remapping algorithm is equivalent to a non linear transport. Thus we easily generalize the results of the previous section to remapping.

The repair algorithm for remapping is then :

1) reconstruct a *high order* approximation as in (43-45) ;
2) make the correction step as in subsection 2.2 ;
3) repair.

The generalization of the correction step of subsection 2.2 is easy to write down, since the notations are quite similar between section 2 and section 3. It is sufficient to notice that once $p \in \mathbf{N}^\star$, the size of the boxes, has been chosen, then the correction step checks if the inequalities (7-8) become true. If one of them is not true, then we need to minimize the value of $c_{j+\frac{1}{2}}^n$ at the borders of the boxes until the inequalities are true.

Then the convergence result of theorem 2.5 is extended to Repair remapping algorithms. For a given size of the boxes p, the difference between the Repair Algorithm and the upwind or donor cell method is bounded by estimate (40) (with $C_1 = C_3$).

4 Numerical results

In the numerical result presented, we tried the Lax-Wendroff scheme plus Repair and the Downwind scheme plus Repair. The velocity is $a = 1$.

The Lax-Wendroff scheme is given by

$$\textbf{(LW)} \quad \frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{u_j^n + \frac{1}{2}(1-\nu)(u_{j+1}^n - u_j^n) - u_{j-1}^n + \frac{1}{2}(1-\nu)(u_j^n - u_{j-1}^n)}{\Delta x} = 0.$$

The Downwind scheme is

$$\textbf{(DW)} \quad \frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{u_{j+1}^n - u_j^n}{\Delta x} = 0.$$

The LW intends to be representative of high-order prediction schemes. The DW scheme intends to be representative of highly anti-dissipative prediction schemes. Concerning the Repair algorithm we used different box sizes.
We advected two initial profiles: a smooth one ($u_0(x) = -\cos 2\pi x$ on $[0,1]$) and a discontinuous one ($u_0(x) = 1$ for $0 < x < 0.5$ and $u_0(x) = 0$ for $0.5 < x < 1$) on a domain with periodic boundary conditions.

4.1 Figures

Essentially the results LW+Repair show good convergence properties independently to the size of the box p. See Figures 2 and 3. However one sees for the cosine initial condition that $p = 1$ leads to some discrepancy at the extrema of the solution, that $p = 50, 100$ leads to a prediction of the extrema that is less accurate. On the other hand a relative small box of size $p = 4$ gives the best result. On this case, a small amount of local repairing gives good results. For the step initial condition of Figure 3, the spreading is quite the same with all box sizes.

On the other hand the DW+repair results may be very sensitive to the size of the box. For a given number of cells, one gets the best results for the smaller size of the box, see Figure 4, staircases appear, exactly as with the UltraBee scheme as reported in [6], see also [2] and [12]. For a given box size the numerical solution converges to the exact one as the number of cells increases, see Figure 5. Unfortunately the global algorithm where the size of the box is equal to the number of cells (i.e. $p = n$) seems to diverge as the time increases, see Figure 6. Our experiments showed that this pathology is highly sensitive to the CFL number. For $CFL = .5$, the global algorithm seems to be correct. We retain that the global algorithm is not reliable for such test cases.

4.2 Order of convergence

In tables 1 to 3 are gathered some errors in various norms. The size of the boxes is $p = 4$. The final time is $t = 1$. We increase the number of cells from 100 to 1600. The order of convergence of the Lax-Wendroff plus Repair algorithm is $\approx \frac{1}{2}$ in L^1 for a discontinuous profile. This is in accordance with the theoretical result of Theorem 2.5. For a smooth profile the order is ≈ 2 in all norms.

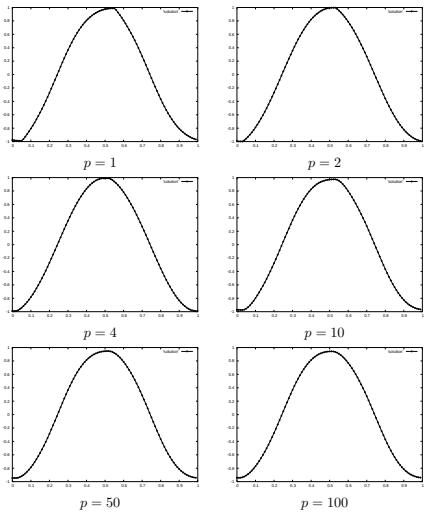


Figure 2: Lax-Wendroff plus Repair : 100 cells. The initial condition is $u_0(x) = -\cos 2\pi x$. The size of the box is $p = 1, 2, 4, 10, 50, 100$ from top left to bottom right. The final time is $t = 10$

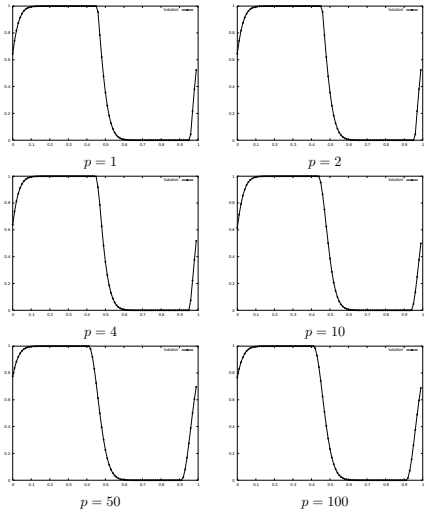


Figure 3: Lax-Wendroff plus Repair : 100 cells. The initial condition is $u_0(x) = 1$ for $0 < x < 0.5$ and $u_0(x) = 0$ for $0.5 < x < 1$. The size of the box is $p = 1, 2, 4, 10, 50, 100$ from top left to bottom right. The final time is $t = 1$

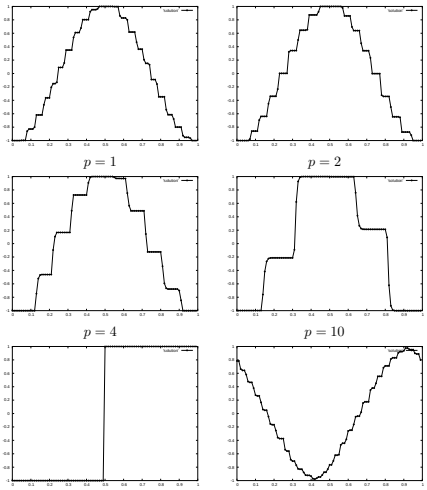


Figure 4: Downwind plus Repair. The initial condition is $u_0(x) = -\cos 2\pi x$. The size of the box is $p = 1, 2, 4, 10, 50, 100$ from top left to bottom right. The number of cells is $N = 100$. The Courant number is $CFL = 0.12345$. The final time is $t = 1$

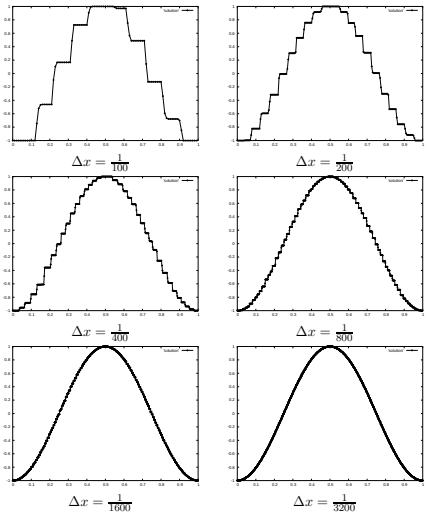


Figure 5: Downwind plus Repair. The initial condition is $u_0(x) = -\cos 2\pi x$. The size of the box is $p = 4$. The number of cells is $n = 100, 200, 400, 800, 1600, 3200$. The final time is $t = 1$

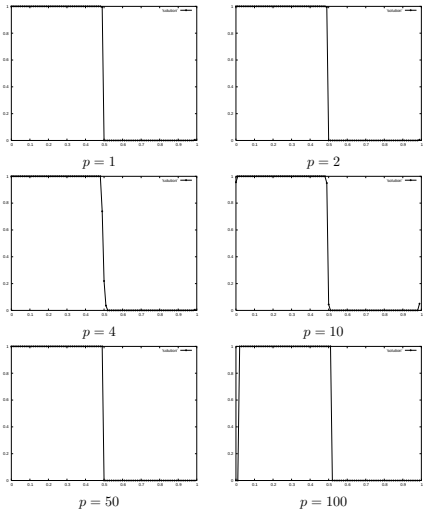


Figure 6: Downwind plus Repair : 100 cells. The initial condition is $u_0(x) = 1$ for $0 < x < 0.5$ and $u_0(x) = 0$ for $0.5 < x < 1$. The size of the box is $p = 1, 2, 4, 10, 50, 100$ from top left to bottom right. The final time is $t = 1$. The results are quite good, except for the global algorithm which seems to be in advance.

128

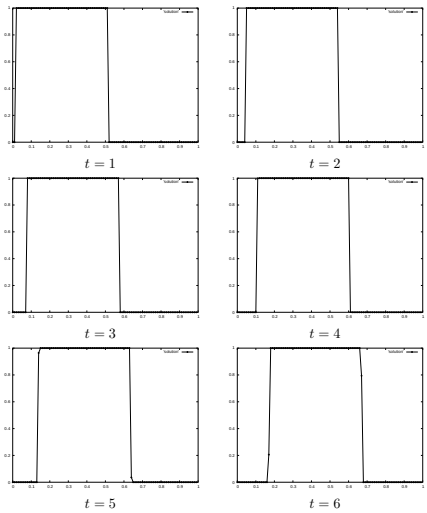


Figure 7: Downwind plus Repair : 100 cells. The initial condition is $u_0(x) = 1$ for $0 < x < 0.5$ and $u_0(x) = 0$ for $0.5 < x < 1$. The size of the box is $p = 100$. The final time is $t = 1, 2, 3, 4, 5, 6$. The global algorithm is in advance.

2.3 ReALE : Reconnection ALE

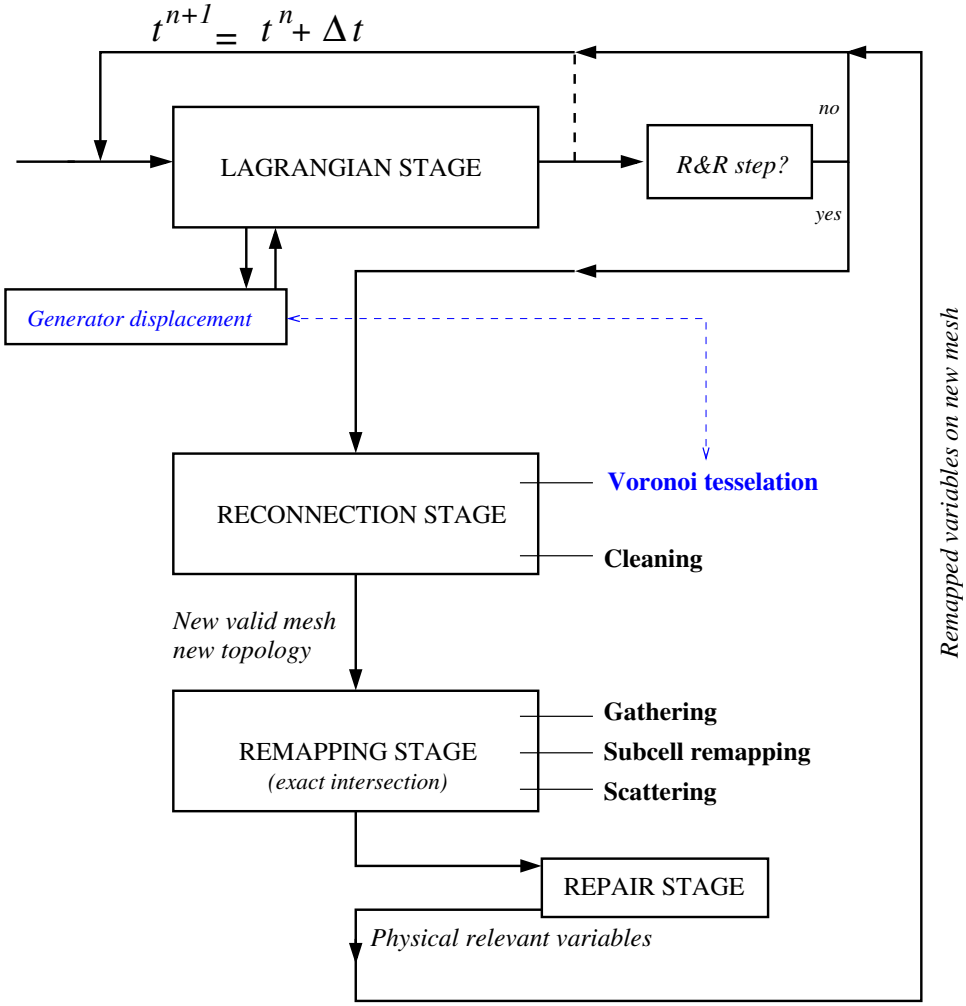


FIGURE 2.6 – *Description of a ReALE (Reconnection ALE) scheme within ALE INC(ubator). A Lagrangian scheme is followed by a reconnection stage which determines the Voronoi tesselation from moving generators (computed within the Lagrangian stage) onto which a remapping phase conservatively projects the physical variables and ultimately provides physically relevant variables (thanks to a so-called repair technique). The remapping must be done by exact intersection between the old Lagrangian mesh and the new mesh from the reconnection stage, the later may have a different connectivity.*

In this section we present the joint effort with colleagues from LANL (M. Shashkov) and CEA (J. Breil, S. Galera, P.-H. Maire) to extend the ALE technique to allow mesh reconnection during the simulation. In Fig.2.6 one proposes a sketch of our Reconnection ALE (ReALE) algorithm to provide the big picture to be compared to a fixed connectivity ALE in Fig. 2.3. Some of the work made within this context is described in details in this chapter. This work has led us to write paper [26] entitled *"ReALE : a reconnection-based arbitrary-Lagrangian-Eulerian method"* which sets the fundation of two reconnection-based ALE codes : a cell-centered ReALE code named CHIC developed and maintained at CELIA in Bordeaux , and ALE INC(ubator) in its ReALE version maintained at IMT in Toulouse. In paper [27] an extension to cylindrical geometry is also proposed.

In this section we recall the genesis of ReALE and the reasons why this work has been undertaken. Then a quick refresher on Voronoi machinery is proposed followed by a some details and specific treatment that a ReALE code needs. Finally ReALE results are provided.

"Why fooling around with a Reconnection ALE method ?"

The genesis of this work starts with the following statement : The most difficult and least developed phase of ALE is the rezoning phase. A review of existing rezone strategies for ALE methods is presented in [124] including analysis of alternative approaches [145, 146, 147, 108, 148, 112, 149, 116, 150, 151, 115]. A review of a more general class of methods, namely moving mesh methods, is presented in [152]. Ideally the mesh has to adapt to the solution. Any adaptive scheme is composed of three main ingredients : an optimal-mesh criterion, an error estimator or error indicator, and an algorithm of the strategy for the mesh improvement. These ingredients answer to the following questions : How should the optimal mesh be defined ? Where are mesh changes required ? And how should the improved mesh be constructed ? For standard ALE methods a strategy for mesh improvement is based on moving the spatial grid.

Generally speaking the goal of rezoning is to improve the efficiency of the ALE method However, to design an adaptive method one needs a quantitative assessment of optimality. The problem is that, for non-linear equations of gas dynamics in 2D and 3D, at the moment, it is not feasible to obtain such quantitative assessment. For this reason practitioners are usually using some qualitative approaches. In real complex ALE simulation the most basic goal of rezoning is simply to run calculation to completion without user intervention and still achieve reasonable accuracy (recall that we always can run ALE in Eulerian=Lagrange-Plus-Remap mode, which will be robust but less accurate). Even this goal is usually not achieved in most production ALE codes. For example, even for very popular methods based on Winslow smoothing, [115, 116], practical simulations require the introduction of numerous geometrical and physics based triggers and lockers, that is, mesh constraints that typically keep a node Lagrangian until some condition is reached e.g. element quality criterion (to detect cell distortion or collapse) or physical condition is reached in surrounding elements (for example, did the cell fully detonate ?) [112, 153, 154, 138].

As it is mentioned in [153, 154], the mesh movement philosophy applied to most applications, related to high-speed multimaterial flows, is to develop algorithms that will move the mesh in such a way as to maintain robustness while staying as close as possible to the Lagrangian mesh motion. The Lagrangian mesh motion naturally follows most flow features of interest such as shocks, material interfaces and steep gradients and allows users to focus zoning in materials of interest. Mesh relaxation is then used in regions of high material deformation to improve mesh quality. In standard ALE methods, which use fixed mesh topology, nodes are moved to refine some areas of the problem at the expense of derefining other parts of the problem. Generally, the increase of mesh resolution is limited, and, most importantly it can degrade the mesh quality leading to robustness problems.

One if not the main reason for this is that standard ALE codes utilize a fixed topology mesh, defined at the outset, which in general will not be able to adapt to the dynamically evolving interface shape

(or contact discontinuity) in spite of efforts at regularization, see the next figure for an illustration.

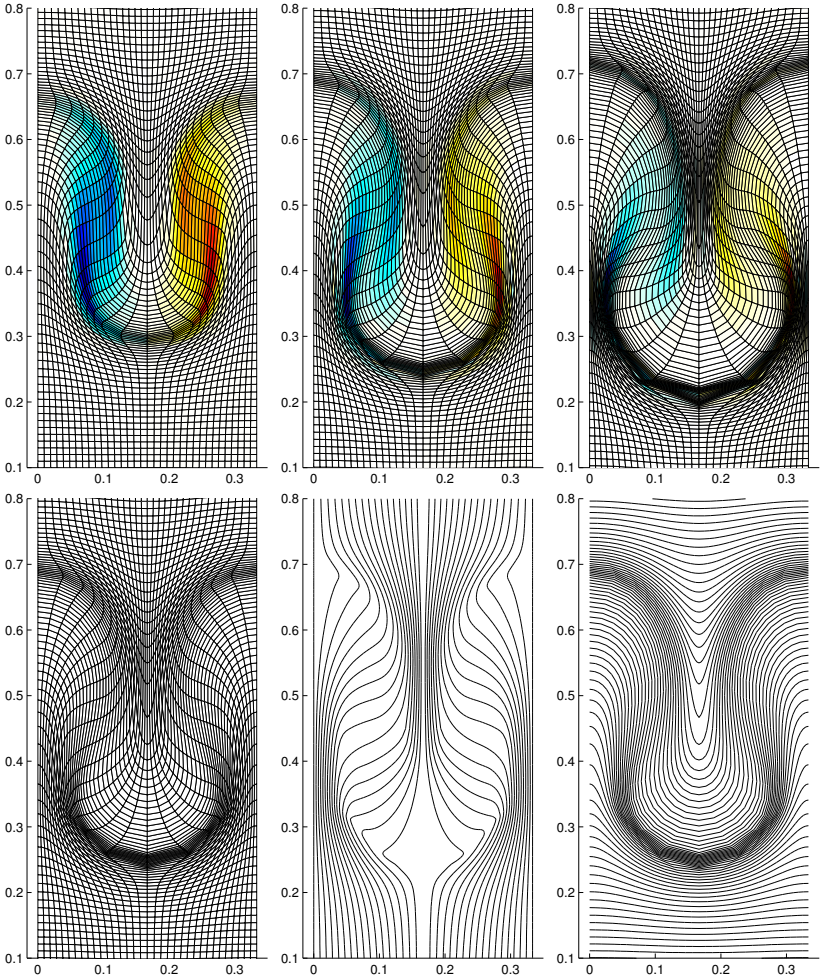


Illustration of mesh stagnation in a fixed ALE simulation. Rayleigh-Taylor problem on logicaly rectangular mesh.
Top panels : mesh fragments and vorticity color map at time moments $t = 7, 8, 9$. Color scale is from blue (minimal negative vorticity) to red (maximal positive vorticity). White color corresponds to zero vorticity.
Bottom panels : fragment of mesh at $t = 8$: mesh fragment (left), "Vertical" logical lines (middle), "Horizontal" logical lines (right).
Because one can not expect the mesh to follow anymore the vortex like motion without tangling, the rezone strategy locks the mesh : the code can not help but continue in an almost Lagrange+Remap strategy.

The most general solution to this difficulty, while preserving a Lagrangian nature of mesh motion is to relax the constraints on mesh topology and allow reconnection. The idea of using mesh reconnection to solve partial differential equations is not new. To the best of our knowledge, in context of computational gas dynamics the ideas related to mesh reconnection were first used in [155] In this seminal paper the authors suggest to use a set of point Lagrangian particles and surround them with domains (parcels) to describe the media. The shape and size of these parcels are determined by the positions of the particles. The connectivity of the set of particles is not fixed but can vary with time depending on relative positions of the particles. After connectivity is established the set of neighboring particles defines the stencil on which Lagrangian equations are discretized.
Paper [155] has all basic ideas that lead to development of so-called free-Lagrange (or free-Lagrangian) methods, [156, 157, 158], which were very popular in 80's and early 90's The name free-Lagrange was introduced in [159] and the corresponding code was called FLAG at LANL.
More recently in the context of cell centered Lagrangian scheme S. Del Pino in [160] has developed a metric-based adaptation technique which also allows automatic triangular mesh reconnection. Allowing mesh reconnection is not a new idea *per se* and several related subjects can be found in the recent literature such as [161], [160] and [162, 163].
In paper [26] a detailed discussion on so called free-Lagrangian methods enlights that these are not genuinely "Lagrangian" methods. Because free-Lagrangian methods are not really Lagrangian then,

Figure 2.7 – *Example of Voronoi meshes from mother nature. Left : dragonfly wings. Middle : girafes. Right : turtle.*

explicitly or implicitly, they incorporate a rezone phase, and consequently, a corresponding remap phase. Some of them explicitely states this remap phase. Nevertheless if the free-Lagrange method does not have a remap phase, errors related to it will manifest itself one way or another. It leads us to the conclusion that methods where connectivity of the mesh can change have to be developed in reconnection-based ALE (ReALE) framework, where rezone stage includes reconnection. Let us note that similar philosophy was used in [147], even so authors of [147] do not call their method ALE or free-Lagrange.

As standard for ALE method, the main elements in ReALE simulation are an explicit Lagrangian phase in which the solution and grid are updated (without changing connectivity), a rezoning phase in which a new grid is defined (which includes changing connectivity and also adding or deleting cells or vertices of the parcels), and a remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) onto the new grid. Our rezoning phase allowing mesh reconnection is based on the Voronoi machinery, however for the sake of clarity we remind some aspects of it in the next paragraph.

"Voronoi machinery"

Voronoi diagrams were first investigated by René Descartes (French philosopher 1596-1650) and applied by Lejeune Dirichlet (Belgium/German mathematician 1805-1859) when exploring quadratic forms, however the diagrams were named after Georgy E. Voronoi (Ukrainian mathematician 1868-1908). To be fair the real inventor of these diagrams is Mother Nature as illustrated in Fig.2.7.

Given generators $\boldsymbol{G}_i$, $i = 1, 2, \ldots, G$, a distance function $d(\boldsymbol{G}_i, \boldsymbol{G}_j)$ the Voronoi cell Ω_j is the set of all points closer to $\boldsymbol{G}_j$ than to any of the other $\boldsymbol{G}_i$

$$\Omega_j = \left\{ \boldsymbol{X} \;\; s.t \;\; d(\boldsymbol{G}, \boldsymbol{G}_j) < d(\boldsymbol{X}, \boldsymbol{G}_i), \; \forall \, i = 1, \ldots, G, \;\; i \neq j \right\}. \tag{2.3}$$

A collection of Voronoi cells $\{\Omega_1, \Omega_2, \ldots, \Omega_G\}$ defines the Voronoi tesselation of $\mathbb{R}^2$ associated to the set of generators. For now we implicitely clip the Voronoi tesselation to the computational polygonal domain.

The "Voronoi machinery" denotes the ability of Voronoi tesselation to assimilate new generators and to perform the necessary modification of connectivity that the presence of these new generators implies. For example in Fig. 2.8 the Voronoi machinery is illustrated when generators (red dots) are successively added (two sequences are shown : 5, 6, 7 and 11, 12, 13 generators). On purpose the

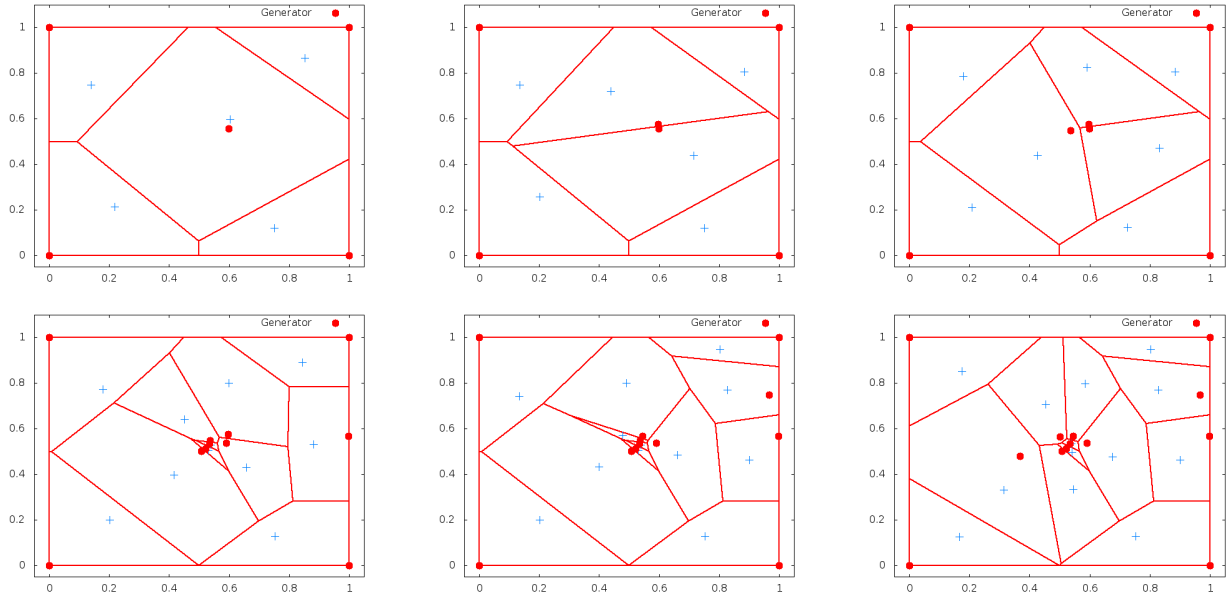


FIGURE 2.8 – *Example of Voronoi meshes obtained from a set of generators (red dots). Cell centroids (blue crosses) generally do not coincide with generators. This illustrate the reconnection ability of the Voronoi machinery when generators are added. Top panels : sequence with* 5, 6, 7 *generators. Bottom panels : sequence with* 11, 12, 13 *generators.*

generators are located close to the domain center, this illustrates the possible roughness of a Voronoi mesh : close cells can have very different surfaces, number and size of edges, shape, etc. Moreover if $\boldsymbol{X}_c$ denotes the centroid of the Lagrangian cell Ω_c, according to

$$\boldsymbol{X}_c = \frac{1}{\mid \Omega_c \mid} \int_{\Omega_c} \boldsymbol{X} \mathrm{d}V,$$

where $\mid \Omega_c \mid$ denotes the volume of the cell Ω_c then we observe in Fig. 2.8 that the centroids (blue crosses) almost never coincide with the generators (red bullets). In fact when these two coincide then the Voronoi tesselation is called Centroidal Voronoi Tesselation (CVT), see Fig.2.9 last panel. In $\mathbb{R}^2$ a CVT is a mesh made of perfect hexagonal cells. One simple constructive iterative algorithm to obtain a CVT (and smooth a Voronoi tesselation when convergence is not reached) is due to Lloyd [164] :

0. Iteration k. Start with generators $\left\{\boldsymbol{G}_i^k\right\}$, $i = 1, \ldots, G$.
1. Build Voronoi cells Ω_i^k associated to $\boldsymbol{G}_i^k$ for all $i = 1, \ldots, G$.
2. Compute $\boldsymbol{X}_i^k$ centroid of Voronoi cell Ω_i^k for all $i = 1, \ldots, G$.
3. Set $\boldsymbol{G}_i^{k+1} = \boldsymbol{X}_i$ for all $i = 1, \ldots, G$.
4. If satisfied with obtained mesh quit, else $k \leftarrow k + 1$ and go back to 1.

In Fig.2.9 one presents an example of the result of the iterative Llyod's algorithm [164] for iterations 1, 2, 3, 10, 20 and 100. This shows that initialy non uniformly distributed generators produce a non-smooth Voronoi tesselation whereas the successive meshes obtained with Llyod's algorithm are more and more regular. At convergence (last panel) a centroidal Voronoi tesselation for which centroid and generator do coincide is produced. A last drawback of Voronoi tesselation is its unpleasant ability to create arbitrary small edges. For a hydrodynamical Lagrangian scheme small edge may

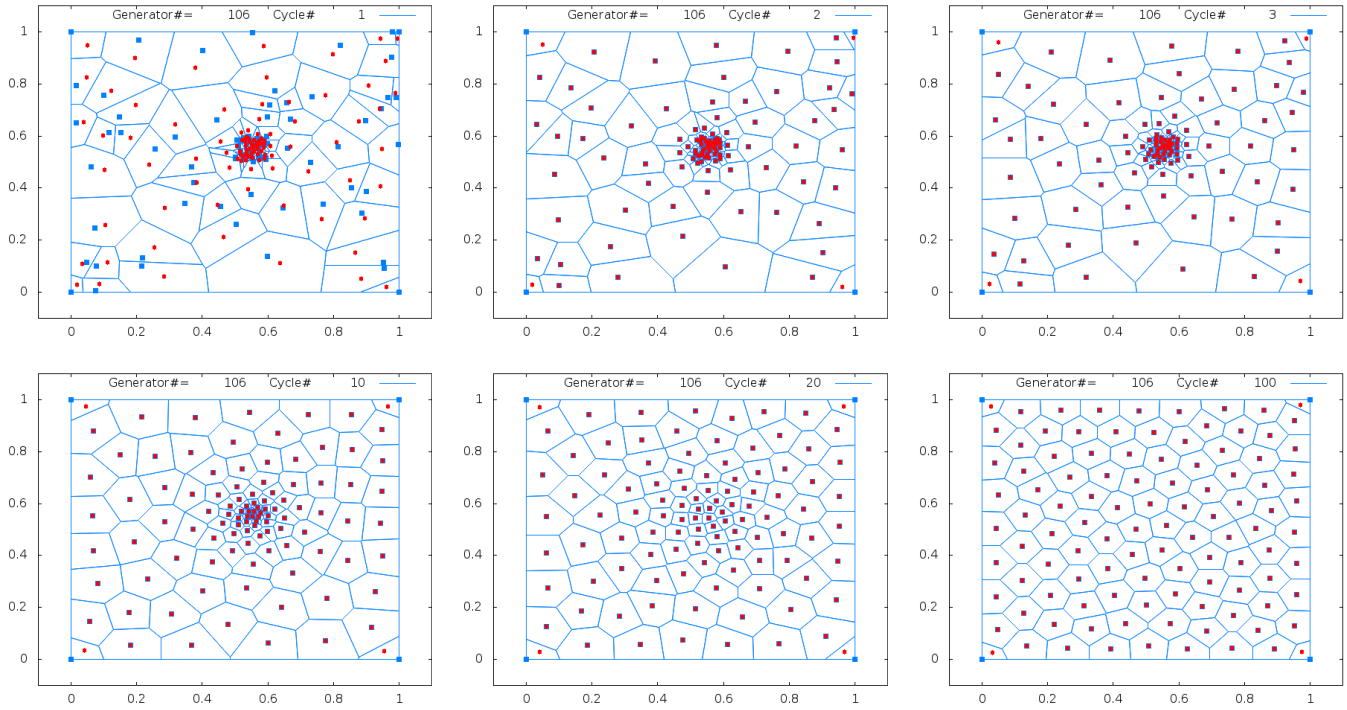


Figure 2.9 – *First panel : example of Voronoi mesh made of* 106 *generators (blue squares). Notice that cell centroid (red circles) usually does not coincide with cell generator. Second to sixth panels : iterations* 2, 3, 10, 20 *and* 100 *of Llyod's algorithm which ultimately produces a centroidal Voronoi tesselation for which centroid and generator do coincide.*

drive the time step to zero, consequently we have added a specific "small edge cleaning" procedure. Given a user-specified threshold edges which length is smaller are discarded and the connectivity is modified accordingly. As instance on the last panel of Fig.2.9 some small edges have been kept, as instance around point $\boldsymbol{X} = (0.2, 0.72)$ or $(0.38, 0.28)$, and some edges have been discarded as instance around $\boldsymbol{X} = (0.78, 0.56)$. As we will see in this the following sections incomplete CVT and Voronoi machinery are heavily used in ReALE.

ReALE specifics : initialisation and Lagrangian phases

For ReALE all three phases (Lagrangian, Rezone and Remap) are supposed to satisfy specific requirements which are different from standard ALE methods. We assume, that at the beginning of the calculation ($t = 0$) as well at the beginning of each time step (after rezone phase) the computational mesh consists of Voronoi cells corresponding to some set of generators. In other words the distribution of generators entirely defines the mesh, see Fig. 2.8 for examples of Voronoi mesh. Because of reconnection in rezone phase, the Lagrangian phase of the ReALE method has to deal with discretization of the Lagrangian equations on general polygonal meshes and corresponding update of this polygonal mesh is supposed to be Lagrangian. There are several papers dealing with discretization of Lagrangian equation on general polygonal meshes [74, 78, 114, 147, 165, 166]. We used the compatible mimetic finite discretizations [53, 54] on staggered mesh, which is historically close to [74, 78, 114] for ALE INC(ubator) (see previous chapter) and the CELIA team used the newly developed cell-centered discretizations based on Godunov approach [45, 49, 50].

ReALE specifics : rezone phase

The rezone phase of ReALE has to include both mesh movement and reconnection procedure. In paper [26] we used a set of generators and the machinery of Voronoi diagrams to do both mesh movement and mesh reconnection, see Fig.2.8 for an illustration of mesh reconnection *via* the Voronoi machinery. More precisely our rezone strategy consists of a special movement of generators. It is close to Lagrangian in some sense, but also include some smoothing procedure based on notion of centroidal Voronoi tesselation (CVT see the previous paragraph) [167]. The final position of a generator at time $t^{n+1} = t^n + \Delta t$ where Δt is the current time step is

$$\boldsymbol{G}_c^{n+1} = \boldsymbol{G}_c^{n+1,\text{lag}} + \omega_c \left(\boldsymbol{X}_c^{n+1} - \boldsymbol{G}_c^{n+1,\text{lag}} \right), \tag{2.4}$$

where the position vector of the generator of the Lagrangian cell Ω_c^n is denoted $\boldsymbol{G}_c^n$, and $\boldsymbol{G}_c^{n+1,\text{lag}}$ is a Lagrangian-like displacement of the generator obtained by :

$$\boldsymbol{G}_c^{n+1,\text{lag}} = \boldsymbol{G}_c^n + \Delta t \boldsymbol{U}_c, \tag{2.5}$$

where $\boldsymbol{U}_c$ is the "Lagrangian" velocity of the generator within the cell. This velocity is computed so that the generator remains located in the new Lagrangian cell. To this end we define this velocity to be the average of the velocities of the points of the cell, namely

$$\boldsymbol{U}_c = \frac{1}{\mid \mathcal{P}(c) \mid} \sum_{p \in \mathcal{P}(c)} \boldsymbol{U}_p^{n+\frac{1}{2}}.$$

Remind that $\mathcal{P}(c)$ denotes the set of points of the Lagrangian cell Ω_c and $\boldsymbol{U}_p^{n+\frac{1}{2}}$ is the time-centered velocity of point p between times t^n and t^{n+1} used to displace mesh point $\boldsymbol{X}_p^n$ to its new position

$\boldsymbol{X}_p^{n+1}$, that is to say following equation

$$\boldsymbol{X}_p^{n+1} = \boldsymbol{X}_p^n + \Delta t\, \boldsymbol{U}_p^{n+1/2}.$$

Finally $\omega_c \in [0;1]$ in (2.4) is a parameter that remains to be determined. The updated position of the generator is therefore defined by mean of a convex combination between the new Lagrangian-like position, $\boldsymbol{G}_c^{n+1,\text{lag}}$ and the centroid $\boldsymbol{X}_c^{n+1}$ of the Lagrangian cell. With this convex combination, the updated generator lies in between its Lagrangian position at time t^{n+1} and the centroid of the Lagrangian cell Ω_c^{n+1}. We note that for $\omega_c = 0$ we get a Lagrangian-like motion of the generator whereas for $\omega_c = 1$ we obtain a centroidal-like motion, which tends to produce a smoothed mesh. This latter case is equivalent to perform one Lloyd iteration [167, 168], see section 2.3 and Fig.2.9. We compute ω_c requiring that the generator displacement satisfies the principle of material frame indifference, that is for pure uniform translation or rotation we want ω_c to be zero. To this end, we construct ω_c using invariants of the right Cauchy-Green strain tensor associated to the Lagrangian cell Ω_c between times t^n and t^{n+1}. First, we define the deformation gradient tensor F

$$\mathsf{F} = \frac{\partial \boldsymbol{X}^{n+1}}{\partial \boldsymbol{X}^n},$$

where $\boldsymbol{X}^{n+1} = (X^{n+1}, Y^{n+1})^t$ denotes the vector position of a point at time t^{n+1} that was located at position $\boldsymbol{X}^n = (X^n, Y^n)^t$ at time t^n. The deformation gradient tensor is nothing but the Jacobian matrix of the map that connects the Lagrangian configurations of the flow at time t^n and t^{n+1}, in the two-dimensional case its components write

$$\mathsf{F} = \begin{pmatrix} \frac{\partial X^{n+1}}{\partial X^n} & \frac{\partial X^{n+1}}{\partial Y^n} \\ \frac{\partial Y^{n+1}}{\partial X^n} & \frac{\partial Y^{n+1}}{\partial Y^n} \end{pmatrix}.$$

The right Cauchy-Green strain tensor, C, is obtained by right-multiplying F by its transpose, i.e.

$$\mathsf{C} = \mathsf{F}^t\mathsf{F}.$$

In our case, C is a 2×2 symmetric positive definite tensor. This tensor reduces to the unitary tensor in case of uniform translation or rotation. It admits two positive eigenvalues, which are denoted λ_1 and λ_2 with the convention $\lambda_1 \leq \lambda_2$. These eigenvalues can be viewed as the rates of expansion in directions given by the eigenvectors during the transformation. To determine ω_c, we first construct the cell-averaged value of the deformation gradient tensor, F_c, and then the cell-averaged value of the Cauchy-Green tensor by setting $\mathsf{C}_c = \mathsf{F}_c^t\mathsf{F}_c$. Noticing that the two rows of the F matrix correspond to the gradient vectors of the X and Y coordinates, we can set $\mathsf{F}^t = \left[\boldsymbol{\nabla}_n X^{n+1}, \boldsymbol{\nabla}_n Y^{n+1}\right]$, where for any functions $\psi = \psi(\boldsymbol{X}^n)$, $\boldsymbol{\nabla}_n\psi = \left(\frac{\partial\psi}{\partial X^n}, \frac{\partial\psi}{\partial Y^n}\right)^t$. With these notations, let us define the cell-averaged value of the gradient of the ψ function over the Lagrangian cell Ω_c^n

$$(\boldsymbol{\nabla}_n\psi)_c = \frac{1}{\mid \Omega_c^n \mid} \int_{\Omega_c^n} \boldsymbol{\nabla}_n\psi \mathrm{d}V = \frac{1}{\mid \Omega_c^n \mid} \int_{\partial\Omega_c^n} \psi \boldsymbol{N} \mathrm{d}S.$$

Here, we have used the Green formula and $\boldsymbol{N}$ is the unit outward normal to the boundary of the cell Ω_c^n refered as to $\partial\Omega_c^n$. Assuming that this cell is a polygon and using the trapezoidal rule we obtain the following approximation for the previous integral

$$(\boldsymbol{\nabla}_n\psi)_c = \frac{1}{\mid \Omega_c^n \mid} \sum_{p=1}^{|\mathcal{P}(c)|} \frac{1}{2}\left(\psi_p^n + \psi_{p+1}^n\right) L_{p,p+1}^n \boldsymbol{N}_{p,p+1}^n, \tag{2.6}$$

where ψ_p^n is the value of ψ evaluated at point $\boldsymbol{X}_p^n$ and $L_{p,p+1}^n \boldsymbol{N}_{p,p+1}^n$ is the unit outward normal to the edge $[\boldsymbol{X}_p^n, \boldsymbol{X}_p^{n+1}]$. Applying (2.6) to $\psi = X^{n+1}$ and $\psi = Y^{n+1}$ we get a cell-averaged expression of the gradient tensor F and then deduce from it the cell-averaged value of the right Cauchy-Green tensor C_c. Knowing this symmetric positive definite tensor in each cell, we compute its real positive eigenvalues $\lambda_{1,c}$, $\lambda_{2,c}$. We finally define the parameter ω_c as follows

$$\omega_c = f(\lambda_1, \lambda_2) = \frac{1 - \alpha_c}{1 - \alpha_{\min}}, \tag{2.7}$$

where $\alpha_c = \frac{\lambda_{1,c}}{\lambda_{2,c}}$ and $\alpha_{\min} = \min_c \alpha_c$. We emphasize the fact that for uniform translation or rotation $\lambda_{1,c} = \lambda_{2,c} = 1$ and $\omega_c = 0$, therefore the motion of the generator is quasi Lagrangian and we fulfill the material frame indifference requirement. For other cases, ω_c smoothly varies between 0 and 1. The behavior of ω_c parameter has been tested with ReALE on Sedov problem [169] in Fig. 2.10 for three different generator displacement strategies : $\omega_c = 0$ to get a Lagrangian-like motion, $\omega_c = 1$ to obtain a centroidal-like motion, and ω_c defined by (2.7) using the strategy previously described. While the quasi-Lagrangian motion produces irregular mesh and results the centroidal-like motion generates an ultra regular and smoothed solution. However the mesh is so smoothed that the flow features (compressed cells after the shock wave as instance) have been litteraly washed out. Also notice that the part of the mesh at radius greater than 1 has not been attained by any wave at time $t = 1$. Consequently one expects the rezone/reconnection strategy to spare this region from smoothing. This is clearly not the case for the centroidal motion[7]. Conversely the deformation-tensor based generator motion furnishes a regular mesh and a more accurate solution than the quasi-centroidal motion. Above and over the generators have followed the fluid motion in an almost Lagrangian fashion without washing out the flow feature. Finally for the region beyond radius one, the generator velocity being zero, the original mesh is maintained untouched. This is due to the fact that $\lambda_{1,c} = \lambda_{2,c} = 1$ then $\omega_c = 0$ leading to a Lagrangian motion of generators the velocity of which is **0** (no wave has attained this region yet).

ReALE specifics : remap phase

In the remapping phase, the Lagrangian solution is transferred (conservatively interpolated) onto the rezoned mesh. Lagrangian mesh is the result of one time step Lagrangian movement of the Voronoi mesh corresponding to the distribution of the generators at the previous time step. The new rezoned mesh is the Voronoi mesh corresponding to the positions of generators created by the rezone phase. During the rezone phase generators are moved in an "almost" Lagrangian way and because Voronoi cells are changing their shape continuously with respect to positions of the generators ; rezoned and Lagrangian meshes are "close". However, in general, the connectivity of the Lagrangian and rezoned mesh are different. Consequently remapping methods have to be able to conservatively transfer flow parameters from one polygonal mesh to another. We use an exact intersection (overlay) based remap [129, 147, 127, 128], see the **Remap phase** paragraph of the section HISTORY AND PRESENTATION in this chapter. However, one could take advantage of how Lagrangian and rezoned meshes are constructed and design more efficient methods [170, 171, 172, 173] which has been done for CHIC code as instance.

7. Modifying untouched regions is one unacceptable feature of many rezoning strategies. Intrinsically, if the initial mesh is not optimal with respect to the underlying metric of the rezoner, sooner or later the algorithm will try to "optimize" the mesh in places where the user does not expect it to happen. Some solutions using triggers and filters can delay this feature but once nodes are marked for rezoning then the flaw will emerge.

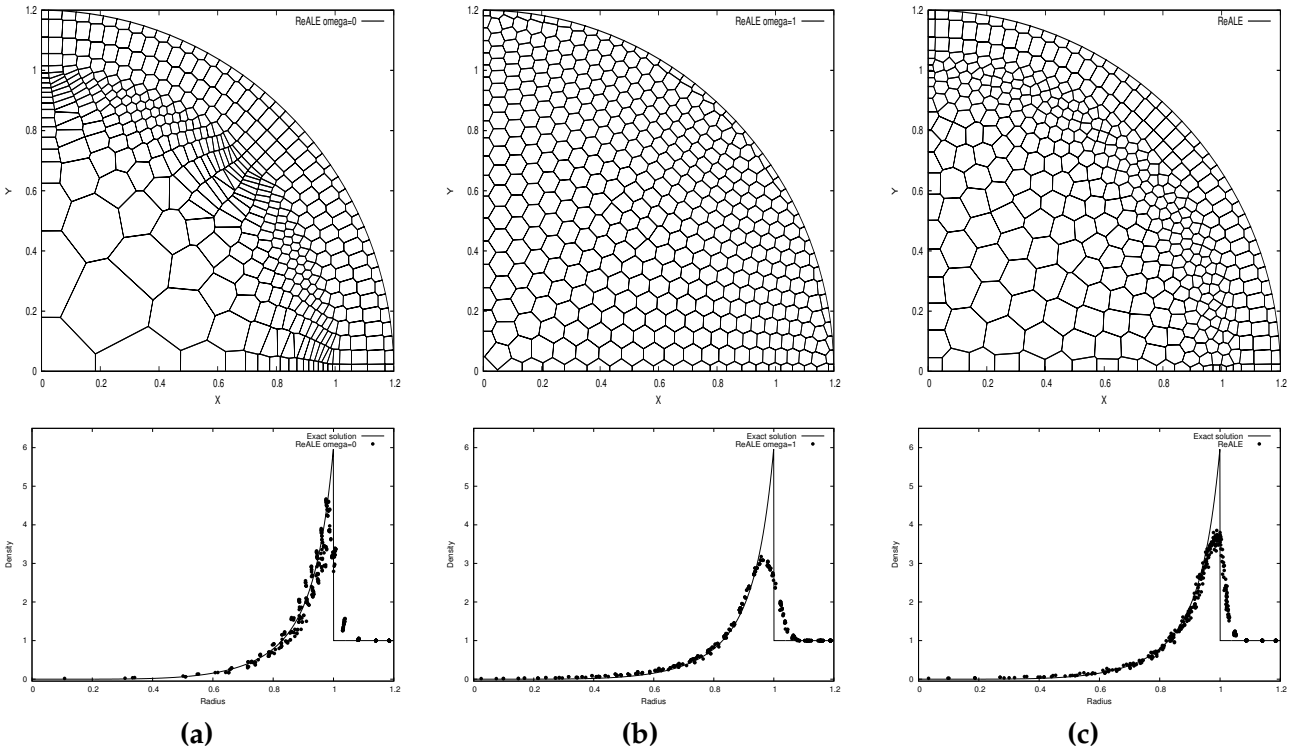


Figure 2.10 – *Results from [26]. Sedov problem at time $t = 1.0$ for different generator displacement strategies— Staggered ALE INC(ubator) code — Top panels : mesh. Bottom panels : density as a function of radius for all cells vs the exact solution (line) — **(a)** Quasi-Lagrangian generator motion $\omega_c = 0$ — **(b)** Quasi-centroidal generator motion $\omega_c = 1$ — **(c)** Deformation-tensor based generator motion $\omega_c = f(\lambda_1, \lambda_2)$.*

ReALE results

In paper [26] we have performed a set of test cases to show the behaviors of the ReALE technique with ALE INC(ubator) and CHIC codes. The Sedov test case has been used as a sanity check as almost no reconnection occurs and it has been shown that the reconnection treatment does not pollute the computation.
Next we have considered a two-material Riemann problem in 2D, the so-called triple point problem depicted in Fig. 2.11.

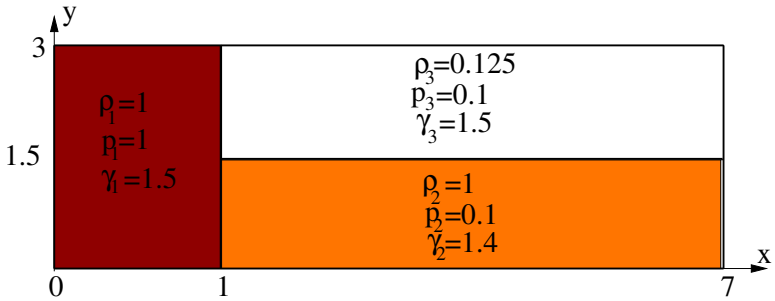


FIGURE 2.11 – *Figure from paper [26]. Triple point problem initialization.*

Due to the difference of density and gamma, two shocks in top and bottom domains propagate with different speeds. This creates a shear along initial horizontal contact discontinuity and a vorticity formation. Capturing the vorticity is the difficult part of such simulation when standard Lagrangian or ALE methods are used. In the following figure one reproduces the triple point problem results for Lagrangian, ALE and ReALE. These figures are devoted to visually measure "how much Lagrangian a method is". In such figures one displays the mesh where the cells have been colored according to in which domain they were initially located (white, red or orange). This way of presenting the results allows to observe if each cell move in an almost Lagrangian fashion. (In fact in this test case the orange cells must roll over with the white ones due to the vortex motion.)
In Fig.2.12 the first panel (time $t = 1.67$) corresponds, more or less, to the time after which any Lagrangian scheme inexorably fails due to mesh tangling. In the second and third panels of Fig.2.12 are displayed the ALE/ReALE results at final time $t = 5$. The second panel clearly shows the stagnation of the ALE mesh. During the Lagrangian stage of ALE the mesh is trying to follow the flow, but the development vorticity eventually leads to a tangled mesh. Next the rezoning step slightly relaxes the mesh and as such acts against the mesh motion. On the next time step of the Lagrangian phase, the mesh is trying to follow vorticity development and, again, is approaching a tangling situation. The fixed connectivity ALE mesh is the result of these competing processes : ultimately the mesh stagnates. ALE regime freezes the mesh to a position and the computation continues in an almost Eulerian fashion (as Lagrange+Remap because the rezone phase systematically backs up the Lagrangian t^{n+1} mesh onto the previous t^n Lagrangian mesh).
ReALE regime, third panel of Fig.2.12 nicely follows the vortex as cells are carried within the vortex in an almost Lagrangian fashion. As a consequence ReALE has a better accuracy as Voronoi cells are able to roll up and reconnect to new neighbors when necessary. This attests the ability of ReALE to properly follow such type of fluid motion.

In Fig. 2.13 one reproduces on the left the mesh and specific internal energy (left panels) and, the mesh where the cells have been colored according to the domain they were initially located (middle panels). On right panels we display the value of ω_c which shows that the regions where $\omega_c > 0.7$ are

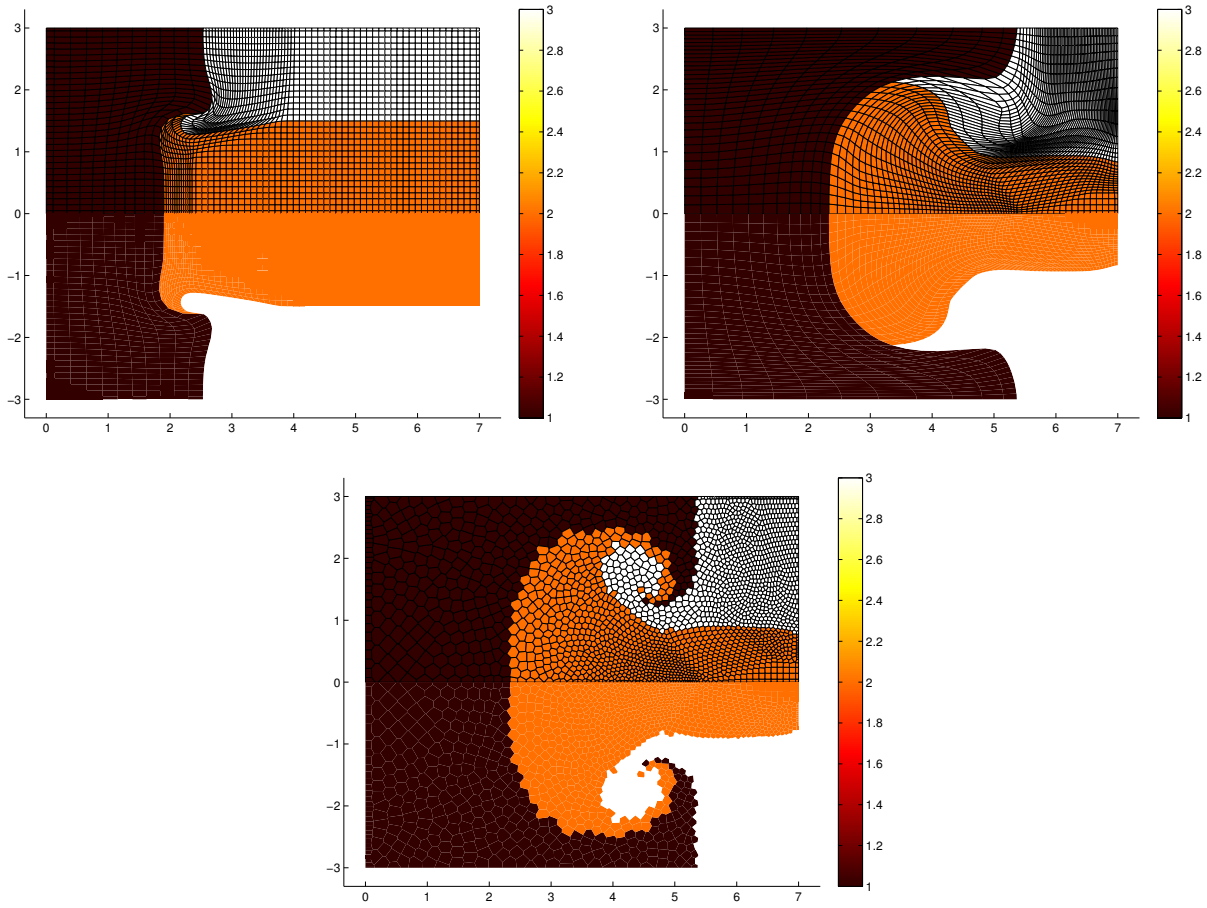


Figure 2.12 – *Results from paper [26]. Triple point problem results for Lagrangian, ALE and ReALE. The colors corresponds to the initial position of the cell within the three materials in Fig. 2.11 (Results from CHIC code.)*

clearly the regions where directional deformation occurs and where the mesh must be smoothed by the CVT technique.

The last test case proposed in [26] is the Rayleigh-Taylor instability. While fixed ALE results present the mesh stagnation as for the triple point problem (ultimately fixed ALE runs in a Lagrangian+Remap regime), the ReALE approach is able to follow the complex motion of the fluids see Fig. 2.14. In this figure one only reproduces the results at later time $t = 15$ when the top heavy fluid has reached the bottom and moves upward. Due to the vorticity, mixing between the two fluids does occur. (Usually with fixed ALE one shows the final time $t = 7$ or 8 before stagnation occurs. In paper [26] we have also presented intermediate times from $t = 8$ to $t = 15$ showing the efficiency of ReALE). The density, mesh, vorticity and cells colored by initial location are shown.

The paper [27] entitled *"ReALE : a Reconnection Arbitrary-Lagrangian-Eulerian method in cylindrical geometry"* has extended the ReALE concept to cylindrical geometry. In this paper we have shown that the whole ReALE concept does adapt to an already existing ALE code in cylindrical geometry (CHIC code in this case). Several test cases in cylindrical geometry have been run to assess this point, Sedov blastwave, an Helium/bubble shock interaction test (compared with experimental

results), and the rise of a light bubble under gravity. In Fig. 2.15 one reproduces the results on the Helium/bubble shock interaction test case for different times (the test case is depicted on the very next figure). In Fig. 2.16 one reproduces the rise of a light bubble under gravity (density, vorticity, velocity vector and mesh are displayed). The velocity vectors clearly show that due to high vorticity mesh reconnection occurs. Details and comments are to be found in paper [27].

The papers [26, 27] are reproduced in the following pages.

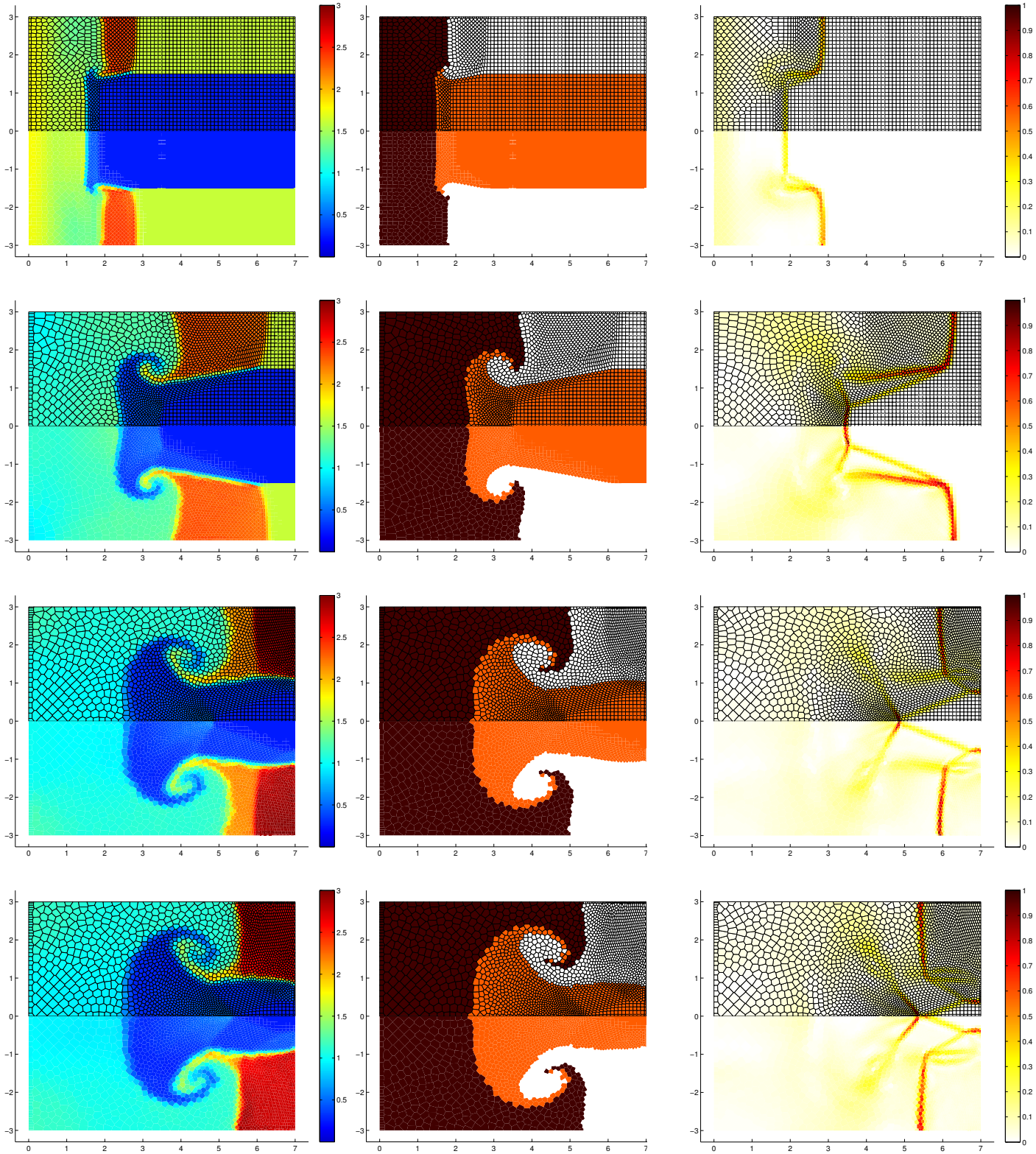


Figure 2.13 – *Numerical results from paper [26]. Triple point problem at several times for ReALE for internal energy and mesh (left) and cell color corresponds to the initial position in the domain (middle) and factor ω_c (right). From top to bottom : times $t = 1, 3, 4.5, 5$. $t = 1$ corresponds to the failure time for the Lagrangian version of the code, $t = 4.5$ roughly corresponds to the mesh stagnation time for the fixed ALE version although the ALE code runs until completion.*

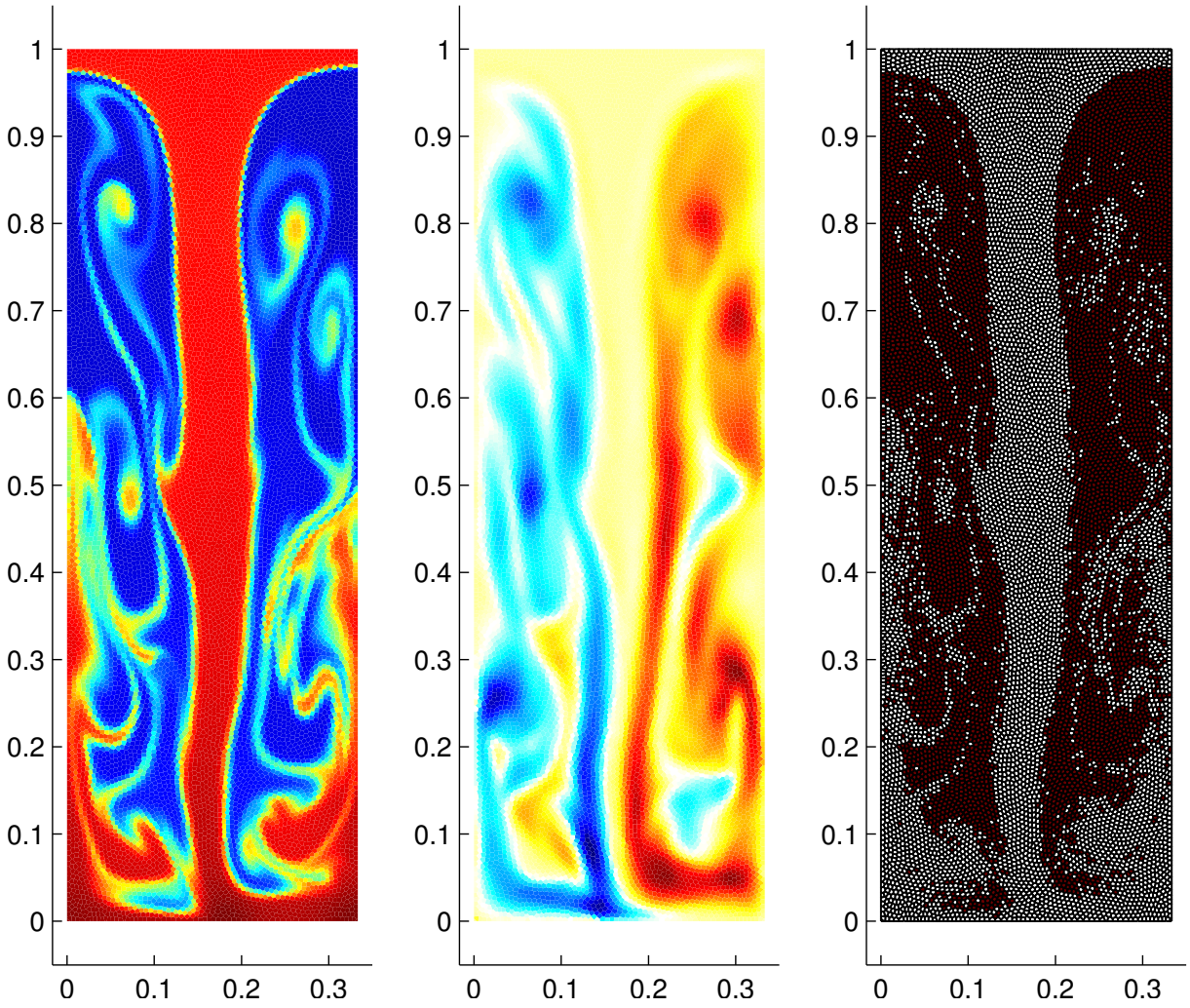


FIGURE 2.14 – *Numerical results from paper [26]. Rayleigh-Taylor instability with CHIC code — ReALE with* 66×200 *generators at* $t = 15$ *— Density, vorticity and, mesh and cells colored from their initial domain (red : from top-heavy fluid, white : from bottom-light fluid). Vorticity scale is from* -11.78 *to* 9.63 *(blue for the minimal value to red for the maximal one).*

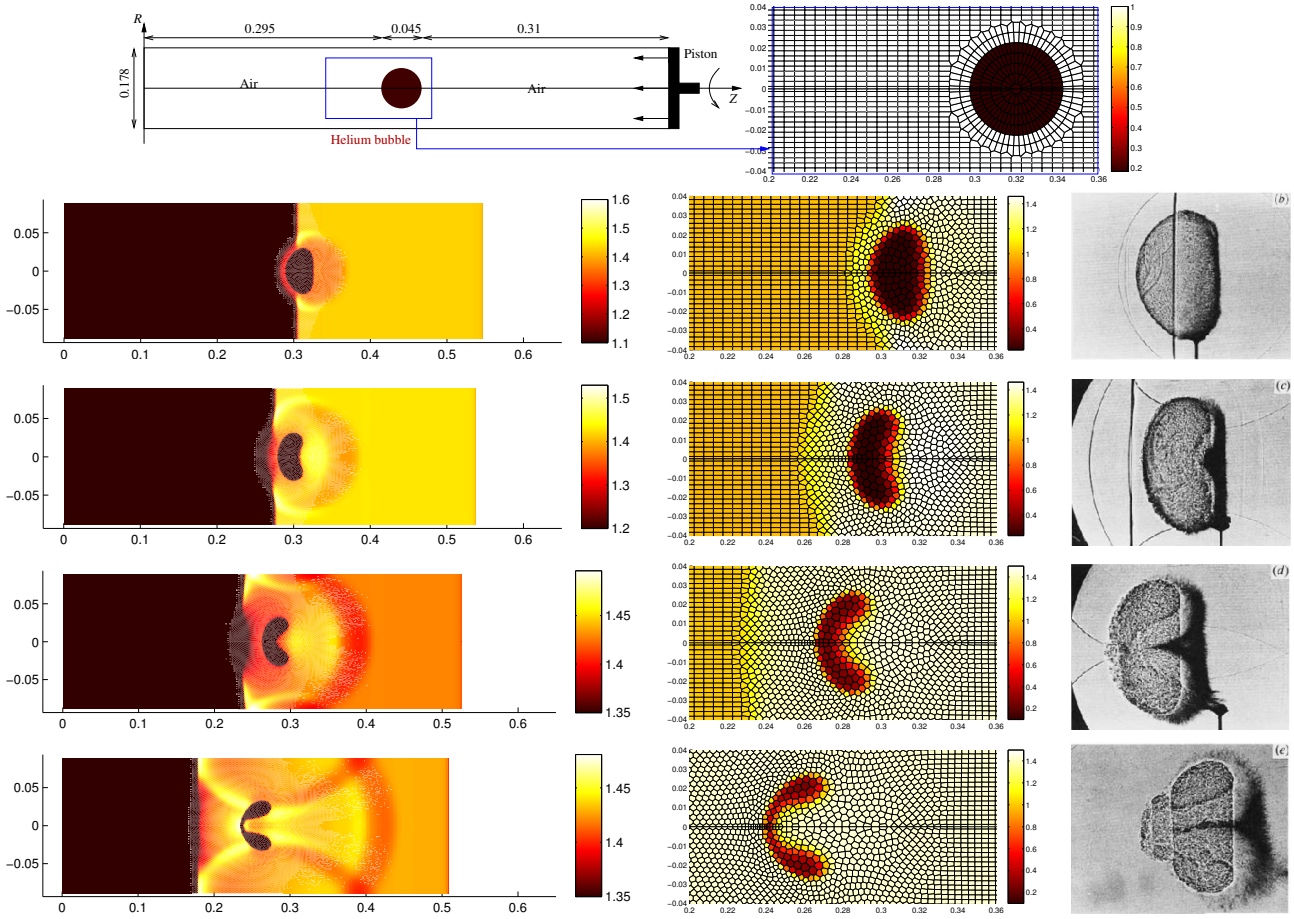


Figure 2.15 – *Numerical results from paper [27]. ReALE in cylindrical geometry results on the $M = 1.25$ shock interaction with a spherical helium bubble — Left : density waves in the domain. Middle : zoom on mesh and density (color) around the bubble. Right : Schlieren graphics (experimental results from [174]) — From top to bottom : $t_b = t_i + 82\ 10^{-6}$, $t_c = t_i + 145\ 10^{-6}$, $t_d = t_i + 223\ 10^{-6}$, $t_e = t_i + 1007\ 10^{-6}$ where $t_i = 657.463\ 10^{-6}$ is the time of the shock/bubble interaction.*

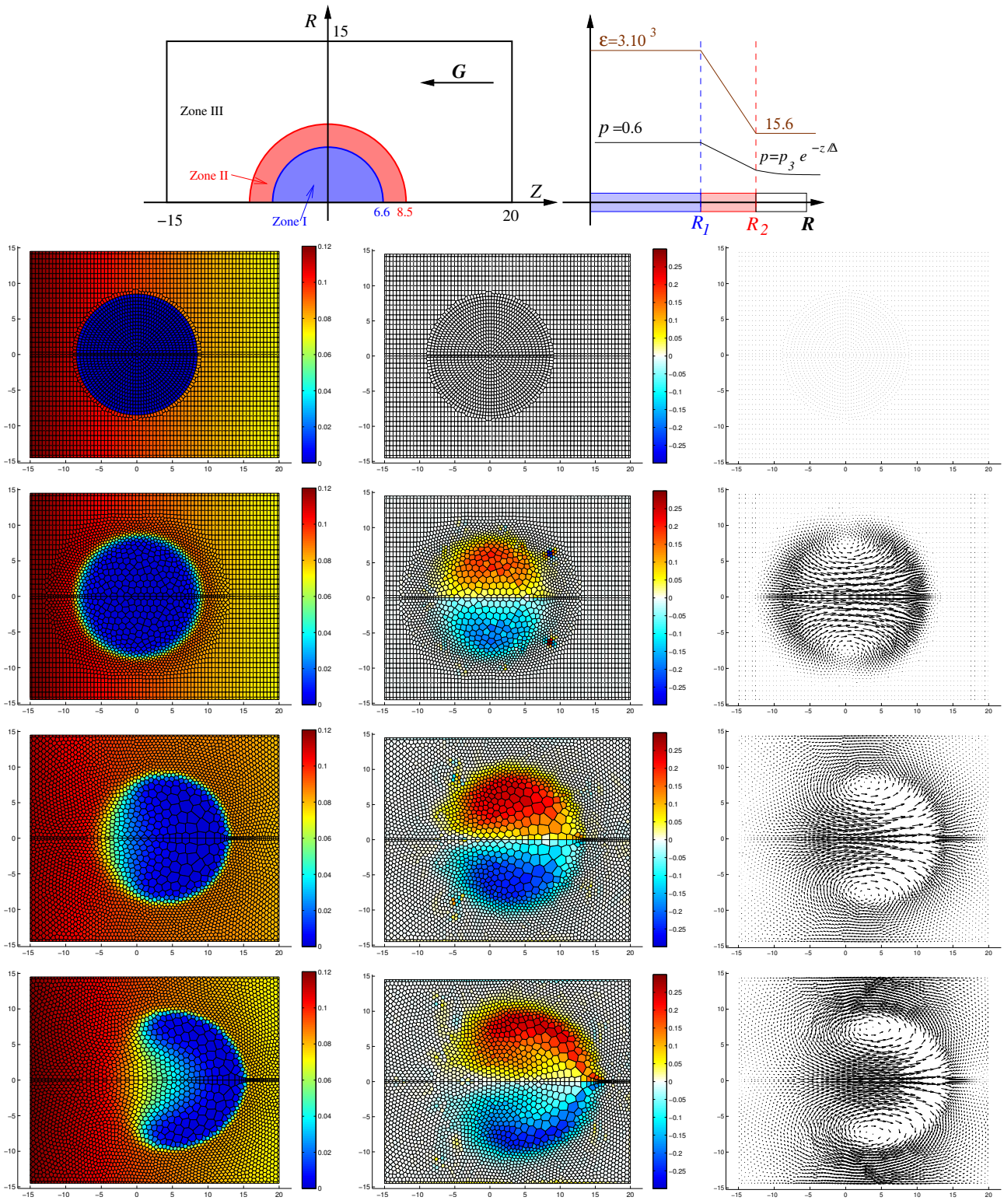


FIGURE 2.16 – *Numerical results from paper [27]. Top sketch : rise of a light bubble under gravity, the light bubble (Zone I) has a radius of R_1 and a transition layer is initialized between R_1 and R_2 (Zone II). The rest of the domain $\boldsymbol{R} > R_2$ is some air at rest (Zone III) where $\boldsymbol{R} = \sqrt{R^2 + Z^2}$. Gravity is oriented in the Z direction. The pressure and internal energy profiles are sketched in the right panel — 3 × 3 panels : numerical results from ReALE in cylindrical geometry — Left column : density and mesh — Middle column : vorticity and mesh — Right column : velocity vectors — Top-bottom : time $t_0 = 0$, $t_1 = 1$, $t_2 = 8$, $t_4 = 14$.*

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

ReALE: A reconnection-based arbitrary-Lagrangian–Eulerian method

Raphaël Loubère [a], Pierre-Henri Maire [b], Mikhail Shashkov [c,*], Jérôme Breil [b], Stéphane Galera [b]

[a] Institut de Mathématiques de Toulouse, CNRS, Université de Toulouse, 118 Route de Narbonne, 31062 Toulouse cedex 9, France
[b] UMR CELIA, Université Bordeaux I 351, Cours de la Libération, 33405 Talence, France
[c] Los Alamos National Laboratory, T-5, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 15 December 2009
Received in revised form 4 March 2010
Accepted 9 March 2010
Available online 18 March 2010

Keywords:
Lagrangian hydrodynamics
Cell-centered scheme
Compressible flow
Staggered scheme
Voronoi mesh
Arbitrary-Lagrangian–Eulerian
Mesh reconnection
Multi-dimensional unstructured polygonal mesh

ABSTRACT

We present a new reconnection-based arbitrary-Lagrangian–Eulerian (ALE) method. The main elements in a standard ALE simulation are an explicit Lagrangian phase in which the solution and grid are updated, a rezoning phase in which a new grid is defined, and a remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) onto the new grid. In standard ALE methods the new mesh from the rezone phase is obtained by moving grid nodes without changing connectivity of the mesh. Such rezone strategy has its limitation due to the fixed topology of the mesh. In our new method we allow connectivity of the mesh to change in rezone phase, which leads to general polygonal mesh and allows to follow Lagrangian features of the mesh much better than for standard ALE methods. Rezone strategy with reconnection is based on using Voronoi tessellation. We demonstrate performance of our new method on series of numerical examples and show it superiority in comparison with standard ALE methods without reconnection.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

In numerical simulations of multi-dimensional fluid flow, the relationship of the motion of the computational grid to the motion of the fluid is an important issue. One of two choices is typically made: a Lagrangian framework or an Eulerian framework. In the Lagrangian framework, the grid moves with the local fluid velocity, while in the Eulerian framework, the fluid flows through a grid fixed in space. More generally, the motion of the grid can be chosen arbitrarily. The philosophy of the arbitrary-Lagrangian–Eulerian methodology (ALE; cf. [64,19,20,70,71,81,100,93]) is to exploit this degree of freedom to improve the accuracy and efficiency of the simulation. The main elements in an standard ALE simulation are an explicit Lagrangian phase in which the solution and grid are updated, a rezoning phase in which a new grid is defined, and a remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) onto the new grid [93]. Clearly ALE also includes Lagrangian approach when the mesh is not changing during rezone phase. It is important to note that Eulerian simulation can be considered as a limiting case of ALE when the rezoned mesh coincides with the mesh at the beginning of the Lagrangian step – such implementation of Eulerian methodology is usually called Eulerian-as-Lagrange-Plus-Remap [123,101] as opposed to direct Eulerian approach [38] in which gas dynamics equations in Eulerian form are directly solved. Therefore, ALE methodology is very flexible and successfully used to solve complicated problems.

The numerical methods related to Lagrangian phase of ALE are relatively well developed (for example [20,28,29,18,31,83,30,32,40,33,109,108,110,34,91,16,104,88,89,87]), as well as the remapping phase (being considered as conservative interpolation or advection) [20,94,77,61,84,101,124,49,21,48,98].

* Corresponding author.
E-mail addresses: raphael.loubere@math.univ-toulouse.fr (R. Loubère), maire@celia.u-bordeaux1.fr (P.-H. Maire), shashkov@lanl.gov (M. Shashkov), breil@celia.u-bordeaux1.fr (J. Breil), galera@celia.u-bordeaux1.fr (S. Galera).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.03.011

Our opinion is that the most difficult and least developed phase of ALE is the rezoning phase. A review of existing rezone strategies for ALE methods is presented in [74] including analysis of alternative approaches [81,120,45,19,57,100,24,129,9,47,128]. A review of a more general class of methods, namely moving mesh methods, is presented in [79].

Ideally the mesh has to adapt to the solution. Any adaptive scheme is composed of three main ingredients: an optimal-mesh criterion, an error estimator or error indicator, and an algorithm of the strategy for the mesh improvement. These ingredients answer to the following questions: How should the optimal mesh be defined? Where are mesh changes required? And how should the improved mesh be constructed? For standard ALE methods a strategy for mesh improvement is based on moving the spatial grid. Generally speaking the goal of rezoning is to improve the efficiency of the ALE method, that is, to achieve a given accuracy with the least amount of "work". "Work" in this context should be understood not only as the CPU time but also as memory and man-hour resources. However, to design an adaptive method one needs a quantitative assessment of optimality. The problem is that, for non-linear equations of gas dynamics in 2D and 3D, at the moment, it is not feasible to obtain such quantitative assessment. For this reason practitioners are usually using some qualitative approaches. In real complex ALE simulation the most basic goal of rezoning is simply to run calculation to completion without user intervention and still achieve reasonable accuracy (recall that we always can run ALE in Eulerian = Lagrange-Plus-Remap mode, which will be robust but less accurate). Even this goal is usually not achieved in most production ALE codes. For example, even for very popular methods based on Winslow smoothing, [128,129], practical simulations require the introduction of numerous geometrical and physics-based triggers and lockers, that is, mesh constraints that typically keep a node Lagrangian until some condition is reached e.g. element quality criterion (to detect cell distortion or collapse) or physical condition is reached in surrounding elements (for example, did the cell fully detonate?) [100,14,15,65].

As it is mentioned in [14,15], the mesh movement philosophy applied to most applications, related to high-speed multi-material flows, is to develop algorithms that will move the mesh in such a way as to maintain robustness while staying as close as possible to the Lagrangian mesh motion. The Lagrangian mesh motion naturally follows most flow features of interest such as shocks, material interfaces and steep gradients and allows users to focus zoning in materials of interest. Mesh relaxation is then used in regions of high material deformation to improve mesh quality. In standard ALE methods, which use fixed mesh topology, nodes are moved to refine mesh some areas of the problem at the expense of coarsening mesh in other parts of the problem. Generally, the increase of mesh resolution is limited, and, most importantly it can degrade the mesh quality leading to robustness problems. One of the most cited papers in mesh rezoning is [24], where authors use variational approach to combine requirements related to maintaining geometric quality of the mesh and some mesh adaptation based on equidistribution of some error indicator. Functional which is responsible for mesh smoothness essentially can be considered as a variational form of Winslow approach for which the corresponding optimization problem is well behaved. In contrast, the functional responsible for error equidistribution, if used by itself, has multiple local minima and its minimization can lead to tangled mesh. A difficulty arises when one tries to combine these – how should one weight the relative importance of these separate goals and still obtain a well behaved optimization process? In particular, the two global functionals have distinct (physical) dimensions, and so, can only be combined with some dimensional constant. At present, there is no theoretical basis for choosing this constant, thus delegating the decision to the user. In practice, bad choice of this parameter can lead to loss of accuracy (if the mesh is over-smoothed) or robustness problem (if the mesh becomes tangled).

In our opinion, the main reason for this is that standard ALE codes utilize a fixed topology mesh, defined at the outset, which in general will not be able to adapt to the dynamically evolving interface shape (or contact discontinuity) in spite of efforts at regularization (see Section 2 for more details). The most general solution to this difficulty, while preserving a Lagrangian nature of mesh motion is to relax the constraints on mesh topology and allow reconnection.[1]

The idea of using mesh reconnection to solve partial differential equations is not new. To the best of our knowledge, in context of computational gas dynamics the ideas related to mesh reconnection were first used in [99].[2] In this seminal paper the authors suggest to use a set of point Lagrangian particles and surround them with domains (parcels) to describe the media. The shape and size of these parcels are determined by the positions of the particles. The connectivity of the set of particles is not fixed but can vary with time depending on relative positions of the particles. After connectivity is established the set of neighboring particles defines the stencil on which Lagrangian equations are discretized.

In our opinion paper [99] has all basic ideas that lead to development of so-called free-Lagrange (or free-Lagrangian) methods, [55,23,121], which were very popular in 80s and early 90s The name free-Lagrange was introduced in [39] and the corresponding code was called FLAG. We need to analyze the main features of free-Lagrange approach in order to precisely position our new reconnection-based ALE method with respect to well established methods.

We start our analysis with methods, which we will call *pure* free-Lagrangian methods, [95,122,54,114,113,107,10]. In these methods, the fluid is represented by point particles surrounded by parcels. Each particle and its corresponding parcel represent a single material, that is, there is no mixed (containing several materials) parcel. Mass exchange between particles is forbidden. Flow variables are stored at the particle within each parcel, which moves in a "Lagrangian" fashion. The shape of the parcel is determined by constructing Voronoi mesh [96], in which each particle (generator, site) is en-

[1] Let us also mention interesting approach of combining standard ALE method with adaptive mesh refinement (AMR) [3]. In this, approach authors allows subdivision of the cell into smaller cells similar to standard Eulerian AMR methods. Formally, it can be considered as connectivity change. Our opinion is that objective of ALE AMR is to increase local resolution. This method has all drawbacks of standard ALE methods with fixed connectivity.

[2] In this paper we are not considering meshless methods, like smoothed particle hydrodynamics – SPH – interested readers can refer for example to [23,80,66].

closed within convex polygonal cell. By definition, in a Voronoi mesh each cell encloses all points in the domain which are closer to the corresponding particle than to any other particle. The mesh is fully reconstructed after some number of time steps to allow grid connectivity to change naturally under influence of shear or vorticity. Connectivity of Voronoi mesh uniquely defines the "neighbors" that is particles which parcels share edge with given particle. These neighbors define the stencil on which gas dynamics equations are approximated. In "pure" free-Lagrange, one approximates some form of Lagrangian equations. Several discretization approaches are possible, for example, mimetic finite difference [95], or Godunov-type methods [10]. The energy equation can be approximated in conservative form [10], or in non-conservative form as in [95]. In later case mimetic approach is used to guarantee conservation of total energy. There are also different approximations used for continuity equation. Most standard approach is to define density as ratio of particle mass (which is constant in time) and volume of the corresponding parcel. However, one also can define the volume from a special evolution equation as it is usually done in Godunov based methods. Let us emphasize that there is no "remapping" phase in "pure" free-Lagrangian methods.

Now let us discuss how pure are "pure" free-Lagrangian methods. Such analysis requires the definition of what is a Lagrangian method. One can define a Lagrangian method as a method where equations in Lagrangian coordinates are approximated. That is to say, governing equations do not have advective terms; in other words no exchange of mass, momentum and energy between these particles is allowed. In case of "pure" free-Lagrangian methods these two criteria are satisfied. But this is not enough to call method Lagrangian. For a Lagrangian method the boundary of parcel has to move in Lagrangian fashion too. Unfortunately, for "pure" free-Lagrange it is not the case because parcels are Voronoi cells and the vertices of the Voronoi cell are not moved independently in a Lagrangian way but defined from positions of the particles in a very non-linear way [95]. It is well known [55,23,121] that a Voronoi cell is not a Lagrangian object and it is rigorously proven in [118] that for "pure" free-Lagrangian methods discrete continuity equation is not consistent (order of approximation is zero) with continuous continuity equation; it also means that the so-called Geometric Conservation Law (GCL), [53,87], which can affect stability of the method, [53], is not fulfilled. It usually implies that small changes in the particle positions (especially if two particles are very close to each other) can lead to very large change of the volume. Non-Lagrangian behavior of Voronoi cell dramatically decreases accuracy near contact discontinuity. There are several approaches which intend to mitigate non-Lagrangian behavior of Voronoi cells. For example, in [95,114] special type of artificial viscosity was introduced which helps to regularize the particle positions and do not allow particles to be to close to each other. In [10] some corrections to velocity of generators are introduced and in [67] special artificial forces are introduced to correct shape of parcels. Another approach is to keep Voronoi mesh inducing connectivity, but define parcel shape using so-called median mesh [107], where the coordinates of vertices are defined by simple interpolation of a particle position and its two consecutive neighbors. It leads to non-convex shape of parcels and abrupt change in parcel volume when the connectivity of the mesh is changing (in contrast volume and shape of Voronoi cells are continuous in time, but are not Lagrangian objects). This clearly contradicts another important property of Lagrangian parcel: its shape is supposed to change continuously in time including instances when connectivity changes. Because of abrupt change of volume of the parcel some remapping is needed. The conclusion of this analysis is that even pure-Lagrangian method are neither "pure" nor "Lagrangian". Still as demonstrated, for example in [10,67,12,68,11,114] these types of methods can be very successful for particular problems, but each problem requires some special strategy to mitigate non-Lagrangian nature of Voronoi cells.

There were several other "free-Lagrangian" methods, which differ in how connectivity of the mesh is established, spatial centering of flow parameters (particle centered, parcel centered, staggered), what form of equations are used, what type of discretization is used, implicit or explicit, some of the methods having explicit remap phase, and so on. Interested reader can refer to the following papers [51,39,50,115,35,36,63,126,6,97,13,5,56]. Very educational are the roundtable discussions published in [55,121]. The conclusion is that these "free-Lagrangian" methods are even less Lagrangian than "pure" free-Lagrangian methods, which we have analyzed in previous paragraph. However, all of these methods allow change of connectivity of the mesh with time.

Our belief is that because free-Lagrangian methods are not really Lagrangian then, explicitly or implicitly, they incorporate a rezone phase, and consequently, a corresponding remap phase. Some of them explicitly states this remap phase. Nevertheless if the free-Lagrange method does not have a remap phase, errors related to it will manifest itself one way or another.

It leads us to the conclusion that methods where connectivity of the mesh can change have to be developed in reconnection-based ALE (ReALE) framework, where rezone stage includes reconnection. Let us note that similar philosophy was used in [45,1,37], even so authors of [45,1,37] do not call their method ALE or free-Lagrange.

Let us finally note that when we were writing this paper we became aware of the paper [119]. In this paper the author proposed a new formulation of continuum hydrodynamics based on an unstructured grid. In this work the mesh is defined as the Voronoi tessellation of a set of discrete mesh-generating points. The gas dynamics equations are solved using a Godunov-type finite-volume scheme on a moving Voronoi mesh. This approach corresponds to a direct ALE strategy (no explicit Lagrangian, rezone and remap phases-governing equations are written in moving coordinate system) in which the motion of the grid is ruled by prescribing the velocities of the mesh generators. The numerical results obtained with this method are quite impressive and display its robustness and accuracy. However, we note that this method may suffer from several limitations. First, this is by construction a mono-material formulation since the author is using a direct ALE strategy. We believe that its multi-material extension is far from being obvious. We also note that this ALE scheme does not fulfill GCL requirement. This flaw can lead to severe problems.

As standard for ALE method, the main elements in ReALE simulation are an explicit Lagrangian phase in which the solution and grid are updated (without changing connectivity), a rezoning phase in which a new grid is defined (which includes changing connectivity and also adding or deleting cells or vertices of the parcels), and a remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) onto the new grid. Flowchart of the entire ReALE algorithm is presented in Section 5. For ReALE all three phases are supposed to satisfy specific requirements which are different from standard ALE methods.

We assume, that at the beginning of the calculation ($t = 0$) as well at the beginning of each time step (after rezone phase) the computational mesh consists of Voronoi cells corresponding to some set of particles (generators, sites), that is, distribution of generators defines the mesh. Initialization of the mesh as well as necessary definitions related to Voronoi diagrams are given in Sections 3 and 4.

Because of reconnection in rezone phase, the Lagrangian phase of the ReALE method has to deal with discretization of the Lagrangian equations on general polygonal meshes and corresponding update of this polygonal mesh is supposed to be Lagrangian. There are several papers dealing with discretization of Lagrangian equation on general polygonal meshes [25,27,26,45,116,117]. In this paper we will use compatible mimetic finite discretizations [28,29] on staggered mesh, which historically close to [25,27,26] and newly developed cell-centered discretizations based on Godunov approach [91,89,87], which satisfy GCL (contrarily to [45], which does not satisfy GCL). Necessary details about discretizations used in Lagrangian phase of our ReALE methodology are presented in Section 6.

The rezone phase of ReALE has to include both mesh movement and reconnection procedure. In this paper we used a set of particles (generators) and the machinery of Voronoi diagrams to do both mesh movement and mesh reconnection. That is our rezone strategy consists of a special movement of generators. It is close to Lagrangian in some sense, but also include some smoothing procedure based on notion of centroidal Voronoi diagrams [43]. The rezone phase of our new ReALE approach is described in Section 7.

In the remapping phase, the Lagrangian solution is transferred (conservatively interpolated) onto the rezoned mesh. Lagrangian mesh is the result of one time step Lagrangian movement of the Voronoi mesh corresponding to the distribution of the generators at the previous time step. The new rezoned mesh is the Voronoi mesh corresponding to the positions of generators created by the rezone phase. During the rezone phase generators are moved in an "almost" Lagrangian way and because Voronoi cells are changing their shape continuously with respect to positions of the generators; rezoned and Lagrangian meshes are "close". However, in general, the connectivity of the Lagrangian and rezoned mesh are different. Consequently remapping methods have to be able to conservatively transfer flow parameters from one polygonal mesh to another. In this paper intersection (overlay) based remap is used [61,45,94,77], however, one can take advantage of how Lagrangian and rezoned meshes are constructed and design more efficient methods [76]. The remapping phase is described in Section 8.

We demonstrate the performance of our method on a set of numerical tests presented in Section 9. Finally, conclusions and future work are summarized in Section 10.

2. Motivation

To motivate our research let us consider the Rayleigh–Taylor instability problem. It consists of two ideal gases with initial densities $\rho_{top} = 2$ and $\rho_{bottom} = 1$; in both cases the adiabatic constant is $\gamma = 1.4$. Initially, the heavier gas is above the lighter gas in a rectangular vessel $[0:1/6] \times [0:1]$, with gravitational field directed vertically downward and with magnitude $g = 0.1$. The interface has been deliberately perturbed as described by formula $y_i(x) = 1/2 + 0.01\cos(6\pi x)$. Initially both gases are at rest; the pressure distribution is approximately hydrostatic and is defined in the lighter gas

$$P = 1 + \rho_{top} g 0.5 + \rho_{bottom} g(0.5 - y),$$

and in the heavier as

$$P = 1 + \rho_{top} g(1 - y).$$

It is well known that such configuration is unstable and as time progresses, the heavier gas will sink and the lighter gas will rise through the formation of bubbles and spikes. Further details of the general theory of Rayleigh–Taylor instabilities can be found in [78]. The time evolution of this problem leads to a rollup of the interface and the generation of significant vorticity. This problem is poorly suited for Lagrangian methods, and is usually tackled using Eulerian or ALE techniques. As discussed in Section 1, for standard ALE methods the mesh does not change connectivity. Improvement of the mesh on the rezone stage is achieved only by moving nodes. Here we consider a standard ALE approach based on cell-centered Lagrangian method described in [90,87], which uses Winslow approach at rezone phase [128]. In Fig. 1(a) we present meshes for time moments $t = 7, 8, 9$; in Fig. 1(b) we present vorticity color map at the same time moments. On Lagrangian stage of ALE the mesh is trying to follow the flow, but because of development vorticity it eventually leads to a tangled mesh. On rezone stage the mesh is slightly relaxed, however on the next time step of the Lagrangian phase, the mesh is trying to follow vorticity development and, again, is approaching a tangling situation. The meshes in Fig. 1(a) are the result of these competing processes.

In standard ALE method the mesh cannot change connectivity, and mesh with fixed connectivity has some limitation in how much it can deform. In Fig. 2 we present fragment of the mesh at $t = 8$ (panel a) as well as only "horizontal" (panel b)

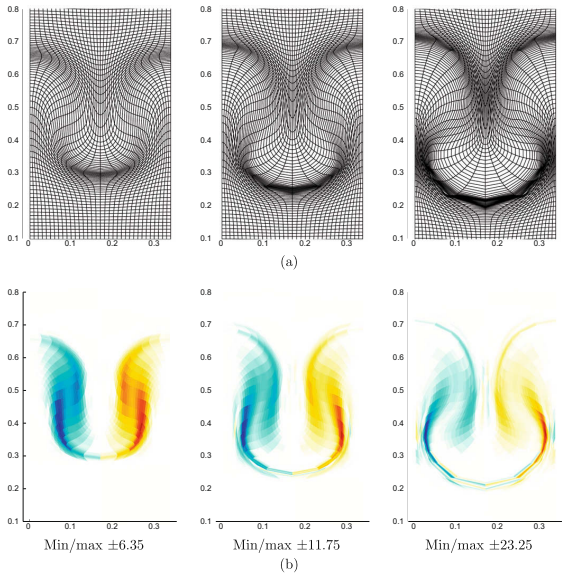


Fig. 1. Rayleigh–Taylor problem at time moments $t = 7, 8, 9$ – (a) Mesh fragments. (b) Vorticity color map. Color scale is from blue (minimal negative vorticity) to red (maximal positive vorticity). White color corresponds to zero vorticity. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and only "vertical" (panel c) logical lines of the mesh. From this figure it is clear that logically rectangular mesh cannot deform much further in regions where vorticity is developing. At some time moment the mesh eventually stagnates in subregions where vorticity is still developing. Similar pictures can be found in [74] for ALE using staggered discretization and reference Jacobian rezone strategy.

This stagnation means that in subregions where vorticity is developing ALE method with fixed connectivity actually becomes Eulerian in its Lagrange + Remap form. It leads to excessive smoothing of flow parameters at remapping stage, and eventually to loss of accuracy in these regions. Moreover, because the mesh is logically rectangular, stagnation of the mesh in some regions leads to locking of the mesh in other regions, resulting to loss of overall accuracy. Also stagnated mesh has a low geometrical quality which additionally contributes to the loss of accuracy. One clearly sees that behavior of vorticity presented in Fig. 1(b) is not physical because it oscillates from positive to negative at neighboring cells, which is especially pronounced at $t = 9$. We believe that the resolution of this problem lies in allowing reconnection during the rezone stage of ALE method. However, allowing reconnection at rezone phase has its implications on all phases of ALE. First of all, we need to decide what is the mechanism of the reconnection. In this paper we use Voronoi tessellation machinery.

3. Voronoi tessellation

Let us consider a convex computational domain Ω in 2D. For a set of generators $\mathbf{G}_c = (x_c, y_c) \in \Omega$ the Voronoi cell Ω_c is defined as follows [96]:

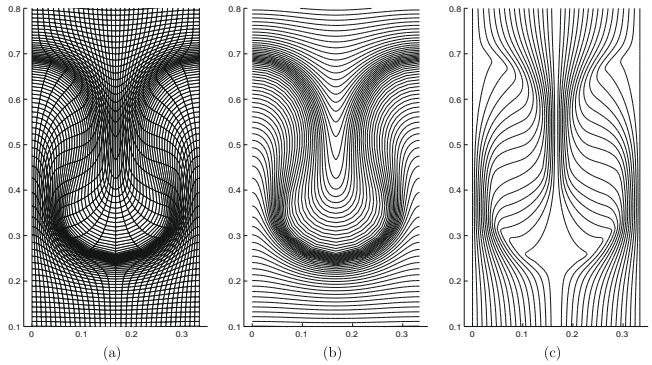


Fig. 2. Fragment of mesh at $t = 8$: (a) Mesh fragment. (b) "Horizontal" logical lines. (c) "Vertical" logical lines.

$$\Omega_c = \{\mathbf{X} = (x, y) \in \Omega : |\mathbf{X} - \mathbf{G}_c| \leqslant |\mathbf{X} - \mathbf{G}_{c'}|, \quad \text{for all } c \neq c'\} \tag{1}$$

Voronoi cell Ω_c is a convex polygon, and the set of Voronoi cells defines the tessellation of Ω, that is, it covers Ω without holes or overlaps. There are several generalizations of the definition of Voronoi cell for non-convex domains, like bounded and constrained Voronoi diagrams [7,8,72], VPS (visibility shortest path) Voronoi diagrams [96, pp. 163–156]. In this paper we will only consider convex computational domains and will not describe these generalizations. We use an incremental algorithm for the construction of Voronoi tessellation which is described in [113]. In general any other available algorithm can be used. In Fig. 3(a) we present Voronoi cells in the unit square which correspond to generators marked by ×. For now on, a cell Ω_c is referred to with its unique index c. By definition cell $\tilde{c}$ is a neighbor of cell c if it shares a face with it. The set of neighbors of cell c is denoted by $\mathcal{C}(c)$. The set of vertices of cell c is denoted by $\mathcal{P}(c)$. Any vertex of a Voronoi mesh is shared by three cells only. The set of faces of cell c is denotes as $\mathcal{F}(c)$, each face shares only by two cells. These relationships completely define the connectivity of the mesh.

Depending on the position of generators, the Voronoi mesh can be genuinely non-uniform. One of the possible measures of non-uniformity is how far is the centroid of a Voronoi cell from the generator corresponding to this cell. Let us introduce, $\boldsymbol{X}_c$, the centroid of the cell Ω_c as follows:

$$\boldsymbol{X}_c = \frac{1}{|\Omega_c|} \int_{\Omega_c} \boldsymbol{X} dV,$$

where $|\Omega_c|$ denotes the volume of the cell Ω_c.

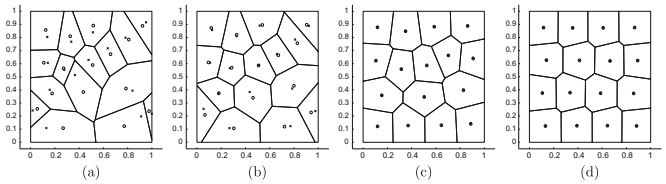


Fig. 3. Lloyd's algorithm to smooth Voronoi mesh: (a) Initial mesh – (b) mesh after one iteration – (c) mesh after 10 iterations – (d) final "converged" mesh – Crosses × correspond to positions of the generators. A circle ○ is the position of the centroid of the Voronoi cell which corresponds to a unique generator.

In all panels of Fig. 3 positions of centroids are marked by circles ○ and position of generators are marked by crosses ×. Clearly, the closer the centroids and the generators are, the more uniform the mesh is. It brings us to an important new definition: centroidal Voronoi tessellation (CVT) [43]. Voronoi tessellation is called centroidal Voronoi tessellation if the position of the cell centroid coincides with the position of the corresponding generator. CVT type of meshes are asymptotically (as number of generators goes to infinity) made of perfect hexagons inside the computational domain. Meshes of CVT type have several attractive properties, in particular discretizations of partial differential equations are usually more accurate on such meshes [69].

There is a very simple algorithm to create CVT depicted in Fig. 3. It starts with an arbitrary distribution of generators, Fig. 3(a), and constructs its corresponding Voronoi tessellation. Then it computes centroids of the constructed Voronoi cells and uses them as generators for a next iteration. The resulting mesh after one iteration is presented in Fig. 3(b). Let us note that regularity of the mesh is visibly improved. The mesh after 10 iterations is presented in Fig. 3(c). Finally the converged mesh is presented in Fig. 3(d). This algorithm is called Lloyd's method, readers can refer to [43,42] for more details.

For the purposes of our paper one iteration of Lloyd's algorithm can be considered as a mesh smoothing step, which, in some sense, is analogous to a Winslow iteration for meshes with fixed connectivity. However, in Lloyd's algorithm connectivity of the mesh may change at each iteration.

4. Initialization

Modeling of any problem starts with the creation of an initial mesh. In our approach, an initial mesh is created by the distribution of generators and the construction of the associated Voronoi mesh. As with any ALE method the initial mesh reflects the knowledge about the underlying physical problem, for example, initial distribution of materials, direction and shape of the main shocks and other important features of the flow. Let us repeat that the initial mesh has to be Voronoi mesh and has to be consistent with rezone strategy. "Consistent" means that the initial mesh has to be constructed in such a way that it will not change dramatically at rezone stage. This issue will be clarified in Sections 7 and 9. Another important issue is that, as for any ALE method, we want to track material boundaries as close as possible. Therefore, initially, we are trying to create a mesh, the cell faces of which coincide with material boundaries. Let us demonstrate some ideas on a simple example of a bubble containing one material in a rectangular computational domain filled with another material, see Fig. 4 (this is a fragment of the initial mesh for the shock–bubble interaction problem described in details in Section 9.3). For this problem it is important to have an orthogonal mesh with mesh lines aligned with coordinate directions outside the bubble because a vertical shock will approach the bubble from the right. As a consequence the shock direction will be aligned with the mesh. A square mesh is the degenerate Voronoi mesh obtained with a regular distribution of generators; these are located at the centers of the squares. In order for the boundary of the bubble to be represented by faces of Voronoi cells, the generators have to be located on the same distance from the boundary in orthogonal direction. In this case because the boundary is a circle of some radius r, generators are located also on concentric circles of smaller, $(r - \Delta r)$, an bigger, $(r + \Delta r)$ radii. Inside the bubble, generators are located on circles, which radii are equally distributed between 0 and r. On each circle we put a different number of generators such that the arc length between generators on each circle is approximately the same for all circles. It then creates a pseudo polar mesh. This mesh is a little bit extended outside the bubble. Technically to combine a square mesh and a pseudo polar mesh we first create generators which produce the square mesh, then generators which are located in disk of radius $r + 2\Delta r$ are removed and replaced by generators corresponding to the pseudo polar mesh. The resulting mesh is presented in Fig 4.

Similar techniques can be applied for fitting more complicated shapes, some ideas are presented in [102].

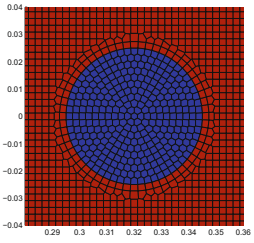


Fig. 4. Mesh for a bubble in a rectangle.

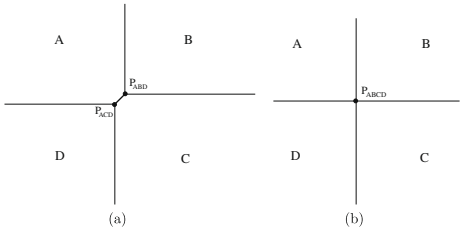


Fig. 5. Cleaning of small edges: (a) before cleaning and (b) after cleaning.

In some cases the faces of a Voronoi cell can be very small. In degenerate cases like a rectangular Voronoi mesh, some edges actually have zero length. This situation is schematically presented in Fig. 5(a). In this figure the edge $P_{ACD}\,P_{ABC}$ has an almost zero length. Time step control for Lagrangian phase of ReALE method is based on minimal edge length over the cell, see Section 6. Therefore, the presence of small edges can dramatically reduce the time step. The solution lays in the introduction of a "cleaning" step that removes edges of the cell, which are small in comparison with a local characteristic length.

In situation presented in Fig. 5 cell A has neighbors $\ldots B, C, D \ldots$. Cells D and B are not a neighbors. Removing edge $P_{ACD}\,P_{ABC}$ is equivalent to collapsing the two points into one point P_{ABCD} (the midpoint of segment $[P_{ACD}, P_{ABC}]$) schematically presented in Fig. 5(b). It means that point P_{ABCD} is now shared by four cells A, B, C, D. Cells A and C do not share any edge, and cells A and C do not have zero length edge anymore. We call this process small-edge cleaning. Cleaning is applied not only to zero length edges but to all edges which are small in comparison with the local characteristic length. More details on cleaning is given in Section 7. It is important to note that after the cleaning step, the computational mesh is not a Voronoi tessellation anymore. In particular each point can be shared by more than three cells.

5. Flowchart of ReALE method

In Fig. 6 we describe the flowchart of ReALE method. Initialization stage is described in previous section. The result of this stage is the mesh $I^{n=0}$, which, in general, is an unstructured polygonal mesh obtained by the cleaning of a Voronoi mesh. On the initialization stage we also define the initial condition for the degrees of freedom $U^{n=0}$ related to each particular Lagrangian scheme, for example, for cell-centered discretization, these are density, velocity and pressure of the cell (any other variable being deduced from them). First, the Lagrangian step uses the mesh I^0. At time step n the Lagrangian scheme starts with the rezoned mesh obtained from the previous time step. On Lagrangian stage the mesh is moving with the flow. The result of the Lagrangian step is the Lagrangian mesh L^{n+1} and all physical quantities on this mesh – U_L^{n+1}. Recall that Lagrangian algorithm has to deal with general polygonal meshes. Let us note that in the Lagrangian stage the generators do not play any role. During the Lagrangian step the mesh does not change connectivity. This stage is described in Section 6.

On the rezone stage we define positions of generators $\mathbf{G}_c^{n+1}$ and construct the associated Voronoi tessellation. It is followed by a cleaning step. The result of the rezone phase is a general polygonal mesh R^{n+1}. In general, connectivity of mesh R^{n+1} is different from connectivity of mesh L^{n+1}. However, the number of cells is the same. Details of the rezone stage are provided in Section 7. For the remap stage we perform a conservative interpolation of flow parameters from Lagrangian mesh L^{n+1} onto rezoned mesh R^{n+1}. Because in general these two meshes are polygonal meshes with different connectivities, one needs to use an intersection (overlay) based remap method. The remapping stage is described in details in Section 8. Finally to start new time step we set I^{n+1} as being the rezoned mesh R^{n+1}. In other words the new Lagrangian step starts with the polygonal mesh obtained as result of rezoning from the previous time step and the physical variables U^{n+1} from the remap phase.

6. Lagrangian phase

In this section we present the discretization of the Lagrangian hydrodynamics over a general two-dimensional polygonal grid that will be the first phase of our ReALE algorithm. We present two discretizations, the first one is based on a staggered placement of the variables, whereas the second one is cell-centered. In Lagrangian hydrodynamics methods, a computational cell moves with the flow velocity. In practice, this means that the cell vertices move with a computed velocity, the cell faces being uniquely specified by the vertex positions. Thus, Lagrangian methods can capture contact discontinuity sharply in multi-material fluid flows. However, in the Lagrangian framework, one has to discretize not only the gas dynamics equations but

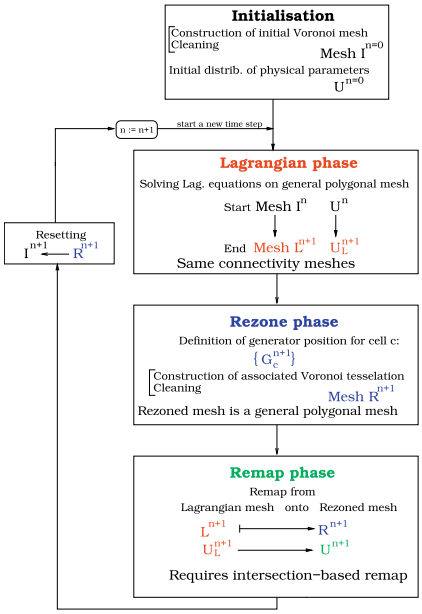


Fig. 6. Flowchart of ReALE Method.

also the vertex motion in order to move the mesh. Moreover, the numerical fluxes of the physical conservation laws must be determined in a compatible way with the vertex velocity so that the geometric conservation law (GCL) is satisfied, namely the rate of change of a Lagrangian volume has to be computed coherently with the node motion. This critical requirement is the cornerstone of any Lagrangian multi-dimensional scheme.

The most natural way to solve this problem employs a staggered discretization in which position, velocity and kinetic energy are centered at points, while density, pressure and internal energy are within cells. The dissipation of kinetic energy into internal energy through shock waves is ensured by an artificial viscosity term. Since the seminal works of von Neumann and Richtmyer [125], and Wilkins [127], many developments have been made in order to improve the accuracy and the robustness of staggered hydrodynamics [32,33,28]. More specifically, the construction of a compatible staggered discretization leads to a scheme that conserves total energy in a rigorous manner [30,29].

An alternative to the previous discretizations is to derive a Lagrangian scheme based on the Godunov method [60]. In the Godunov-type method approach, all conserved quantities, including momentum, and hence cell velocity are cell-centered. The cell-face quantities, including a face-normal component of the velocity, are available from the solution of an approximate Riemann problem at each cell face. However, it remains to determine the vertex velocity in order to move the mesh. In one of the first papers related to application of Godunov methods on general polygonal mesh in 2D [1], the flux computation was not compatible with the node displacement, and hence the GCL was not satisfied. This incompatibility generated additional spurious components in the vertex velocity field whose correction required a very expensive treatment [46]. An

important breakthrough concerning the compatibility between flux discretization and vertex velocity computation has been introduced in [40,88]. In these papers, authors present schemes in which the interface fluxes and the node velocity are computed coherently thanks to an approximate Riemann solver located at the nodes. This original approach leads to first-order conservative schemes which satisfy a local semi-discrete entropy inequality. The multi-dimensional high-order extension of these schemes are developed in [34,91,87,86].

In what follows, we recall briefly the main features of the two Lagrangian schemes, staggered and cell-centered, that will be used to construct our ReALE algorithm. The necessary details concerning the discretization can be found in previously published papers.

6.1. Governing equations

The gas dynamics equations, in Lagrangian form, write as follows:

$$\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) - \nabla \cdot \boldsymbol{U} = 0, \tag{2a}$$

$$\rho \frac{d}{dt}\boldsymbol{U} + \nabla P = 0, \tag{2b}$$

$$\rho \frac{d}{dt}E + \nabla \cdot (P\boldsymbol{U}) = 0, \tag{2c}$$

where $\frac{d}{dt}$ is the material time derivative. Here, $\rho, P, \boldsymbol{U}$ and E denote the density, pressure, velocity and specific total energy of the fluid. The previous equations express the conservation of volume, momentum and total energy in a frame which moves with the fluid. The thermodynamic closure of this set of equations is obtained by the addition of an equation of state which is taken to be of the form

$$P = P(\rho, \varepsilon),$$

where the specific internal energy, ε, is related to the specific total energy by $\varepsilon = E - \frac{1}{2}\|\boldsymbol{U}\|^2$. We note that for smooth flows, by subtracting kinetic energy equation from total energy equation, we get the time rate of change of specific internal energy

$$\rho \frac{d}{dt}\varepsilon + P\nabla \cdot \boldsymbol{U} = 0, \tag{3}$$

using (2a), this equation rewrites

$$\rho \frac{d}{dt}\varepsilon + P\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) = 0. \tag{4}$$

Recalling the Gibbs relation, $T\,dS = d\varepsilon + Pd(\frac{1}{\rho})$, where T denotes the temperature and S the specific entropy, it turns out that the previous internal energy Eq. (4) is equivalent to the conservation of entropy. We emphasize that this conclusion is valid only for smooth flows. The case of non-smooth flows, such as shock waves, is taken into account requiring that the second law of thermodynamics must be satisfied. To this end, we write the internal energy equation in the following non-conservative form:

$$\rho \frac{d}{dt}\varepsilon + P\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) = \rho T \frac{d}{dt}S \geqslant 0. \tag{5}$$

This thermodynamic framework will be used in what follows to derive the staggered discretization.

Let us remark that Eq. (2a) is also named the Geometric Conservation Law (GCL) and is strongly linked to the motion of the fluid which is ruled by the trajectory equation

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{U}(\boldsymbol{X}(t), t), \quad \boldsymbol{X}(0) = \boldsymbol{x}. \tag{6}$$

Here, $\boldsymbol{X}(t)$, denotes the position vector of a fluid particle at time $t > 0$, whose initial location was $\boldsymbol{x}$.

Here, we are deriving a discretization that is compatible with the GCL. By GCL compatibility, we mean that we are deriving a discrete divergence operator for the volume Eq. (2a) by requiring consistency of the divergence of the velocity field with the time rate of change of volume of the cell (cf. [92]). To highlight this particularly important point, it is instructive to re-write Eq. (2a) in finite-volume form. To this end, let us consider a finite Lagrangian volume, $V_c(t)$, characterized by a constant mass, m_c, and a mass density defined as $\rho_c(t) = \frac{m_c}{V_c(t)}$. With this notation, one might substitute the following for Eq. (2a):

$$\frac{d}{dt}V_c - V_c \int_{\partial V_c(t)} \boldsymbol{U} \cdot \boldsymbol{N}\, dS = 0, \tag{7a}$$

$$\frac{d}{dt}m_c = 0, \tag{7b}$$

$$\rho_c(t) = \frac{m_c}{V_c(t)}, \tag{7c}$$

Eq. (7a) states that the time rate of change of the cell volume must be equal to the volume swept by the element boundary during its displacement with the flow velocity. One has to discretize it with great care so that the discrete GCL over a cell remains compatible with the discrete motion of the cell vertices. Namely, the time rate of change of a Lagrangian volume has to be computed coherently with the node motion. It means that change of volume computed from Eq. (7a) has to be the same as one corresponding to computing volume at new time step from geometry of the cell, that is, in discrete case from positions of vertices of the cell, which are advance in time according to trajectory Eq. (6).

Another way of expressing CGL is to require that discrete divergence operator **DIV** is defined consistently with the following formula

$$\mathbf{div}\,\mathbf{U} = \frac{1}{V}\frac{dV}{dt} \tag{8}$$

where V is volume of fluid parcel. In discrete case fluid parcel is represented by cell V_c and its volume depend on time not directly but by means of coordinates of vertices dependence on time. That is,

$$(\mathbf{DIV}\,\mathbf{U})_c = \frac{1}{V_c}\frac{dV_c}{dt} = \frac{1}{V_c}\sum_p \left(\frac{dV_c}{dx_p}\frac{dx_p}{dt} + \frac{dV_c}{dy_p}\frac{dy_p}{dt}\right) = \frac{1}{V_c}\sum_p \left(\frac{dV_c}{dx_p}u_p + \frac{dV_c}{dy_p}v_p\right), \tag{9}$$

where sum is taken over all vertices p of cell c, and x_p, y_p, u_p, v_p, coordinates and velocities of vertices correspondingly. One can find details in (cf. [92]). The essence of GCL is do discretize Eq. (7a) consistently with Eq. (9). In application to cell-centered discretization process of deriving GCL compatible discretization is described in detail in [88]. This is the cornerstone of any Lagrangian discretization.

6.2. Compatible staggered scheme

The staggered discretization used in this paper is based on a staggered placement of the variables. Namely, the kinematic variables, including the velocity, are located at the nodes while the thermodynamic variables (density, pressure and specific internal energy) are defined at the cell center. We note that this placement of the variables allows the staggered scheme to fulfill naturally the GCL compatibility requirement and at the same time to construct a discrete divergence operator.

The discretizations of momentum and specific internal energy are derived from each other by use of important concept of compatible discretization [28], which is based on detailed balance between kinetic and internal energy and uses subzonal masses and subzonal forces. This compatible hydrodynamics algorithm is thus designed to conserve momentum and total energy exactly in discrete form. The dissipation of kinetic energy into internal energy through shock waves is ensured by means of an artificial viscosity which can be edge based [33] or tensorial [28]. This mechanism leads to a dissipation that is coherent with the second law of thermodynamics. The subzonal pressure method is also used for the control of hourglass-type motion [32]. Finally, the time integration method is a predictor–corrector technique which is detailed in [30]. The extension of this compatible Lagrangian hydrodynamics algorithm to unstructured grids, where each zone is a polygon with an arbitrary number of sides, has been presented in [29].

6.3. Cell-centered scheme

This discretization employs a centered placement of the variables. That is density, pressure, momentum and total energy are piecewise constant over each cell. The interface fluxes and the nodal velocity are computed by means of a node-centered approximate Riemann solver. The resulting nodal velocity allows to calculate zone volumes in a consistent manner with their geometric definition. In this way, the GCL compatibility requirement is ensured [40,88]. The main new feature of the algorithm used here, is the introduction of four pressures on each edge, two for each node on each side of the edge [88]. This extra degree of freedom allows to construct a nodal solver which fulfills two properties. First, momentum and total energy are rigorously conserved at the discrete level. Second, a semi-discrete entropy inequality is provided, which shows that kinetic energy is correctly dissipated into internal energy through shock wave. The node based feature of this scheme makes it naturally unstructured and thus able to deal with polygonal meshes. The high-order extension is derived using a one-step time integrator, based on the Generalized Riemann Problem (GRP) methodology [87]. It consists in solving the high-order Riemann problem with piecewise linear polynomial, whereby the approximate solution is given as a time power series expansion right at the interface. The acoustic version of the GRP method has been implemented and extended to the framework of the two-dimensional node-centered Riemann solver. In this way, we get an acoustic node-centered Generalized Riemann solver which provides the time derivatives of the nodal velocity and pressures, needed for the high-order flux computation.

6.4. Time step control

The Lagrangian discretizations require a time step control to ensure the stability of the schemes. Let Δt^n denotes the current time step, the next time step, Δt^{n+1}, for both discretizations is evaluated using several criteria. The first one is a standard CFL criterion based on the characteristic time

$$\Delta t_{\mathrm{cfl}} = C_{\mathrm{cfl}} \min_c \frac{l_c^n}{a_c^{\star,n}},$$

where C_{cfl} is a strictly positive coefficient, l_c^n a characteristic zone length defined as the minimum of edge lengths in the cell. The generalized sound speed, $a_c^{\star,n}$, writes $a_c^{\star,n} = \sqrt{\frac{\gamma_c P_c^{\star,n}}{\rho_c^n}}$, where γ_c is the ratio of specific heats, ρ_c^n the mean zone density. The generalized pressure, $P_c^{\star,n}$, writes $P_c^{\star,n} = P_c^n$ for the cell-centered scheme and $P_c^{\star,n} = P_c^n + Q_c^n$ for the staggered scheme, where P_c^n denotes the mean zone pressure and Q_c^n the scalar part of the artificial viscosity tensors in a zone.

The second criterion ensures that a zone does not change its volume by too large an amount in a time step. To this end, we define the characteristic time

$$\Delta t_{\mathrm{vol}} = C_{\mathrm{vol}} \min_c \frac{1}{|(\nabla \cdot \boldsymbol{U})_c^n|},$$

where C_{vol} is a user defined coefficient and $(\nabla \cdot \boldsymbol{U})_c^n$ stands for the discrete divergence operator related to cell c.

Finally, the new value of the time step reads

$$\Delta t^{n+1} = \min\left(\Delta t_{\mathrm{cfl}}, \Delta t_{\mathrm{vol}}, C_{\mathrm{mul}}\Delta t^n\right).$$

Here, C_{mul} is a multiplicative coefficient which does not allow time step to increase too fast. For numerical applications, we set $(C_{\mathrm{cfl}}, C_{\mathrm{vol}}, C_{\mathrm{mul}}) = (0.25, 0.1, 1.05)$. Note that for the staggered discretization the new value of the time step is always chosen on the predictor step.

When we have to deal with polygonal cells containing small edges we need to supplement the previous time step control with a criterion that prevents the cells from being non-convex or tangling during the Lagrangian phase. To this end, let us consider a polygonal cell Ω_c and one of its vertex indexed by p. We also consider the previous and the next point of p in the counter-clockwise ordered list of points of Ω_c. We label them respectively p^- and p^+. The triangle formed with these points is denoted T_{pc}, its area writes

$$|T_{pc}| = \frac{1}{2}(\boldsymbol{X}_p\boldsymbol{X}_{p^+} \times \boldsymbol{X}_p\boldsymbol{X}_{p^-}) \cdot \boldsymbol{e}_z,$$

where $\boldsymbol{e}_z$ supplements the orthonormal basis $(\boldsymbol{e}_x, \boldsymbol{e}_y)$, i.e. $\boldsymbol{e}_z = \boldsymbol{e}_x \times \boldsymbol{e}_y$. It is well known that the cell Ω_c is convex provided that the area of T_{pc} is strictly positive for each point p of the cell. We will use this sufficient condition to predict an admissible time step. During the Lagrangian phase, the position vector of the points is updated according to

$$\boldsymbol{X}_p^{n+1} = \boldsymbol{X}_p^n + \Delta t \boldsymbol{U}_p^{n+\frac{1}{2}},$$

where $\boldsymbol{U}_p^{n+\frac{1}{2}}$ is the time-centered point velocity and Δt is the current time step. It turns out that the area of T_{pc}^{n+1} is a quadratic function of Δt. Thus, for each triangle T_{pc} we compute the strictly positive time step so that $T_{pc}^{n+1} > 0$. Next, we compute its minimum over the cell and the global minimum over the whole polygonal mesh to get the characteristic time Δt_{pol}. Finally, we modify the time step control as follows to take into account this new criterion

$$\Delta t^{n+1} = \min\left(\Delta t_{\mathrm{cfl}}, \Delta t_{\mathrm{vol}}, C_{\mathrm{pol}}\Delta t_{\mathrm{pol}}, C_{\mathrm{mul}}\Delta t^n\right),$$

where the safety coefficient is set to $C_{\mathrm{pol}} = 0.25$.

6.5. Multi-species thermodynamic closure

In this paragraph, we describe the multi-species thermodynamic closure model that we are using to obtain an effective equation of state for our multi-component fluid mixture. Every components are completely miscible from a continuum view point. Let us denote by the subscript f the fth component of the mixture. We suppose that each fluid follows a gamma gas law, namely its pressure, P_f, and specific internal energy, ε_f, write as function of temperature T_f

$$P_f = \frac{R}{\mathcal{M}_f}\rho_f T_f,$$

$$\varepsilon_f = \frac{R}{(\gamma_f - 1)\mathcal{M}_f}T_f,$$

where R denotes the perfect gas constant, γ_f the polytropic index of fluid f and $\mathcal{M}_f$ its molar mass. Each fluid is characterized by its mass fraction C_f, which represents the ratio between the mass of the fluid f and the total mass of the mixture. The mixture EOS closure problem requires to find the equilibrium mixture pressure, P, and temperature, T, such that

$$\frac{1}{\rho} = \sum_{f=1}^{F} \frac{C_f}{\rho_f}, \quad \text{volume conservation},$$

$$\varepsilon = \sum_{f=1}^{F} C_f \varepsilon_f, \quad \text{energy conservation},$$

$$P_f = P, \quad \forall f = 1 \cdots F, \quad \text{pressure equilibrium},$$

$$T_f = T, \quad \forall f = 1 \cdots F, \quad \text{temperature equilibrium},$$

where F denotes the total number of fluids, ρ, ε are the density and the specific internal energy of the mixture and ρ_f, ε_f the density and specific internal energy of fluid f. The solution of the previous set of equations allows to write the following effective mixture gamma gas law:

$$P = (\gamma - 1)\rho\varepsilon,$$

where γ is the effective polytropic index of the mixture, which writes

$$\gamma = 1 + \frac{\sum_{f=1}^{F} \frac{C_f}{M_f}}{\sum_{f=1}^{F} \frac{C_f}{(\gamma_f - 1)M_f}}. \tag{10}$$

During the Lagrangian phase, the concentration of each fluid evolves following the trivial equation $\frac{d}{dt} C_f = 0$.

7. Rezone phase

7.1. Generators displacement

Let Ω_c^n and Ω_c^{n+1} denotes the Lagrangian cells at time t^n and $t^{n+1} = t^n + \Delta t$ where Δt is the current time step. The position vector of the generator of the Lagrangian cell Ω_c^n is denoted $\boldsymbol{G}_c^n$. In this section we define the new position of the generator at time t^{n+1}. First, we compute a Lagrangian-like displacement of the generator by setting

$$\boldsymbol{G}_c^{n+1,\mathrm{lag}} = \boldsymbol{G}_c^n + \Delta t \boldsymbol{U}_c, \tag{11}$$

where $\boldsymbol{U}_c$ is the "Lagrangian" velocity of the generator within the cell. This velocity is computed so that the generator remains located in the new Lagrangian cell. To this end we define this velocity to be the average of the velocities of the points of the cell, namely

$$\boldsymbol{U}_c = \frac{1}{|\mathcal{P}(c)|} \sum_{p \in \mathcal{P}(c)} \boldsymbol{U}_p^{n+\frac{1}{2}}.$$

Here, $\mathcal{P}(c)$ denotes the set of vertices of the Lagrangian cell Ω_c and $\boldsymbol{U}_p^{n+\frac{1}{2}}$ is the time-centered velocity of point p between times t^n and t^{n+1}.

Let us introduce, $\boldsymbol{X}_c^{n+1}$, the centroid of the Lagrangian cell Ω_c^{n+1}, according to

$$\boldsymbol{X}_c^{n+1} = \frac{1}{|\Omega_c^{n+1}|} \int_{\Omega_c^{n+1}} \boldsymbol{X}\, dV,$$

where $|\Omega_c^{n+1}|$ denotes the volume of the cell Ω_c^{n+1}. The updated position of the generator is defined by mean of a convex combination between the new Lagrangian-like position, $\boldsymbol{G}_c^{n+1,\mathrm{lag}}$ and the centroid $\boldsymbol{X}_c^{n+1}$ of the Lagrangian cell at time t^{n+1}

$$\boldsymbol{G}_c^{n+1} = \boldsymbol{G}_c^{n+1,\mathrm{lag}} + \omega_c \left(\boldsymbol{X}_c^{n+1} - \boldsymbol{G}_c^{n+1,\mathrm{lag}} \right), \tag{12}$$

where $\omega_c \in [0;1]$ is a parameter that remains to determine. With this convex combination, the updated generator lies in between its Lagrangian position at time t^{n+1} and the centroid of the Lagrangian cell Ω_c^{n+1}. We note that for $\omega_c = 0$ we get a Lagrangian-like motion of the generator whereas for $\omega_c = 1$ we obtain a centroidal-like motion, which tends to produce a smoothed mesh. This latter case is equivalent to perform one Lloyd iteration [43,42]. It remains to determine ω_c.

7.2. Computation of the ω parameter

The first role of the ω parameter is to construct a convex combination between the Lagrangian grid and the centroidal Voronoi grid, that is why we want it to be in $[0,1]$. Moreover, we want it to vanish smoothly for rigid rotation and translation and recover the pure Lagrangian motion in these cases. In a nutshell, we want it to be Galilean invariant. To construct a parameter that fulfills the previous requirements we utilize mechanical objects that characterize properly the mechanical features of the flow.

A good candidate for this task is the right Cauchy–Green tensor, [22], evaluated between two consecutive Lagrangian time steps. It can be regarded as quantifying the ratio of squared lengths of infinitesimal fibers between the initial and the initial configurations. To be more precise, let $d\boldsymbol{X}_1 = L_1 \boldsymbol{N}_1$ denotes a material fiber of length L_1 in the initial configuration, where $\boldsymbol{N}_1$ is a unit vector. This elementary fiber is stretched and rotated to $d\boldsymbol{X}_2 = L_2 \boldsymbol{N}_2$, in the final configuration, where $\boldsymbol{N}_2$ is a unit vector. Introducing F as the deformation gradient between the initial and the final configuration, one has $d\boldsymbol{X}_2 = \mathsf{F} d\boldsymbol{X}_1$. Therefore, one can deduce that the length of the fiber in the final configuration is given by

$$L_2^2 = d\boldsymbol{X}_2 \cdot d\boldsymbol{X}_2 = L_1 \mathsf{F} \boldsymbol{N}_1 \cdot L_1 \mathsf{F} \boldsymbol{N}_1 = L_1^2 \mathsf{F}^t \mathsf{F} \boldsymbol{N}_1 \cdot \boldsymbol{N}_1. \tag{13}$$

Hence, the ration of the squared lengths writes as

$$\left(\frac{L_2}{L_1} \right)^2 = \mathsf{C} \boldsymbol{N}_1 \cdot \boldsymbol{N}_1,$$

where $\mathsf{C} = \mathsf{F}^t \mathsf{F}$ is the Cauchy–Green tensor, which is symmetric definite positive and fulfills the requirements of Galilean invariance. Here, the subscripts 1 and 2 refer to the initial and final configurations corresponding to the beginning and the end of the Lagrangian time step.

Returning to our standard notations, we construct ω_c using invariants of the right Cauchy–Green strain tensor associated with deformation of the Lagrangian cell Ω_c between times t^n and t^{n+1}.

Let us recall some general notions of continuum mechanics to define this tensor. First, we define the deformation gradient tensor F

$$\mathsf{F} = \frac{\partial \boldsymbol{X}^{n+1}}{\partial \boldsymbol{X}^n},$$

where $\boldsymbol{X}^{n+1} = (X^{n+1}, Y^{n+1})^t$ denotes the vector position of a point at time t^{n+1} that was located at position $\boldsymbol{X}^n = (X^n, Y^n)^t$ at time t^n. The deformation gradient tensor is nothing but the Jacobian matrix of the map that connects the Lagrangian configurations of the flow at time t^n and t^{n+1}, in the two-dimensional case its components write

$$\mathsf{F} = \begin{pmatrix} \frac{\partial X^{n+1}}{\partial X^n} & \frac{\partial X^{n+1}}{\partial Y^n} \\ \frac{\partial Y^{n+1}}{\partial X^n} & \frac{\partial Y^{n+1}}{\partial Y^n} \end{pmatrix}.$$

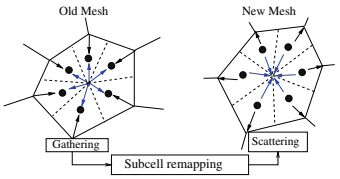


Fig. 7. Subcell remapping is performed in three phases – First, variables are gathered on subcells (left). Second a subcell-based conservative remapping is performed. Third, a scattering stage redistributes subcell-based conservative variables on nodes and cell centers (right).

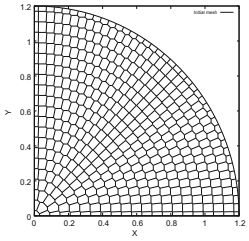


Fig. 8. Initial mesh for Sedov problem.

The right Cauchy–Green strain tensor, C, is obtained by right-multiplying F by its transpose, i.e.

$$\mathsf{C} = \mathsf{F}^t \mathsf{F}.$$

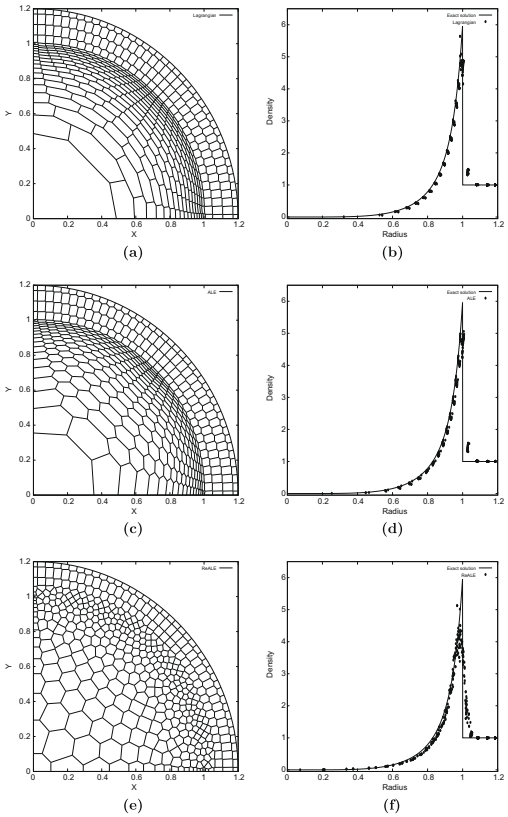


Fig. 9. Sedov problem at time $t = 1.0$ for Lagrangian, ALE and ReALE strategies – Cell-centered CHIC code – Left column: Mesh. Right column: Density as a function of radius for all cells *vs.* the exact solution (line) – (a) and (b) Lagrangian – (c) and (d) ALE – (e) and (f) ReALE.

In our case, C is a 2×2 symmetric positive definite tensor. We notice that this tensor reduces to the unitary tensor in case of uniform translation or rotation. It admits two positive eigenvalues, which are denoted λ_1 and λ_2 with the convention $\lambda_1 \leqslant \lambda_2$. These eigenvalues can be viewed as the rates of expansion in a given direction during the transformation. To determine ω_c, we first construct the cell-averaged value of the deformation gradient tensor, F_c, and then the cell-aver-

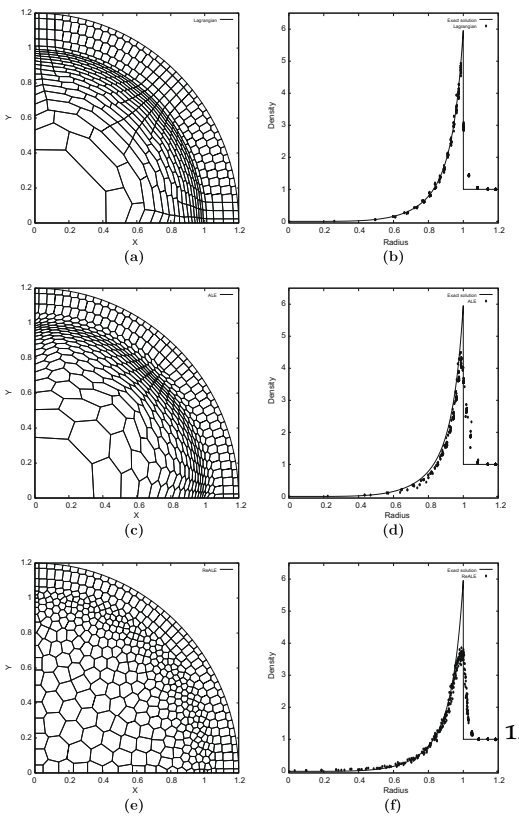


Fig. 10. Sedov problem at time $t = 1.0$ for Lagrangian, ALE and ReALE strategies – Staggered ALE INC. code – Left column: Mesh. Right column: Density as a function of radius for all cells *vs.* the exact solution (line) – (a) and (b) Lagrangian – (c) and (d) ALE – (e) and (f) ReALE.

aged value of the Cauchy–Green tensor by setting $C_c = F_c^t F_c$. Noticing that the two rows of the F matrix correspond to the gradient vectors of the X and Y coordinates, we can set $F^t = [\nabla_n X^{n+1}, \nabla_n Y^{n+1}]$, where for any functions

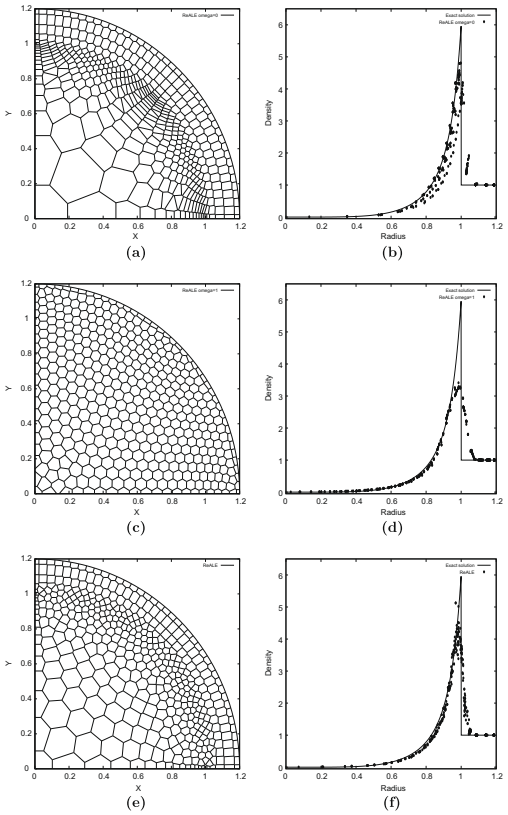


Fig. 11. Sedov problem at time $t = 1.0$ for different generator displacement strategies – Cell-centered CHIC code – Left column: Mesh. Right column: Density as a function of radius for all cells *vs.* the exact solution (line) – (a) and (b) Quasi-Lagrangian generator motion $\omega_c = 0$ – (c) and (d) Quasi-centroidal generator motion $\omega_c = 1$ – (e) and (f) Deformation-tensor based generator motion $\omega_c = f(\lambda_1, \lambda_2)$.

$\psi = \psi(X^n)$, $\nabla_n \psi = (\frac{\partial \psi}{\partial X^n}, \frac{\partial \psi}{\partial Y^n})^t$. With these notations, let us define the cell-averaged value of the gradient of the ψ function over the Lagrangian cell Ω_c^n

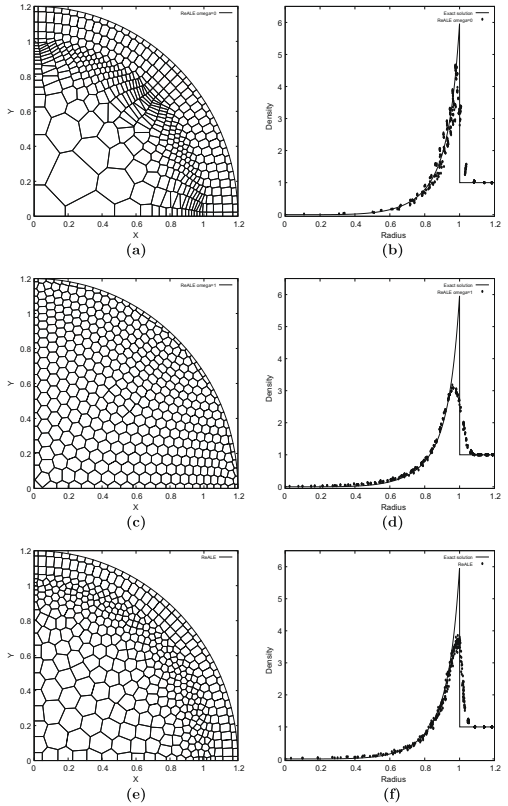


Fig. 12. Sedov problem at time $t = 1.0$ for different generator displacement strategies – Staggered ALE INC. code – Left column: Mesh. Right column: Density as a function of radius for all cells *vs.* the exact solution (line) – (a) and (b) Quasi-Lagrangian generator motion $\omega_c = 0$ – (c) and (d) Quasi-centroidal generator motion $\omega_c = 1$ – (e) and (f) Deformation-tensor based generator motion $\omega_c = f(\lambda_1, \lambda_2)$.

$$(\nabla_n \psi)_c = \frac{1}{|\Omega_c^n|} \int_{\Omega_c^n} \nabla_n \psi \, dV = \frac{1}{|\Omega_c^n|} \int_{\partial \Omega_c^n} \psi \mathbf{N} \, dS.$$

Here, we have used the Green formula and $\mathbf{N}$ is the unit outward normal to the boundary of the cell Ω_c^n referred as to $\partial \Omega_c^n$. Knowing that this cell is a polygon, we make use of the trapezoidal rule to obtain the following approximation for the previous integral

$$(\nabla_n \psi)_c = \frac{1}{|\Omega_c^n|} \sum_{p=1}^{|\mathcal{P}(c)|} \frac{1}{2} \left(\psi_p^n + \psi_{p+1}^n \right) L_{p,p+1}^n \mathbf{N}_{p,p+1}^n, \tag{14}$$

where ψ_p^n is the value of ψ evaluated at point X_p^n and $L_{p,p+1}^n \mathbf{N}_{p,p+1}^n$ is the outward normal to the edge $[X_p^n, X_p^{n+1}]$. Applying (14) to $\psi = X^{n+1}$ and $\psi = Y^{n+1}$ we get a cell-averaged expression of the gradient tensor F and then deduce from it the cell-averaged value of the right Cauchy–Green tensor C_c.

Knowing this symmetric positive definite tensor in each cell, we compute its real positive eigenvalues $\lambda_{1,c}$, $\lambda_{2,c}$. We finally define the parameter ω_c as follows:

$$\omega_c = \frac{1 - \alpha_c}{1 - \alpha_{\min}}, \tag{15}$$

where $\alpha_c = \frac{\lambda_{1,c}}{\lambda_{2,c}}$ and $\alpha_{\min} = \min_c \alpha_c$. We emphasize the fact that for uniform translation or rotation $\lambda_{1,c} = \lambda_{2,c} = 1$ and $\omega_c = 0$, therefore the motion of the generator is quasi Lagrangian and we fulfill the material frame indifference requirement. For other cases, ω_c smoothly varies between 0 and 1. Note that more complex formulae for ω_c are also possible, however we limit ourselves to the previous simple formula and postpone deeper investigations in this area for future papers.

In Section 9 we demonstrate sensitivity of choice of parameter ω_c on geometrical quality of the mesh and accuracy of ReALE calculations on the example of Sedov problem. We also present color map for values of the ω_c for 2D Riemann problem, which demonstrate its ability to follow main features of the flow. These results justify the choice of ω_c described by Eq. (15).

7.3. Cleaning

Once the new position of generators G_c^{n+1} are computed one constructs the corresponding Voronoi mesh. This mesh needs a last treatment as this Voronoi mesh may have arbitrary small faces (edges). Such faces can drastically and artificially reduce the time step, and, more important can lead to a lack of robustness. Consequently one defines a cutoff length $L_c^c = \varepsilon L_c$, where L_c is a characteristic length of the cell and ε a small parameter. Any face f of cell c of length l_f smaller than the cutoff length L_c^c, is removed from the Voronoi mesh , which lead to corresponding change in the connectivity. More specifically the vertices of such a face are merged (see Fig. 5 in Section 4); one vertex is then discarded from the vertex list and the connectivity structure. In our calculations we have chosen $\varepsilon = 0.01$ and $L_c = \frac{\sum_{f \in \mathcal{F}(c)} l_f}{|\mathcal{F}(c)|}$ being the average of face lengths of cell c with $\mathcal{F}(c)$ is the set of edges of cell c.

This "cleaned" polygonal mesh is no more of Voronoi kind but is well suited from a computational point of view.

In current code results of cleaning procedure depend on order in which cells are processed, that is, cleaning is order dependent. For unstable flows like Rayleigh–Taylor problem, it can lead to symmetry breaking, as one can see in Section 9.4.2.

One clearly can make cleaning procedure more symmetric and order "independent". For example, we can use Jacobi like cleaning procedure, when on the first pass one marks edges to be eliminated based on the minimal cut off length from two neighboring cells and then, on second pass, eliminates them all at the same time and makes corresponding changes in data structures.

One need to recognize that there is always be potential to developing non-symmetry in unstable flows like Rayleigh–Taylor example because of round off error. It can happen even without any cleaning.

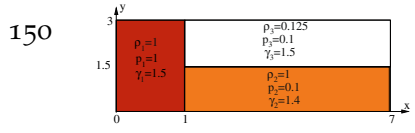


Fig. 13. Triple point problem.

8. Remap phase

The remapping phase consists of a conservative interpolation of physical variables from the Lagrangian polygonal mesh at the end of the Lagrangian step onto the new polygonal mesh after the rezone step. The remapping phase must provide valid physical variables to the Lagrangian scheme, moreover conservation of mass, momentum and total energy must be ensured, and, second-order accuracy conservative interpolation must be performed.

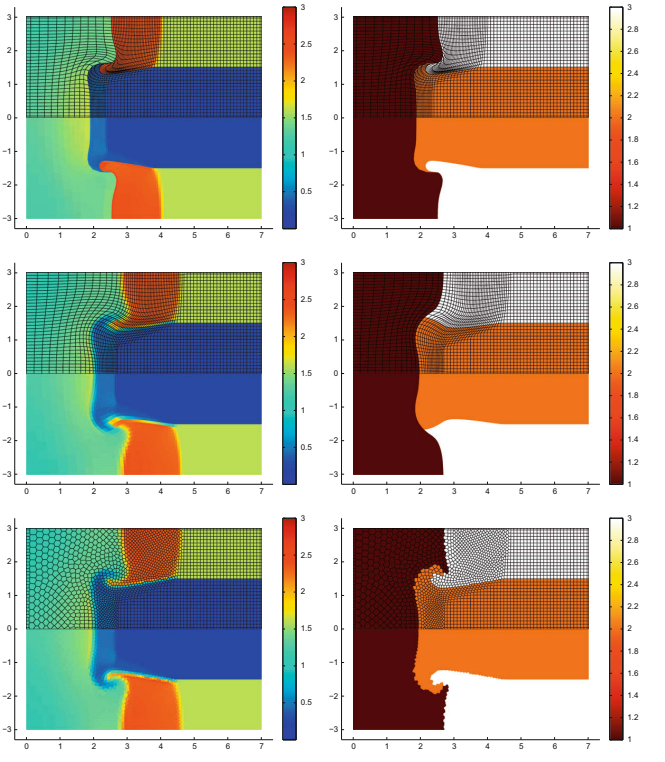


Fig. 14. Triple point problem at time t_{end} for which Lagrangian scheme fails – Cell-centered CHIC code – Left column: Internal energy and mesh. Right column: Cell color corresponds to the initial domain (Ω_1,Ω_2 and Ω_3) – Top: Lagrangian method; Middle: ALE method; Bottom: ReALE method. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

As the old (Lagrangian) and new (rezoned) polygonal meshes may not have the same connectivity, the remapping phase of our ReALE codes is based on exact intersection of *a priori* two different polygonal meshes.

Primary variables are cell-centered density, velocity and specific total energy for the cell-centered discretization whereas they are subcell density, nodal velocity and cell-centered specific internal energy for staggered discretization. Conservative quantities are cell-centered mass, momentum and total energy for the cell-centered approach whereas they are subcell mass, momentum and total energy for the staggered discretization.

8.1. Cell-centered based remap

If the primary variables are located at the same position, as it is the case for the cell-centered Lagrangian scheme, then the remapping phase is fairly simple [77,61,49]. The quantities on the old Lagrangian mesh are cell-centered density, velocity and total energy that must be transfered on the rezoned mesh. First piecewise linear representations of cell-centered variables ρ_c, $\rho_c\mathbf{U}_c$, $\rho_c E_c$ are constructed on the Lagrangian mesh. Then a slope limiting process [17] is performed to enforce physically justified bounds. Conservative quantities, namely mass, momentum and total energy, are obtained by integration of these representations. New conservative quantities are calculated by integration over polygons of intersection of new (rezoned) and old (Lagrangian) meshes. Finally, primary variables are simply recovered by division by new volume $\tilde{V}_c$ (for density) or new mass $\tilde{m}_c$ (for momentum and energy).

8.2. Subcell-centered based remap

Some difficulties arise when staggered location of variables is used, as for any staggered Lagrangian scheme. Conservative quantities are therefore not located on the same "entity"; mass is located at subcells, momentum at points. As a consequence, total energy is not properly defined at a given location. In [84] a gathering-remapping-scattering algorithm has been developed (see Fig. 7 for a sketch); the main idea being to gather any conservative variable at subcell level, remap on subcell base similarly to the cell-centered remapper from the previous paragraph, and scatter back primary variables from subcells (density) to cell (internal energy) or node (velocity). More specifically this method consists in the following three stage algorithm:

- Gathering: Mass, momentum, internal and kinetic energies are defined on subcells from subcell density, nodal velocity and cell-centered specific internal energy in such a way that conservation is preserved.
- Subcell remapping: Conservative remapping from Lagrangian mesh subcells onto rezoned mesh subcells.
- Scattering: Conservative recovery of primary variables (subcell density, nodal velocity, cell-centered internal energy) on the rezoned mesh.

This remapping stage is followed by a repair technique to ensure physically justified bound preservation [77,112,85].

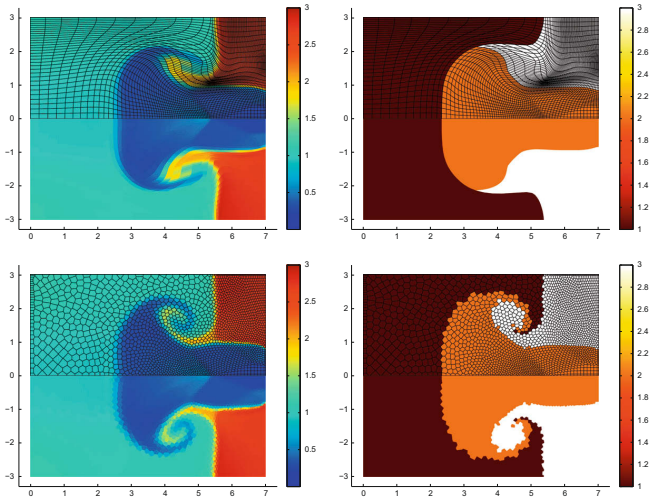


Fig. 15. Triple point problem at final time to observe stagnation of ALE mesh – Cell-centered CHIC code – Left column: Internal energy and mesh. Right column: Cell color corresponds to the initial domain (Ω_1,Ω_2 and Ω_3) – Top: ALE method; Bottom: Cell-centered (CHIC-based) ReALE method. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

8.3. Remapping of concentrations

To use the multi-species EOS, we need to remap the concentrations of the F fluids from the Lagrangian grid onto the rezoned one. To this end, we first compute the mass of fluid f in the Lagrangian cell Ω_c^{n+1}, $m_{f,c} = \int_{\Omega_c^{n+1}} \rho C_f dV$. We note that $m_c = \sum_{f=1}^{F} m_{f,c}$ since $\sum_{f=1}^{F} C_{f,c} = 1$. Then, the mass of each fluid is interpolated conservatively onto the rezoned grid following the methodology previously described for the cell-centered quantities. We denote its new value by $\tilde{m}_{f,c}$. At this point we notice that $\tilde{m}_c \neq \sum_{f=1}^{F} \tilde{m}_{f,c}$, this discrepancy comes from the fact that our second-order remapping does not preserve linearity due to the slope limiting. Hence, we define the new concentrations $\tilde{C}_{f,c} = \frac{\tilde{m}_{f,c}}{\tilde{m}_{f,c}}$ and impose the renormalization $\tilde{C}_{f,c} \leftarrow \frac{\tilde{C}_{f,c}}{\sum_{f=1}^{F} \tilde{C}_{f,c}}$ so that $\sum_{f=1}^{F} \tilde{C}_{f,c} = 1$. We point out that this renormalization does not affect the global mass conservation.

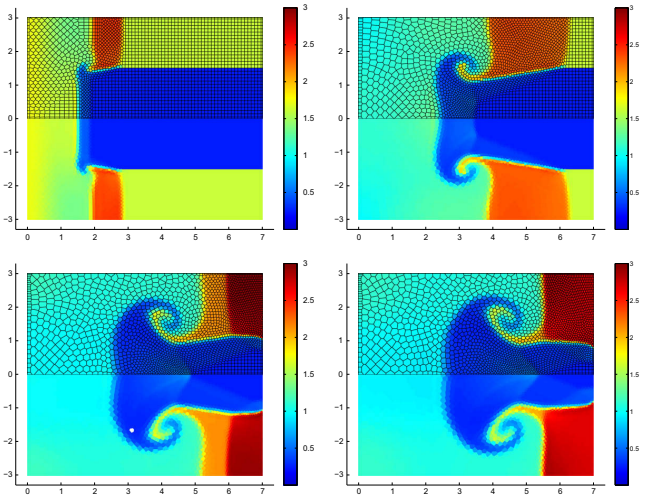


Fig. 16. Triple point problem at several times for ReALE – Cell-centered CHIC results for internal energy and mesh –From top-left to bottom-right: times $t = 1, 3, 4.5, 5$.

9. Numerical tests

In this section we present the numerical results obtained by the cell-centered ReALE code based on CHIC ALE code [87], and, the staggered ReALE code based on ALE INC(ubator) [82]. All calculations are performed in Cartesian geometry – (x,y).

The first test is the well-known Sedov test case in planar geometry, it is used as a sanity check as no physical vorticity is expected to occur and therefore reconnection-based methods are not required. The second is a "triple point" problem. It involves interaction of the shock with obstacle, which leads to vorticity formation. Most of the ReALE studies are performed using this problem. Third problem, shock interaction with a helium bubbler, is run in order to show the predictive capabilities of ReALE technique-we compare numerical results with experimental data. Finally, a Rayleigh–Taylor instability is run in order to assess the feasibility of capturing physical instability in an almost Lagrangian fashion. For this problem we compare our results with results obtained by the front tracking code (FronTier – [41,58,59]) and a an implicit large eddy simulation (ILES) incompressible Eulerian code (RTI3D – [4]).

All these tests, besides the Sedov test, generate high vorticity which is a classical cause of failure for Lagrangian schemes. For ALE codes with fixed connectivity it usually leads to a conflict between: a physics-based vortex-like motion with a tendency to tangle the mesh and, a geometrical-based opposite motion enforced by the rezoning to avoid bad geometric quality cells. Such a conflict leads to a stagnation of the mesh that reconnection is intended to cure.

9.1. Sedov problem

Let's consider the Sedov blast wave problem in Cartesian coordinates. This problem models an intense explosion in a perfect gas; it is an example of a diverging shock wave.

The computational domain is Ω, a quarter of a disk of radius 1.2 centered at the origin.

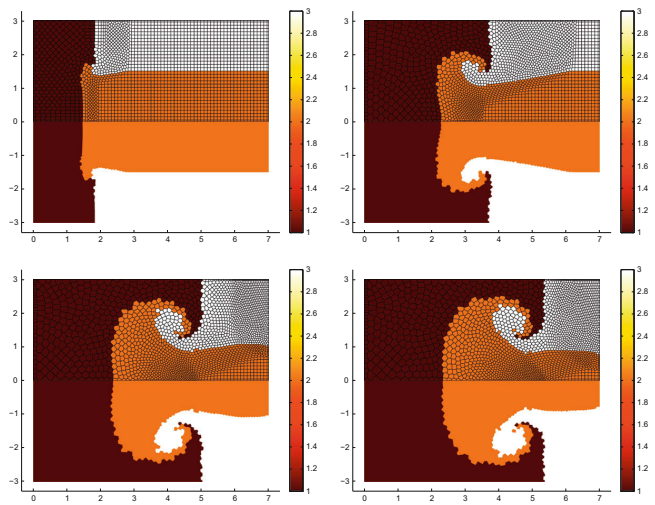


Fig. 17. Triple point problem at several times for ReALE – Cell-centered CHIC results – Cell color corresponds to the initial domain. From top-left to bottom-right: times $t = 1, 3, 4.5, 5$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The initial conditions are characterized by $(\rho_0, P_0, \mathbf{U}_0) = (1, 10^{-6}, 0)$ for a perfect gas with polytropic index set to $\gamma = \frac{7}{5}$. We model an initial delta-function energy source at the origin by prescribing internal energy in the cell adjacent to the origin (see Fig. 8)

$$\varepsilon_{or} = \frac{\mathcal{E}_0}{V_{or}}, \tag{16}$$

where V_{or} denotes the volume of the cell and $\mathcal{E}_0$ is the total amount of released energy. For $\mathcal{E}_0 = 0.244816$, the front of a diverging shock is located at the radius $R = 1$ at the time $t = 1$. The peak density reaches the value 6. Symmetry boundary conditions are applied on the axis whereas zero velocity boundary condition is applied at radius 1.2.

The initial polygonal mesh is a Voronoi tessellation computed using 441 generators, see Fig. 8. The positions of generators are arranged similarly to how it described in Section 4 for meshing bubble, Fig. 4.

To run this test we do not need ALE, and *a fortiori* ReALE, technique; pure Lagrangian schemes usually perform well. However, we will present the Lagrangian, ALE and ReALE results (both for the cell-centered and staggered code) for the sake of comparison. Different generator motions are also compared.

9.1.1. Lagrangian, ALE and ReALE results

We present the meshes in Fig. 9 (resp. Fig. 10) panels (a)–(c)–(e) for the cell-centered CHIC-based methods (resp. for the staggered ALE INC.-based methods). The density for these methods is presented as a function of the cell radius for all cells in Figs. 9 and 10 panels (b), (d), (f); it is plotted against an exact solution (straight line). As known, Lagrangian schemes behave properly for the Sedov test case as seen on Figs. 9 and 10 panel (a) and (b). The final Lagrangian mesh presents expanded cells in the rarefaction wave and compressed ones after the shock wave. ALE techniques (panels (c) and (d)) can improve the smoothness of the mesh; some numerical diffusion is added during the remapping phase as can be seen on panels (d) of Figs. 9 and 10. ReALE technique used with deformation-tensor based generator motion (see panels (e) and (f)) is able to produce a smooth mesh and density profiles comparable with standard ALE.

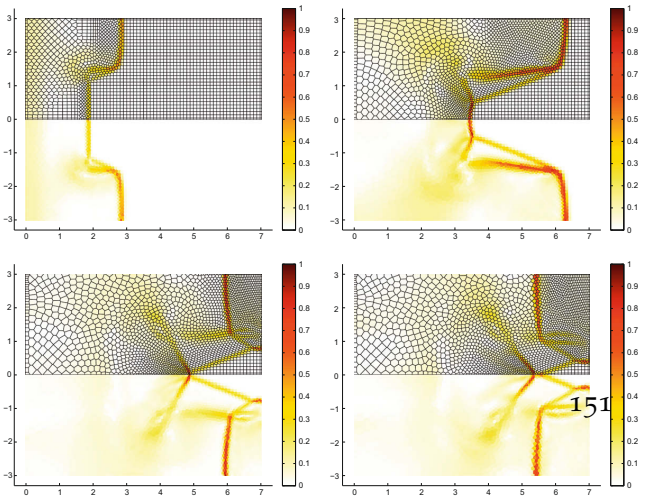


Fig. 18. Triple point problem at several times for ReALE – Cell-centered CHIC results for variable ω_c – From top-left to bottom-right: times $t = 1, 3, 4.5, 5$.

The CHIC-based methods give higher pick density value in comparison with ALE INC.-based methods because formally Lagrangian phase in CHIC method is second-order accurate and Lagrangian phase in ALE INC. method is first-order accurate. Small differences in ALE results (panels (c) and (d)) can be attributed to slightly different rezone strategy.

9.1.2. Generator displacement

The same problem is run with the cell-centered CHIC-based ReALE code and staggered ALE INC.-based ReALE code with three different generator the quasi-Lagrangian one ($\omega_c = 0$), the quasi-centroidal one ($\omega_c = 1$) and the motion based on the deformation tensor ($\omega_c = f(\lambda_1, \lambda_2)$).

The results for CHIC-based ReALE code presented in Fig. 11. As expected the quasi-Lagrangian generator motion leads to non-smooth mesh which, however, adapted quite well to the flow. One can mention that symmetry is not well preserved, which in particular can be attributed to non-smoothness of the mesh (Fig. 11(a) and (b)). On the other hand, the quasi-centroidal generator motion (Fig. 11(c) and (d)), leads to a smooth mesh that is not adapted anymore to the fluid flow; density is over-smoothed due to excessive remapping. Finally the motion based on the deformation tensor described in Section 7 (Fig. 11(e) and (f)), produces a locally smooth mesh and keeps finer cell region after the shock wave passes through, and coarser cell region after expansion.

In Fig. 12 we present the same results for the staggered ALE INC.-based ReALE, code and the same conclusions apply.

Results presented here justify rezone strategy based on the analysis of the deformation tensor described in Section 7 and the rest of the paper in all numerical ReALE simulations we use only this rezone strategy.

The Sedov test case is not extremely demanding neither for Lagrangian nor ALE scheme. Contrarily, the next test cases involve generation of the vorticity and are intended to demonstrate the capabilities of the new ReALE method.

9.2. Triple point problem – two material Riemann problem

This problem is a three state two material 2D Riemann problem in a rectangular vessel. The triple point problem simulation domain is $\Omega = [0;7] \times [0;3]$ as described in Fig. 13. Ω is split into three regions filled with two perfect gases leading to

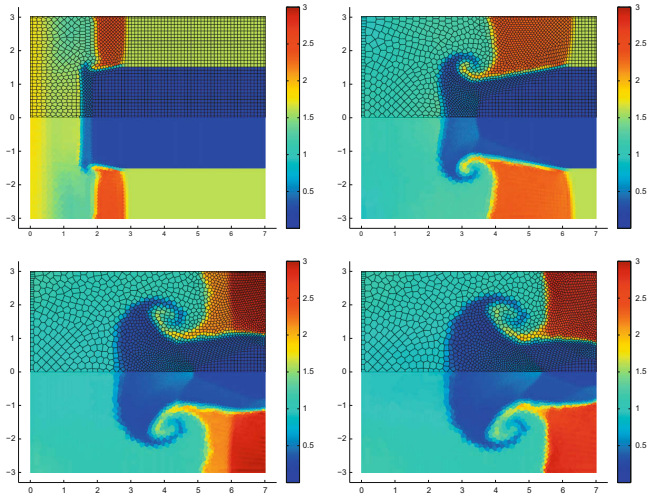


Fig. 19. Triple point problem at several times for ReALE – Staggered-ALE INC. results – internal energy and mesh. From top-left to bottom-right: times $t = 1$, 3, 4.5, 5.

a two material problem. The high pressure high density state is $\Omega_1 = [0;1] \times [0;3]$, the low pressure high density state is $\Omega_2 = [1;7] \times [0;1.5]$ and the low pressure low density is $\Omega_3 = [1;7] \times [1.5;3]$. The initial densities are $\rho_1 = \rho_3 = 1$, $\rho_2 = 0.125$, the initial pressures are $p_1 = 1$, $p_3 = p_2 = 0.1$, the initial velocity is zero everywhere. The perfect equation of state is used with $\gamma_1 = \gamma_3 = 1.5$, $\gamma_2 = 1.4$.

Due to the discrepancy in density, two shocks in domains Ω_2 and Ω_3 propagate with different speeds. This creates a shear along initial contact discontinuity and a vorticity formation. We note that the Lagrangian computation fails before vortex is developed due to the mesh tangling. Capturing the vorticity is the difficult part of such simulation when standard ALE method is used.

Initially 72×32 generators are positioned on a perfect quadrangular grid leading to 2304 degenerate Voronoi cells. The generators are located in such a way that, initially there is no mixed cells and the triple point coincides with a vertex of the mesh. This mesh is intentionally coarse, such that differences between the methods can be visually observed. The boundary conditions are reflective ones.

The fluid flow after the breakup of the initial discontinuity is characterized by a left facing rarefaction wave and two right facing shock waves separated by an "horizontal" contact discontinuity. These two shocks travels with different speeds since the densities of the materials are different. This leads to a strong vortex formation. The final time is $t = 5$.

9.2.1. Results obtained by cell-centered ReALE method

Lagrangian scheme failure. Time $t_{fail} \simeq 1.67$ corresponds, more or less, to the time after which any Lagrangian scheme inexorably fails. In Fig. 14 we present the Lagrangian (top panels), ALE (middle panels) and ReALE (bottom panels) results for the cell-centered CHIC code for this time moment. The internal energy and meshes are displayed in the left panel (top part is mesh and internal energy, bottom part is color map for internal energy). In the right panels of Fig. 14 we present the mesh for each method such that the cells have been colored according to in which domain (Ω_1 in dark-red, Ω_2 in orange, Ω_3 in white) they were initially located. The Lagrangian scheme produces results that follow the motion of the fluid

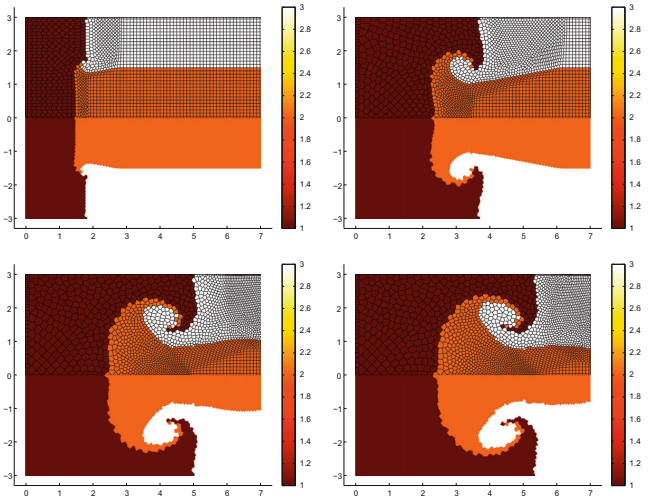


Fig. 20. Triple point problem at several times for ReALE – Staggered ALE INC. results – Cell color corresponds to the initial domain. From top-left to bottom-right: times $t = 1$, 3, 4.5, 5. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

but the scheme fails because of mesh tangling in vicinity of the vortex. The ALE method does not allow the mesh to follow the fluid motion and the mesh does not tangle. The ReALE seems to allow the cells to follow the vortex-shaped fluid motion, and up to this moment color map related to the initial location for Lagrangian and ReALE methods are very close, Fig. 14. This is indication that our rezone strategy is able to keep "centers" of the cells very close to its Lagrangian positions.

It is important to note that due to our choice of parameter ω_c, participating in movement of generators in ReALE method, meshes in front of shocks have not changed.

"Stagnation" of the mesh in ALE method. ALE and ReALE perform up to the final time $t = 5$. In Fig. 15 we present results for ALE and ReALE for final time-arrangements are the same as in Fig. 14. Results of the ALE calculation clearly show a mesh "stagnation" behavior as can be seen on the initial domain color map, whereas the ReALE allows the cells to be carried along with the vortex.

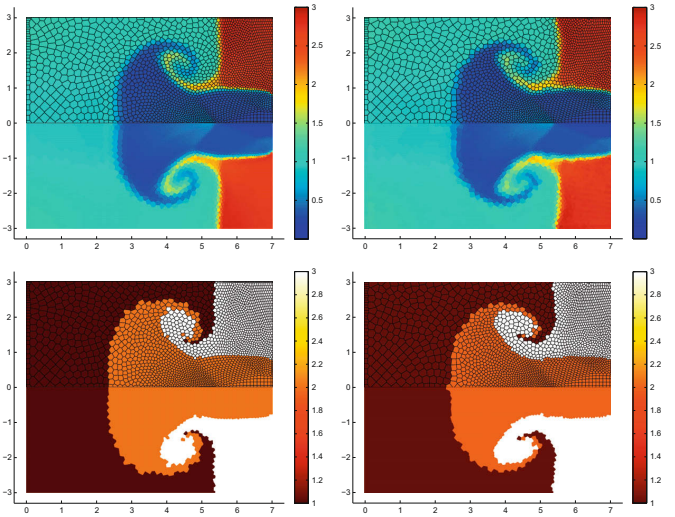


Fig. 21. Triple point problem. Comparison of cell-centered (left panels) and staggered (right panels) ReALE methods at the final time moment. Top panels – internal energy and mesh, bottom panels – coloring by initial region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

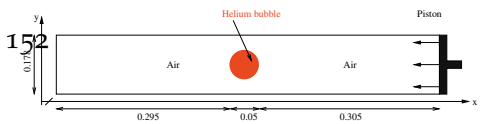


Fig. 22. Shock/Bubble interaction. A piston (right-boundary) moving to the left and compresses air initially at rest sending a shock wave that passes through an helium bubble.

In ALE calculation (top panels of Fig. 15), the mesh is "freezes" near the vortex center and the computation continues in an almost Eulerian fashion (as Lagrange + Remap because the rezone phase systematically backs up the Lagrangian t^{n+1} mesh onto the previous t^n Lagrangian mesh).

In ReALE calculation (bottom panels of Fig. 15), mesh is not Lagrangian, but it preserves Lagrangian character of the flow, because mesh follows the vortex as generators are carried within the fluid in an almost Lagrangian fashion (bottom-right panel in Fig. 15).[3] As a consequence ReALE has a better accuracy, which can be seen comparing internal energy color maps for ALE and ReALE. In ReALE results one can see roll up formation and in ALE results it is not that pronounced. Let as note that quantitative analysis of accuracy will be presented in the separate paper.

ReALE results. In Fig. 16 we present dynamics of the ReALE simulation by showing the mesh and the specific internal energy at different time moments $t = 1$ (beginning of vortex development), $t = 3$ (before the fastest shock reaches right wall), $t = 4.5$ (after shock reflection), $t = 5$ (after reflected shock reaches contact discontinuity). The meshes with initial domain coloring are presented in Fig. 17.

Features of the flow and ω_c factor. In Fig. 18 are presented the ω_c factors for different snapshots of the simulation. The ω_c factor shows the deformation encountered by the mesh between two last time steps. Comparison of these plots with structure of the main waves presented in Fig. 16 clearly demonstrate that ω_c dynamically detect main features of the flow. It is important to note that ω_c is very small in high vorticity regions (as it intended to be) and therefore movement of the generators in these areas is almost Lagrangian.

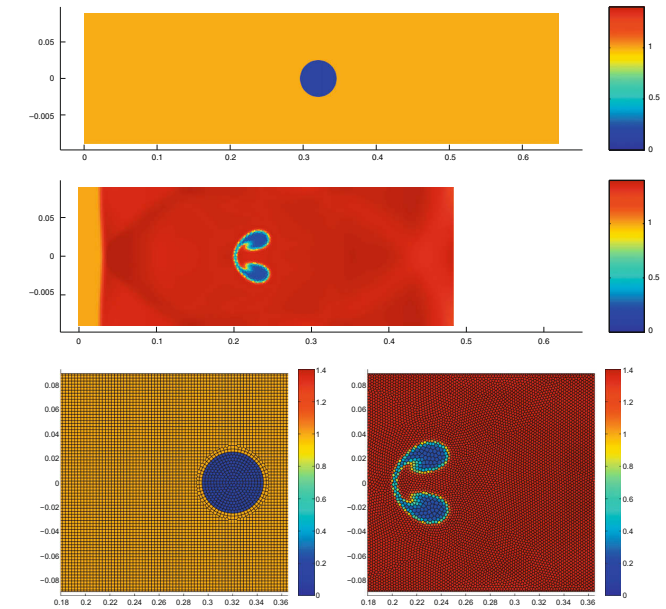


Fig. 23. Shock/Bubble interaction problem. Top: Initial density; Middle: Final density; Bottom: Zoom on the initial bubble and final bubble density and mesh.

9.2.2. Comparison of cell-centered and staggered ReALE methods

In this section we give brief comparison of ReALE calculations based on cell-centered CHIC method with results obtained by ReALE method based on staggered ALE INC. method.

In Fig. 19 we present internal energy and mesh plots for ALE INC.-based ReALE method for different time moments. It has to be compared with CHIC-based ReALE results presented in Fig. 16. In Fig. 20 we present coloring by initial region for staggered method.

Finally, we present side by side comparison of two methods in Fig. 21.

One can see only very small differences between cell-centered and staggered ReALE methods.

Results presented in this section and results for Sedov problem allows us to conclude that ReALE approach can be used for either cell-centered or staggered discretization basic Lagrangian scheme.

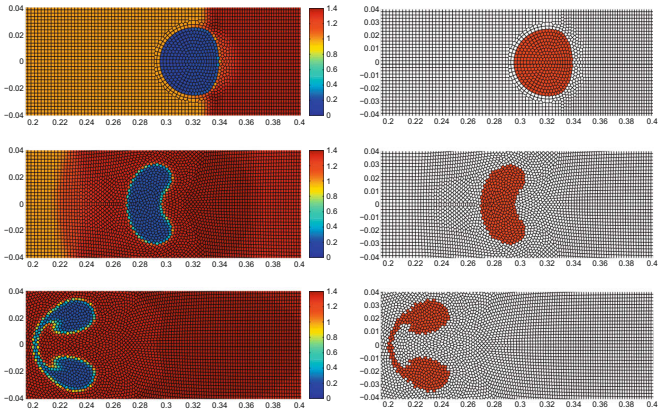


Fig. 24. Shock/Bubble interaction problem at $t = 1005.15 \times 10^{-6}$ (top panel), $t = 1101.44 \times 10^{-6}$ (middle panel) and $t = t_{end} = 1342.153 \times 10^{-6}$ (bottom panel) – Left column: Density and mesh. Right column: Coloring by the initial domains (white: air, red: bubble). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

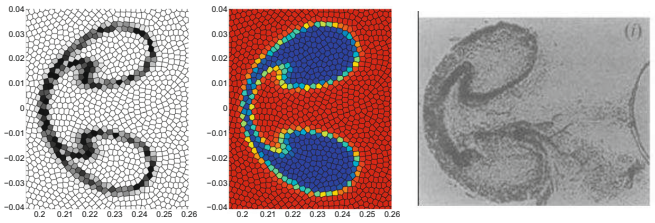


Fig. 25. Shock/Bubble interaction problem at $t_{end} = 1342.153 \times 10^{-6}$– Left: Zoom of color map for $I_c = C_{h,c}(1 - C_{h,c})$: grayscale goes from $I_c = 0$ (white) to $I_c = 1/4$ (black); Middle: Zoom of color map for density: colorscale goes from $\rho_c = 0.23$ (blue) to $\rho_c = 1.42$ (red); – Right: Schlieren image from experimental data in [62]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

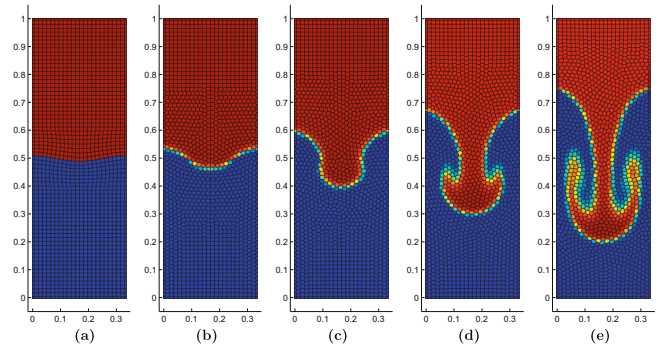


Fig. 26. Rayleigh–Taylor instability – ReALE results at times $t = 0, 3, 5, 7, 9$ (from (a) to (e)) – Mesh and density – Color scale is from blue ($\rho = 1$) to red ($\rho = 2$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Detailed comparison of cell-centered and staggered discretizations is beyond of the goal. Actually it will require full new paper. At this moment it will be premature to recommend one discretization over the other. Our goal in this paper is to demonstrate that ReALE methodology can be used for both types of discretizations. For the sake of space, in the reminder of the paper we will present only results for cell-centered discretization.

9.3. Bubble shock interaction

The computational domain is $\Omega = [0; 0.65] \times [0; 0.178]$. The bubble is a disk defined by the coordinates of its center $(x_c, y_c) = (0.320, 0)$ and its radius $R_b = 0.025$ (see Fig. 22). We prescribe reflective boundary conditions at each boundary except the right-boundary (initially at $x = 0.65$); right-boundary is the piston which moves inward with velocity $\boldsymbol{V}^\star = (u^\star, 0)$. The incident shock wave is defined by its Mach number, $M_s = 1.22$. The bubble and the air are initially at rest. The initial data for Helium bubble is $(\rho_1, P_1) = (0.182, 10^5)$, its molar mass is $\mathcal{M}_1 = 5.269 \times 10^{-3}$ and its polytropic index is $\gamma_1 = 1.648$. The initial data for air is $(\rho_2, P_2) = (1, 10^5)$, its molar mass is $\mathcal{M}_2 = 28.963 \times 10^{-3}$ and its polytropic index is $\gamma_2 = 1.4$. It is two material problem. Specific internal energies are $\varepsilon_1 = 8.4792 \times 10^5$ and $\varepsilon_2 = 2.5 \times 10^5$. Using the Rankine–Hugoniot relations, we find that the x-velocity of the piston is given by $u^\star = -124.824$. The x-component of the incident shock velocity is $D_c = -456.482$. The incident shock wave hits the bubble at time $t_i = 668.153 \times 10^{-6}$. The stopping time for our computation is $t_{end} = t_i + 674 \times 10^{-6} = 1342.153 \times 10^{-6}$. It corresponds to the time for which experimental Schlieren graph (results from [62]) is displayed in [103] (Fig. 9(i)). Initial mesh is constructed with a set of 19,061 generators and designed to produce a Voronoi mesh that has a mesh lines which matches the bubble boundary (see bottom left panel in Fig. 23, where a zoom of the initial mesh around the bubble is presented). The description of initial mesh generation given in Section 4. Initial density is presented in top panel in Fig. 23. The middle panel presents the final density which corresponds to the deformed bubble. The bottom panels of this figure show a zoom on the initial and final bubble location and the underlying initial and final meshes.

In Fig. 24 (left column) we present density at several intermediate time moments for a zoomed region around the bubble ($t = 1005.15 \times 10^{-6}$ (top panel), $t = 1101.44 \times 10^{-6}$ (middle panel) and $t = t_{end} = 1342.153 \times 10^{-6}$ (bottom panel)). Color maps related to coloring by initial region are presented in right column of Fig. 24 (white color for the cells originally in the air; red color for the cells originally in the bubble). One can conclude that Lagrangian motion is well preserved by ReALE method.

In Fig. 25 (left panel) we present the cell-centered value $I_c = C_{h,c}(1 - C_{h,c})$ at final time $t_{end} = 1342.153 \times 10^{-6}$, where $C_{h,c}$ is the concentration of helium in cell c. Originally $C_{h,c}$ is equal to 1 if generator c belongs to helium bubble and 0 otherwise. Therefore, I_c is initially 0 for all cell. As the simulation advances, mixed cells are created close to the interface between the different materials, leading to values $I_c \neq 0$. The I_c map show how much interface is diffused; grayscale spreads from $I_c = 0$ (white) to $I_c = 1/4$ (black). As seen in left panel in Fig. 25 interface region is spread only over one or two cells.

Experimental results for this test case can be found in [62] and high-resolution numerical experiments performed with an AMR Eulerian code in [103]. In Fig. 25 (middle and right panels) a comparison of the Schlieren image of the experimental results from [62] (right panel) with our numerical results (middle panel) at $t_{end} = 1342.153 \times 10^{-6}$.

Presented numerical results show that qualitatively ReALE technique can reproduce experimental results (Fig. 25). We are planning to perform quantitative comparison with available experimental results as well as with high-resolution numerical results presented in [103] in special paper.

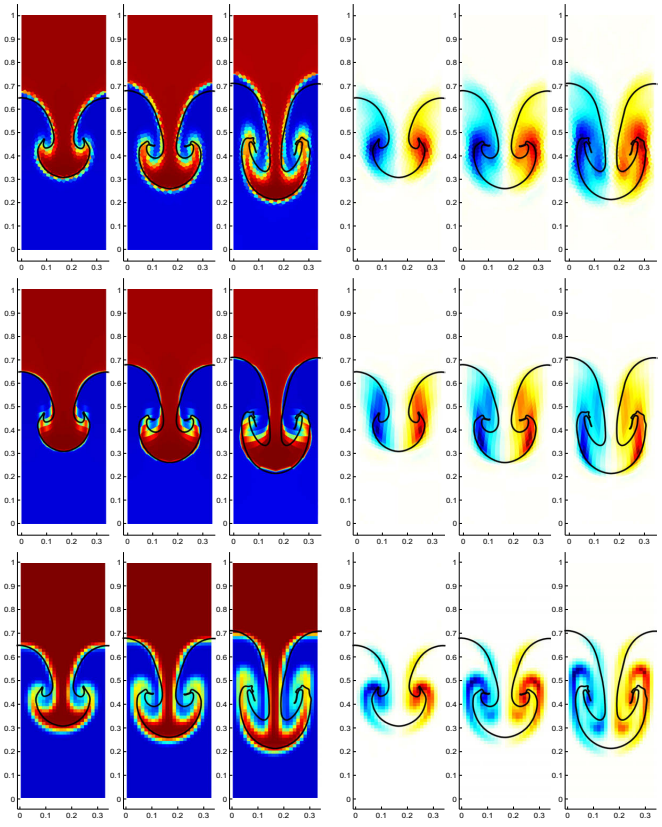


Fig. 27. Rayleigh–Taylor instability – Top panel: ReALE results. Middle panel: ALE results. Bottom panel: RTI3D Eulerian code results – Left to Right: Density at $t = 7, 8, 9$, Vorticity at $t = 7, 8, 9$. Color scale is from blue (minimal negative vorticity) to red (maximal positive vorticity). White color corresponds to zero vorticity. Frontier interface is the thick black line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

9.4. Rayleigh–Taylor instability

Our final test in this paper is Rayleigh–Taylor instability problem. It consists of two ideal gases with densities $\rho_{top} = 2$ and $\rho_{bottom} = 1$; in both cases the adiabatic constant $\gamma = 1.4$. Initially, the heavier gas is above the lighter gas in rectangular vessel $\Omega = [0{:}1/3] \times [0{:}1]$, with gravitational field directed vertically downward and with magnitude $g = 0.1$. The interface has been deliberately perturbed as described by formula $y(x) = \frac{1}{2} + \frac{1}{100}\cos(6\pi x)$. Initially both gases are at the rest; the pressure distribution is approximately hydrostatic and is defined in the lighter gas

$$p_{bottom}(x, y) = 1 + \frac{1}{2}\rho_{top} g + \rho_{bottom} g\left(\frac{1}{2} - y\right),$$

and in the heavier gas:

$$p_{top}(x, y) = 1 + \rho_{top} g(1 - y).$$

It is well known that this configuration is unstable and as time progresses, the heavier gas will sink and the lighter gas will rise through the formation of bubbles and spikes. This problem does not involve any shock wave, but the vorticity is so high that pure Lagrangian schemes eventually fail.

9.4.1. Initial phase of instability

The ReALE simulation starts with a Voronoi cleaned mesh obtained *via* 24×72 generators (24 in x-direction, 72 in y, see Fig. 26(a)). These are initially set so that the interface $y(x)$ is well approximated by edges of cells (see Fig. 26(a)).

In Fig. 26 we present the density and mesh for several time moments: $t = 0, 3, 5, 8, 9$.

As a matter of comparison and verification we use results obtained by others methods:

- FronTier (front tracking) code [41,58,59] is used to get a reference solution for the interface between the two fluids. The results of this code are used by the courtesy of J. Grove of the Los Alamos National Laboratory. FronTier is run with much more finer resolution (106×320 cells). This interface is plotted with a black thick lines in Fig. 27.
- RTI3D code (an implicit large eddy simulation (ILES) incompressible Eulerian code based on control volume approach, using a second-order Van-Leer method for volume fraction and momentum advection, see [4]). It is used with a grid resolution of 24×74 fixed cells. This Eulerian code provides a qualitative spatial and temporal behavior of vorticity. The results of this code are used by the courtesy of M.J. Andrews of the Los Alamos National Laboratory.
- Finally the CHIC-based ALE code without reconnection is used with an initial logically rectangular grid of 24×72.

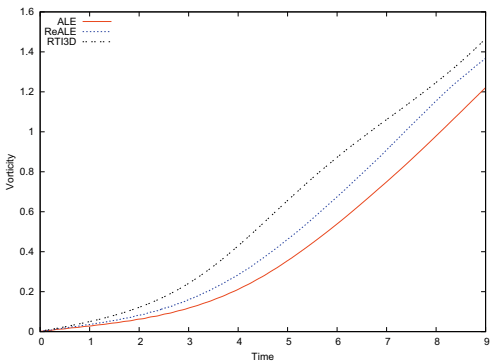


Fig. 28. Time evolution of the mean vorticity for the Rayleigh–Taylor instability – ReALE, ALE and RTI3D results are displayed.

In Fig. 27 we present the density ($t = 7, 8, 9$) and the vorticity ($t = 7, 8, 9$) for the ReALE, ALE codes (top and middle panels) and the RTI3D code (bottom panel). The FronTier interface is plotted as a thick black line on the top of each graph. The color scale for density is the same for all calculations, it is from 1 to 2. The general shape is resolved reasonably well by all three methods The tip of the interface is better resolved by ReALE method, Fig. 27 – left panels.

In the right panels in Fig. 27 we present color maps for vorticity, $\mathbf{curl}_c^h\mathbf{u}$ which is finite difference approximation of **curlu** in the cell Ω_c based on Green formula. For vorticity we have following ranges. For time $t = 7$ – ReALE: −3.53/3.55; ALE: −3.04/

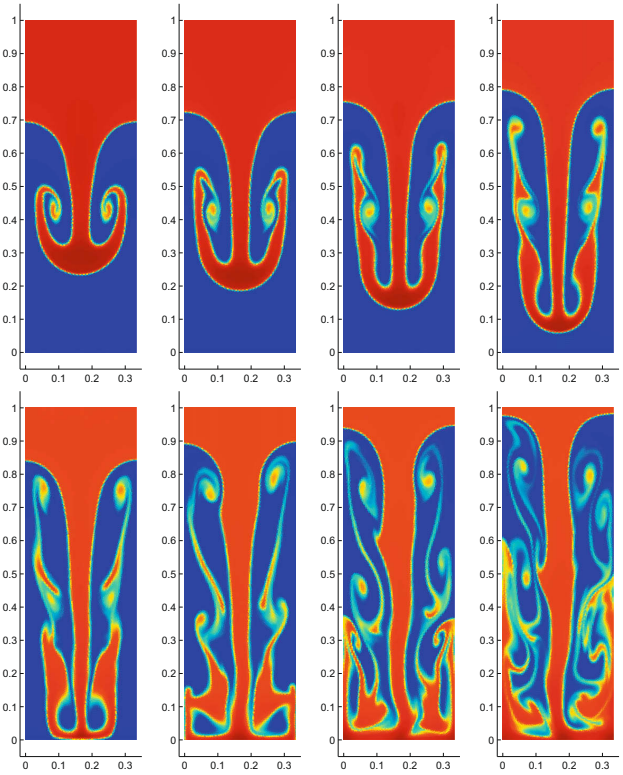


Fig. 29. Rayleigh–Taylor instability – ReALE with 66×200 generators at $t = 8, 9, 10, 11, 12, 13, 14, 15$ (from top-left to bottom-right) – Density.

3.04; RTI3D: −5.00/5.00. For time $t = 8$ – ReALE: −3.72/3.84; ALE: −3.68/3.68; RTI3D: −4.95/4.95. For time $t = 9$ – ReALE: −3.81/3.89; ALE: −4.52/4.52; RTI3D: −5.61/5.61.

We think that because RTI3D is incompressible code its vorticity color map is more sharp. ReALE clearly gives results closer to RTI3D and much sharper than ALE results. The ReALE results seem to better match spatial distribution of the vorticity, which matches interface shape. We wish to point that with this low resolution the ALE simulation code produces acceptable results. However, if a finer resolution is to be used, the mesh becomes very pinched and stretched so that numerical oscillations are generated (see Fig. 1 from the motivation section as example). As a consequence vorticity and, in general, most physical variables are contaminated leading to stability issues. In ReALE due to reconnection mesh follows the fluid, which allows to obtain more meaningful results (see Section 9.4.2 for high-resolution ReALE results).

Finally in Fig. 28 we show the mean vorticity, as a function of time for all three methods ReALE, ALE and RTI3D. Discrete mean vorticity is defined as follows:

$$\sqrt{\frac{\sum_c \left\{ |\mathbf{curl}_c^h\mathbf{u}|^2 |\Omega_c| \right\}}{|\Omega|}} \sim \sqrt{\frac{\int_\Omega |\mathbf{curlu}|^2 dV}{|\Omega|}}.$$

We observe that the ReALE suppresses vorticity less than ALE does. Qualitatively time evolution of mean vorticity is very similar for all three methods.

9.4.2. ReALE simulation of the later stages of the Rayleigh–Taylor instability

In this section we present high-resolution (66×200 generators are used leading to 13,200 cells) results for later stages of Rayleigh–Taylor instability. We perform our calculations up to time $t = 15$ when the heavy fluid has reached the bottom of the vessel and lighter fluid almost reached top of the vessel. In Fig. 29 we present color maps of the density at time moments $t = 8, 9, 10, 11, 12, 13, 14, 15$. In the last three figures one can clearly see violation of symmetry with respect to central vertical line. This is because this flow is very unstable and also because cleaning procedure in our rezone strategy is non-symmetric and depends on cell ordering.

Finally, in Fig. 30, we present density, vorticity and cells colored by initial domain at time $t = 15$. Right panel in Fig. 30 also shows the mesh.

Results presented in this section shows that ReALE method can be used to run problems with strong shear deformation without any special tuning of the parameters of the rezone strategy during the calculations. This is impossible for standard ALE methods without mesh reconnection, where user intervention is usually required.

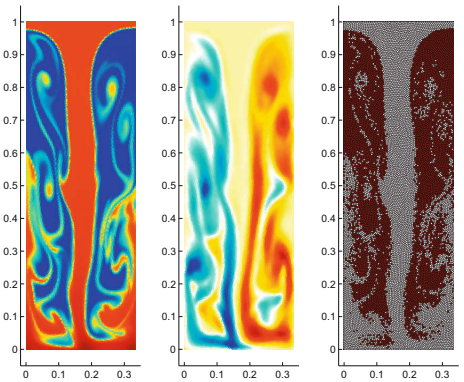


Fig. 30. Rayleigh–Taylor instability – ReALE with 66×200 generators at $t = 15$ – Density, vorticity and, mesh and cells colored by their initial domain; Vorticity scale is from −11.78 to 9.63 (blue for the minimal value to red for the maximal one). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

10. Conclusion and perspectives

We have presented a new reconnection-based ALE method. It includes three main elements.

- An explicit Lagrangian phase in which the solution on polygonal mesh is updated (without changing mesh connectivity).
- A rezoning phase in which a new grid is defined using specific movement of generators and formalism of Voronoi diagrams. It allows the change of mesh connectivity. In this work we keep number of cells unchanged, but number of vertices of each cell can change due to connectivity evolution. Generator movement is chosen in such a way that cell movement is close to Lagrangian and cell shape is close to regular hexagon.
- A remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) from one polygonal mesh to another.

On numerical examples we have demonstrated that our new method is more accurate and robust in comparison with standard ALE methods with fixed connectivity.

We recognize that our new method requires more testing, which we are planning to do in the future. We also recognize that the question of efficiency is very important. The particular implementation of our method used to obtain numerical results in this paper was not intended to be optimal and uses pieces which originally were not intended to work together. For this reason we do not present any comparison of efficiency of ALE and ReALE. We will do it in the future paper.

Also we are planning to explore different mechanisms for mesh adaptation. In the framework of current ReALE method at rezone stage we can develop new strategies for the choice of ω parameter to reflect features of the flow. We also could move vertices (as in standard ALE) after reconnection is done. We are planning to use mechanism of weighted Voronoi diagrams [44] to introduce adaptivity by choosing weight proportional to some monitor function, which may be an error indicator or may just reflect some physics which requires mesh refinement. The adaptation can be also achieved by adding or deleting cells. In principle, ReALE-like methods are perfectly suited for this strategy (cf. [37]).

In next paper we are planning to present 2D results in $r - z$- axisymmetric geometry, which will allow to demonstrate performance of our method for interesting and more realistic problems.

Also in future we will incorporate interface reconstruction methods such as volume of fluid (VOF) [105,106] and moment of fluid (MOF) [52,2,75]. Moreover, we plan to incorporate more advanced closure models for mixed multi-material cells [111]. This will increase accuracy of multi-material calculations.

There is no conceptual difficulties in extending ReALE methodology to 3D, however, first, we plan to improve efficiency of our new method in 2D by making code parallel.

Finally, we are planning to incorporate material strength into ReALE code. We are considering approach described in the recent paper [73].

Acknowledgments

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. The authors thank A. Solovjov for allowing to use his code for Voronoi mesh generation. The authors thank M. Kucharik, J. Dukowicz, F. Adessio, H. Trease, G. Ball, A. Barlow, P. Váchal, V. Ganzha, B. Wendroff, J. Campbell, D. Burton V. Tishkin, A. Favorskii, V. Rasskazova, N. Ardelyan, S. Sokolov for stimulating discussions over many years.

The authors also are very grateful to M.J. Andrews of the Los Alamos National Laboratory for the results of his RTI3D code on the Rayleigh–Taylor instability test problem. The authors also thank J. Grove from the Los Alamos National Laboratory for providing the Frontier results and allowing their use.

References

[1] F.L. Adessio, D.E. Carroll, K.K. Dukowicz, J.N. Johnson, B.A. Kashiwa, M.E. Maltrud, H.M. Ruppel, CAVEAT: a computer code for fluid dynamics problems with large distortion and internal slip, Technical Report LA-10613-MS, Los Alamos National Laboratory, 1986.
[2] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2007) 2096–2132.
[3] R.W. Anderson, N.S. Elliott, R.B. Pember, An arbitrary-Lagrangian–Eulerian method with adaptive mesh refinement for the solution of the Euler equations, J. Comput. Phys. 199 (2004) 598–617.
[4] M.J. Andrews, A. Lukyanov, computation of convective transport in transient two-phase flow, Int. J. Numer. Meth. Fluid 21 (3) (1995) 205–222.
[5] N.V. Ardelyan, A.V. Kosmachevskov, K.V. Kosmachevskii, M.N. Sablin, Numerical simulating 2D plasma dynamics problems on the base of implicit free-Lagrange method, in: Proceedings of WEHSFF 2007 (ECCOMAS Thematic Conference) – WEST-EAST HIGH SPEED FLOW FIELD CONFERENCE, November 19–22, 2007, Moscow, Russia, 2007. Available on web: <http://wehsff.imamod.ru/pages/s7.htm>.
[6] N.V. Ardelyan, K.V. Kosmatchevskii, Implicit free-Lagrange method for computing two-dimensional magnetogasdynamic flows, Comput. Math. Model. 6 (4) (1995) 209–224.
[7] F. Aurenhammer, Voronoi diagrams – a survey of fundamental geometric data structures, ACM Comput. Surveys 23 (3) (1991) 345–405.
[8] F. Aurenhammer, R. Klein, Voronoi diagrams, Technical Report 198, Fern Universität Hagen, Department of Computer Science, Germany, 1996. Available on web: <http://www.pi6.fernuni-hagen.de/publ/tr198.pdf>.
[9] M.J. Baines, Moving Finite Elements, Oxford Science Publications, Clarendon Press, Oxford, 1994.
[10] G.J. Ball, A free-Lagrange method for unsteady compressible flow: simulation of a confined cylindrical blast wave, Shock Waves 5 (1996) 311–325.

[11] G.J. Ball, Sliding interfaces in a 2D free-Lagrange Godunov scheme on a Voronoi mesh, in: Workshop on Numerical Methods for Multi-material Flows, Institut Henri Poincaré, Paris, September 23–25, 2002. Available on web: <http://www.ann.jussieu.fr/~despres/ABSTRACT/gjball_abs.pdf>.
[12] G.J. Ball, B.P. Howell, T.G. Leighton, M.J. Schofield, Shock-induced collapse of a cylindrical air cavity in water: a free-Lagrange simulation, Shock Waves 10 (2000) 265–276.
[13] R.A. Barabanov, O.I. Butnev, S.G. Volkov, B.M. Jogov, V.A. Pronin, Method Meduza-3D for computation of three-dimensional problems of gas dynamics on irregular meshes (In Russian), Quest. Atomic Sci. Tech. Ser.: Math. Mod. Phys. Process. 2 (2005) 15–26.
[14] A.J. Barlow, ALE and AMR mesh refinement techniques for multi-material hydrodynamics problems, ICFD Workshop on Mesh Refinement Techniques 7th December 2005, <http://www.icfd.rdg.ac.uk/Workshops/AMR/meshrefTechf.pdf>.
[15] A.J. Barlow, Challenges and recent progress in developing numerical methods for multi-material ALE hydrocodes, ICFD 25 Year Anniversary Conference 15–16th September 2008, Oxford University, <http://www.icfd.rdg.ac.uk/ICFD25/Talks/ABarlow.pdf>.
[16] A.J. Barlow, A compatible finite element multi-material ALE hydrodynamics algorithm, Int. J. Numer. Meth. Fluid 56 (2008) 953–964.
[17] T.J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: D. Kroner, M. Ohlberger, C. Rohde (Eds.), An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws, Lecture Notes in Computational Science and Engineering, Springer, Berlin, 1997, pp. 195–284.
[18] A.L. Bauer, D.E. Burton, E.J. Caramana, R. Loubère, M.J. Shashkov, P.P. Whalen, The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics, J. Comput. Phys. 218 (2) (2006) 572–593.
[19] D.J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs, Comput. Meth. Appl. Mech. Eng. 72 (1989) 305–350.
[20] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Meth. Appl. Mech. Eng. 99 (1992) 235–394.
[21] D.J. Benson, Momentum advection on a staggered mesh, J. Comput. Phys. 100 (1992) 143–162.
[22] A.F. Bower, Applied Mechanics of Solids, CRC Press, 2009.
[23] J.U. Brackbill, J.J. Monaghan (Eds.), Proceedings of the workshop on particle methods in fluid dynamics and plasma physics, Comput. Phys. Commun. 8(1), 1987.
[24] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. 46 (1982) 342–368.
[25] D.E. Burton, Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity, Advances in the Free-Lagrange Method, Springer-Verlag, New York, 1990.
[26] D.E. Burton, Consistent finite-volume discretization of hydrodynamics conservation laws for unstructured grids, Report UCRL-JC-118788, Lawrence Livermore National Laboratory, 1994.
[27] D.E. Burton, Multidimensional discretization of conservation laws for unstructured polyhedral grids, Report UCRL-JC-118306, Lawrence Livermore National Laboratory, in: Proceedings of Second International Workshop on Analytical Methods and Process Optimization in Fluid and Gas Mechanics (SAMGOP), Arzamas-16, Russia, 1994.
[28] J.C. Campbell, M.J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys. 172 (4) (2001) 739–765.
[29] J.C. Campbell, M.J. Shashkov, A compatible Lagrangian hydrodynamics algorithm for unstructured grids, Selttuk J. Appl. Math. 4 (2) (2003) 53–57.
[30] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1) (1998) 227–262.
[31] E.J. Caramana, R. Loubère, "Curl-q": a vorticity damping artificial viscosity for Lagrangian hydrodynamics calculations, J. Comput. Phys. 215 (2) (2006) 385–391.
[32] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distorsion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comput. Phys. 142 (1998) 521–561.
[33] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multidimensional shock wave computations, J. Comput. Phys. 144 (1998) 70–97.
[34] G. Carré, S. Del Pino, B. DesprTs, E. Labourasse, A cell-centered Lagrangian hydrodynamics scheme in arbitrary dimension, J. Comput. Phys. 228 (14) (2009) 5160–5183.
[35] R.A. Clark, Compressible Lagrangian hydrodynamics without Lagrangian cells, Report LAUR-85-1882, Los Alamos National Laboratory, 1985.
[36] R.A. Clark, The evolution of HDRO, Comput. Phys. Commun. 48 (1988) 61–64.
[37] M.C. Cline, J.K. Dukowicz, F.L. Addessio, CAVEAT-GT: a general topology version of the caveat code, Technical Report LA-11812-MS, Los Alamos National Laboratory, 1990.
[38] P. Colella, A direct Eulerian MUSCL scheme for gas dynamics, SIAM J. Stat. Comput. 6 (1) (1985) 104–117.
[39] W.P. Crowley, FLAG: a free-Lagrange method for numerically simulating hydrodynamic flows in two dimensions, in: M. Holt (Ed.), Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, vol. 8, September 15–19, 1970, University of California, Berkeley, Lecture Notes in Physics, Springer, Berlin, 1971, pp. 37–43.
[40] B. DesprTs, C. Mazeran, Lagrangian gas dynamics in two dimensions and Lagrangian systems, Arch. Rational Mech. Anal. 178 (2005) 327–372.
[41] J. Du, B. Fix, J. Glimm, X. Li, Y. Li, L. Wu, A simple package for front tracking, J. Comput. Phys. 213 (2) (2006) 613–628.
[42] Q. Du, M. Emelianenko, M. Gunzburger, Convergence of the Lloyd algorithm for computing centroidal Voronoi tesselations, SAIM J. Numer. Anal. 44 (1) (2006) 102–119.
[43] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tesselations: applications and algorithms, SIAM Rev. 41 (1999) 637–676.
[44] Q. Du, M. Gunzburger, Grid generation and optimization based on centroidal Voronoi tesselations, Appl. Math. Comput. 133 (2002) 591–607.
[45] J.K. Dukowicz, M.C. Cline, F.S. Addessio, A general topology Godunov method, J. Comput. Phys. 82 (1989) 29–63.
[46] J.K. Dukowicz, B. Meltz, Vorticity errors in multidimensional Lagrangian codes, J. Comput. Phys. 99 (1992) 115–134.
[47] J.K. Dukowicz, A simplified adaptive mesh technique derived from the moving finite element method, J. Comput. Phys. 56 (1984) 324–342.
[48] J.K. Dukowicz, J. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (2000) 318–335.
[49] J.K. Dukowicz, J.W. Kodis, Accurate conservative remapping (rezoning) for arbitrary-Lagrangian–Eulerian computations, SIAM J. Stat. Comput. 8 (1987) 305–321.
[50] V.F. Dyachenko, The free point method for problems of continuous media, USSR Comp. Math. Math. Phys. 5 (4) (1965) 680–688.
[51] V.F. Dyachenko, Some new method for numerically solving nonstationary gas dynamics problems, Comput. Meth. Appl. Mech. Eng. 2 (3) (1973) 265–277.
[52] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (2008) 5361–5384.
[53] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of the flow problems on the moving grids, J. Comput. Phys. 174 (2001) 669–694.
[54] A.P. Favorskij, M. Yu. Shashkov, A.V. Solovjov, V.F. Tishkin, The Dirichlet particle method for the simulation of 2-D gas dynamic flows with strong deformations, in: A.A. Samarskii, M.P.Sapagovas (Eds.), Mathematical Modelling and Applied Mathematics, Proceedings of IMACS International Conference, Moscow/USSR June 18–23, 1990, pp. 153–162 (Elsevier Science Publishers North Holland, Amsterdam, 1992).
[55] M.J. Fritts, W.P. Crowley, H. Trease (Eds.), The Free-Lagrange Method: Proceedings of the First International Conference of Free-Lagrange Methods, Lecture Notes in Physics, vol. 238, Springer-Verlag, New York, 1985.
[56] R.R. Giddings, Mesh movement via optimal transportation, in: Conference on Numerical Methods for Multi-material Flows Czech Technical University in Prague on September 10–14, 2007. Available on web: <http://www-troja.fjfi.cvut.cz/~multimat07/prezentace.html>.
[57] S. Giuliani, An algorithm for continuous rezoning of the hydrodynamic grid in arbitrary-Lagrangian–Eulerian computer codes, Nucl. Eng. Des. 72 (1982) 205–212.
[58] J. Glimm, J. Grove, X.-L. Li, K.-M. Shyue nad, Q. Zhang, Y. Zeng, Three-dimensional front tracking, SIAM J. Sci. Comput. 19 (1998) 703–727.
[59] J. Glimm, J.W. Grove, X.L. Li, W. Oh, D.H. Sharp, A critical analysis of Rayleigh–Taylor growth rates, J. Comput. Phys. 169 (2001) 652–677.

[60] S.K. Godunov, A. Zabrodine, M. Ivanov, A. Kraiko, G. Prokopov, RTsolution numTrique des problFmes multidimensionnels de la dynamique des gaz, Mir, 1979.
[61] J. Grandy, Conservative remapping and region overlays by intersecting arbitrary polyhedra, J. Comput. Phys. 148 (2) (1999) 133–466.
[62] J.-F. Haas, B. Sturtevant, Interaction of weak-shock waves, J. Fluid Mech. 181 (1987) 41–76.
[63] V.M. Hazins, V.V. Svetsov, A conservative stable smoothness-enhancing free-Lagrangian method, J. Comput. Phys. 105 (2) (1993) 187–198.
[64] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary-Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (1974) (reprinted in J. Comput. Phys. 135 (1997) 203–216).
[65] P. Hoch, Mesh quality and conservative projection in Lagrangian compressible hydrodynamic, in: Conference on Numerical Methods for Multi-material Fluid Flows, Czech Technical University in Prague on September 10–14, 2007, <http://www-troja.fjfi.cvut.cz/multimat07/presentations/tuesday/Rebourcet_H>.
[66] W.G. Hoover, Smooth Particle Applied Mechanics: The State of the Art, World Scientific Publishing Co. Pvt. Ltd., 2006.
[67] B.P. Howell, G.J. Ball, Damping of mesh-induced errors in free-Lagrange simulations or Richmayer–Meshkov instability, Shock Waves 10 (2000) 253–264.
[68] B.P. Howell, G.J. Ball, A free-Lagrange augmented Godunov method for simulation of elastic–plastic solids, J. Comput. Phys. 175 (2002) 128–167.
[69] Y. Huang, H. Quin, D. Wang, Centroidal Voronoi tesselation-based finite element superconvergence, Int. J. Numer. Meth. Eng. 76 (2008) 1819–1839.
[70] D.S. Kershaw, M.K. Prasad, M.J. Shaw, J.L. Milovich, 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method, Comput. Meth. Appl. Mech. Eng. 158 (1998) 81–116.
[71] P. Kjellgren, J. Hyvarien, An arbitrary-Lagrangian–Eulerian finite element method, Comput. Mech. 21 (1998) 81–90.
[72] R. Klein, A. Lingas, A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon, Int. J. Comput. Geomet. Appl. 6 (3) (1996) 263–278.
[73] G. Kluth, B. Després, Discretization of the hyperelasticity on unstructured mesh with cell-centered scheme, Technical Report R09063, Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie, 2009. Available on web: <http://www.ann.jussieu.fr/publications/2009/R09063.html>.
[74] P. Knupp, L.G. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary-Lagrangian–Eulerian methods, J. Comput. Phys. 176 (2002) 93–128.
[75] M. Kucharík, R. Garimella, S. Schofield, M. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, J. Comput. Phys. 229 (7) (2010) 2432–2452.
[76] M. Kucharík, M. Shashkov, Extension of efficient, swept-integration-based conservative remapping method for meshes with changing connectivity, Int. J. Numer. Meth. Fluid 56 (8) (2007) 1359–1365.
[77] M. Kucharík, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping methods, J. Comput. Phys. 188 (2003) 462–471.
[78] H.J. Kull, Theory of the Rayleigh–Taylor instability, Phys. Rep. 206 (1991) 197–325.
[79] K. Lipnikov, M. Shashkov, The error-minimization-based strategy for moving mesh methods, Commun. Comput. Phys. 1 (1) (2006) 53–81.
[80] G.R. Liu, Mesh Free Methods, Moving Beyond the Finite Element Method, CRC Press, 2003.
[81] R. Lohner, C. Yang, Improved ALE mesh velocities for moving bodies, Commun. Numer. Meth. Eng. 12 (1996) 599–608.
[82] R. Loubère, First steps into ale inc(ubator) – version 2.0.0, Technical Report, Los Alamos National Laboratory Report LAUR-04-8840, 2004.
[83] R. Loubère, E.J. Caramana, The force/work differencing of exceptional points in the discrete, compatible formulation of Lagrangian hydrodynamics, J. Comput. Phys. 216 (1) (2006) 1–18.
[84] R. Loubère, M. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, J. Comput. Phys. 204 (23) (2004) 155–160.
[85] R. Loubère, M. Staley, B. Wendroff, The repair paradigm: new algorithms and applications to compressible flow, J. Comput. Phys. (2005) 385–404.
[86] P.-H. Maire, A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry, J. Comput. Phys. 228 (2009) 6882–6915.
[87] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys. 228 (2009) 2391–2425.
[88] P.-H. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered Lagrangian scheme for compressible flow problems, SIAM J. Sci. Comput. 29 (4) (2007) 1781–1824.
[89] P.-H. Maire, J. Breil, A second-order cell-centered Lagrangian scheme for two-dimensional compressible flow problems, Int. J. Numer. Meth. Fluid 56 (2008) 1417–1423.
[90] P.-H. Maire, J. Breil, S. Galera, A cell-centered arbitrary-Lagrangian–Eulerian (ALE) method, Int. J. Numer. Meth. Fluid 56 (2008) 1161–1166.
[91] P.-H. Maire, B. Nkonga, Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics, J. Comput. Phys. 228 (2009) 799–821.
[92] L.G. Margolin, M. Shashkov, P. Smolarkiewicz, A discrete operator calculus for finite difference approximations, Comp. Meth. Appl. Mech. Eng. 187 (2000) 365–383.
[93] L.G. Margolin, Introduction to "An arbitrary-Lagrangian–Eulerian computing method for all flow speeds", J. Comput. Phys. 135 (1997) 198–202.
[94] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (1) (2003) 266–298.
[95] N.V. Mikhailova, V.F. Tishkin, N.N. Tyurina, A.P. Favorskii, M.Y. Shashkovm, Numerical modelling of two-dimensional gas-dynamic flows on a variable-structure mesh, USSR Comput. Math. Math. Phys. 26 (5) (1988) 74–84.
[96] A. Okabe, B. Boots, K. Sugihara, S.N. Chiu, Spatial Tesselations, Concepts and Applications of Voronoi Diagrams, Wiley, 2000.
[97] T.I. Orlova, V.V. Svetsov, Deep penetration of projectile material into the planet during the impact, Earth Moon Planets 71 (3) (1995) 255–263.
[98] P.J. O'Rourke, M.S. Sahota, A variable explicit/implicit numerical method for calculating advection on unstructured meshes, J. Comput. Phys. 143 (1998) 312–345.
[99] J.R. Pasta, S. Ulam, Heuristic numerical work in some problems of hydrodynamics, Math. Tables Aids Comput. XIII (65) (1959) 1–12.
[100] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Comput. Meth. Appl. Mech. Eng. 187 (2000) 591–619.
[101] R. Pember, R. Anderson, Comparison of direct Eulerian Godunov and Lagrange plus remap artificial viscosity schemes for compressible flow, Technical Report 2001-2644, AIAA, 2001.
[102] I.G. Pushkina, V.F. Tishkin, Adaptive grids from Dirichlet cells for mathematical physics problems: a methodology for grid generation, examples, Math. Model. 12 (3) (2000) 97–109.
[103] J. Quirk, S. Karni, On the dynamics of a shock–bubble interaction, J. Fluid Mech. 318 (1996) 129–163.
[104] B. Rebourcet, Comments on the filtering of numerical instabilities in Lagrangian hydrocodes, in: Conference on Numerical Methods for Multi-material Fluid flows, Czech Technical University in Prague on September 10–14, 2007, <http://www-troja.fjfi.cvut.cz/multimat07/presentations/tuesday/Rebourcet_f>.
[105] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 121 (1998) 112–152.
[106] M. Rudman, Volume tracking methods for interfacial flow calculations, Int. J. Numer. Meth. Fluid 24 (1997) 671–691.
[107] M.S. Sahota, H.E. Trease, A three-dimensional free-Lagrange code for multimaterial flow simulations, Report LAUR-90-3383, Los Alamos National Laboratory, 1990.
[108] G. Scovazzi, Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations, Comput. Meth. Appl. Mech. Eng. 196 (2007) 966–978.

[109] G. Scovazzi, M.A. Christon, T.J.R. Hughes, J.N. Shadid, Stabilized shock hydrodynamics: I. A Lagrangian method, Comput. Meth. Appl. Mech. Eng. 196 (2007) 923–966.
[110] G. Scovazzi, E. Love, M.J. Shashkov, Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: theoretical framework and two-dimensional computations, Comput. Meth. Appl. Mech. Eng. 197 (2008) 1056–1079.
[111] M. Shashkov, Closure models for multimaterial cells in arbitrary-Lagrangian–Eulerian hydrocodes, Int. J. Numer. Meth. Fluid 56 (2007) 1497–1504.
[112] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, J. Comput. Phys. 198 (1) (2004) 265–277.
[113] M.Yu. Shashkov, A.V. Solovjov, A generalization of the notion of Dirichlet cell for non-convex domain, Preprint No. 32 of Keldysh Institute of Applied Mathematics, USSR Academy of Sciences, Moscow, Russia, 1990 (in Russian).
[114] M.Yu. Shashkov, A.V. Solovjov, Numerical simulation of two-dimensional flows by the free-Lagrangian flow simulations, Report TUM-M9105, Mathematisches Institut, Technische Universität, Munchen, 1991. Available at <http://cnls.lanl.gov/~shashkov>.
[115] I.D. Sofronov, V.V. Rasskazova, I.V. Nesterenko, The use of nonregular nets for solving two-dimensional non-stationary problems in gas dynamics, in: N.N. Yanenko, Yu.I. Shokin (Eds.), Numerical Methods in Fluid Dynamics, MIR, Moscow, 1984, pp. 82–121.
[116] S.S. Sokolov, A.I. Panov, A.A. Voropinov, I.G. Novikov, I.V. Sobolev, A.V. Yalozo, Method TIM for computation of three-dimensional problems of mechanics of continuum media using unstructured polyhedral Lagrangian meshes (in Russian), Quest. Atomic Sci. Tech. Ser.: Math. Mod. Phys. Process. 3 (2005) 37–52.
[117] S.S. Sokolov, A.A. Voropinov, I.G. Novikov, A.I. Panov, I.V. Sobolev, A.A. Pushkarev, Method TIM-2D for computation of problems of mechanics of continuum media using unstructured polygonal with arbitrary connections at the nodes (in Russian), Quest. Atomic Sci. Tech. Ser.: Math. Mod. Phys. Process. 4 (2006) 29–44.
[118] A. Solovjov, E. Solovjova, M. Shashkov, V. Tishkin, A. Favorskii, Approximation of finite difference operators on mesh of Dirichlet cells, Differ. Equat. 22 (1986) 863–872.
[119] V. Springel, E pur si muove: Galiliean-invariant cosmological hydrodynamical simulations on a moving mesh, Mon. Not. R. Astron. Soc. (in press), arXiv:0901.4107, <http://arxiv.org/pdf/0901.4107>.
[120] C. Stoker, C. Gay, F. Bay, J.-L. Chenot, A velocity approach for the ALE-method applied for 2D and 3D problems, in: Simulation of Material Processing: Theory, Methods and Applications, Balkema, Rotterdam, 1998.
[121] H. Trease, M.J. Fritts, W.P. Crowley (Eds.), Advances in the Free-Lagrange Method: Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method: Proceedings of the Next Free-Lagrange Conference, Lecture Notes in Physics, vol. 395, Springer-Verlag, New York, 1990.
[122] H.E. Trease, Three-dimensional free-Lagrangian method hydrodynamics, Comput. Phys. Commun. 48 (1988) 39–50.
[123] B. van Leer, Towards the ultimate conservative difference scheme, J. Comput. Phys. 32 (1979) 101–136.
[124] W.B. VanderHeyden, B.A. Kashiwa, Compatible fluxes from van Leer advection, J. Comput. Phys. 146 (1998) 1–28.
[125] J. von Neumann, R.D. Richtmyer, A method for the numerical calculations of hydrodynamical shocks, J. Appl. Phys. 21 (1950) 232–238.
[126] R. Whitehurst, A free-Lagrange method for gas dynamics, Mon. Not. R. Astron. Soc. 277 (1995) 655–680.
[127] M.L. Wilkins, Calculation of elastic plastic flow, Math. Comput. Phys. 3 (1964).
[128] A.M. Winslow, Equipotential zoning of two-dimensional meshes, Technical Report UCRL-7312, Lawrence Livermore National Laboratory, 1963.
[129] A.M. Winslow, Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh, J. Comput. Phys. 1 (1966) 149–172.

Computers & Fluids 46 (2011) 59–69

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

ReALE: A Reconnection Arbitrary-Lagrangian–Eulerian method in cylindrical geometry

Raphaël Loubère [a,*], Pierre-Henri Maire [b], Mikhail Shashkov [c]

[a] Institut de Mathématiques de Toulouse, CNRS, Université de Toulouse, France
[b] Commisariatá l'Energie Atomique CEA-CESTA, 33114 Le Barp, France
[c] Los Alamos National Laboratory, T-5, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 29 April 2010
Received in revised form 28 August 2010
Accepted 31 August 2010
Available online 21 September 2010

Keywords:
ReALE
Cylindrical geometry
Lagrangian hydrodynamics
Voronoi mesh
Arbitrary-Lagrangian–Eulerian
Mesh reconnection
Polygonal mesh

ABSTRACT

This paper deals with the extension to the cylindrical geometry of the recently introduced Reconnection algorithm for Arbitrary-Lagrangian–Eulerian (ReALE) framework. The main elements in standard ALE methods are an explicit Lagrangian phase, a rezoning phase, and a remapping phase. Usually the new mesh provided by the rezone phase is obtained by moving grid nodes without changing connectivity of the underlying mesh. Such rezone strategy has its limitation due to the fixed topology of the mesh. In ReALE we allow connectivity of the mesh to change in rezone phase, which leads to general polygonal mesh and permits to follow Lagrangian features much better than for standard ALE methods. Rezone strategy with reconnection is based on using Voronoi tesselation machinery. In this work we focus on the extension of each phase of ReALE to cylindrical geometry. The Lagrangian, rezone with reconnection and remap phases are revamped to take into account the cylindrical geometry. We demonstrate the efficiency of our ReALE in cylindrical geometry on series of numerical examples.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

A new reconnection-based Arbitrary-Lagrangian–Eulerian (ALE) framework called Reconnection ALE (ReALE) has been recently introduced in [1]. The main elements in standard ALE methods are an explicit Lagrangian phase, a rezoning phase, and a remapping phase. Usually the new mesh provided by the rezone phase is obtained by moving grid nodes without changing connectivity of the underlying mesh. Such rezone strategy has its limitation due to the fixed topology of the mesh and may lead to stagnation of the mesh in certain situations [1]. Contrarily to classical ALE framework, the rezone part of ReALE allows topological mesh reconnection using the machinery of Voronoi tesselation [2]. The new feature of this technique is an underlying set of generators moving with the fluid as "pseudo-Lagrangian particles". The new generator position is a combination between its Lagrangian new position and the displaced Lagrangian cell centroid. These generators, as particles, can change neighbors especially when shear or vortex motions occur. The Voronoi machinery is then used on this set of generators to define the rezone mesh: Each generator is associated to the same Voronoi cell which, accordingly, may have changed its neighborhood. This Voronoi mesh is the rezone mesh onto which the physical variables are further remapped. Consequently as the Lagrangian and rezone meshes are *a priori* different, the conservative remap phase must be modified to handle polygonal meshes possibly with different connectivity.

In [1] the 2D Cartesian geometry was only considered as to prove the feasibility of this ALE with reconnection approach. Contrarily in this work we investigate the extension of ReALE to cylindrical geometry. Although staggered and cell-centered Lagrangian schemes were considered in [1] to prove the generality of ReALE, in this work we focus on Lagrangian cell-centered discretization because the presentation and implementation are simpler. However there is no theoretical limitation in using a staggered placement of variable for ReALE in cylindrical geometry. High-order cell-centered discretization of the Lagrangian hydrodynamics equations has been described [3]; all conserved quantities, including momentum, and hence cell velocity are cell-centered. Extension to cylindrical geometry has also been studied in [4] in a control volume or area-weighted discretization. The control volume scheme conserves momentum, total energy and satisfies a local entropy inequality in its first-order semi-discrete form. The main difference between these approaches relies on the problem of preserving spherical symmetry in two-dimensional cylindrical geometry. Being given a one-dimensional spherical flow on a polar grid, equally spaced in angle, Maire [4] analyzed the ability of the schemes to maintain spherical symmetry. It turns out that the control volume formulation does not preserve symmetry whereas the area-weighted formulation does similarly to staggered Lagrangian schemes [5]. In the context of ReALE the preservation of symmetric polar grid is not a goal as we are dealing with polygonal meshes by nature. We leave this issue for later investigation. However this cylindrical geometry extension of ReALE is motivated since in many application problems, such as inertial confinement problems, physical domains have axisymmetric features. This paper is organized as follows; we first recall some notion of cylindrical geometry, then in a second section we derive the cell-centered Lagrangian scheme. In the third section the rezone and remap parts are extended to cylindrical geometry. Numerical test case are provided in the fourth section where comparisons to exact solution and/or experimental solution are proposed. Finally conclusions and perspectives are drawn.

* Corresponding author. Tel.: +33 561 557 652.
E-mail address: raphael.loubere@math.univ-toulouse.fr (R. Loubère).

0045-7930/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2010.08.024

2. Cylindrical geometry

We are interested in discretizing the equations of the 2D Lagrangian hydrodynamics in cylindrical geometry, taking into account under the same form both Cartesian and cylindrical geometry. To this end, we re-use the notations introduced by Dukowicz in [6]. In the Lagrangian formalism the rates of change of mass, volume, momentum and total energy are computed assuming that the computational volumes follow the material motion. This representation leads to the following set of equations for an arbitrary moving control volume $V(t)$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \rho \,\mathrm{d}V = 0, \tag{1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \mathrm{d}V - \int_{V(t)} \nabla\cdot\boldsymbol{U}\,\mathrm{d}V = 0, \tag{1b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \rho\boldsymbol{U}\,\mathrm{d}V + \int_{V(t)} \nabla P\,\mathrm{d}V = \boldsymbol{0}, \tag{1c}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)} \rho E\,\mathrm{d}V + \int_{V(t)} \nabla\cdot(P\boldsymbol{U})\,\mathrm{d}V = 0. \tag{1d}$$

where $\frac{\mathrm{d}}{\mathrm{d}t}$ denotes the material, or Lagrangian, time derivative. Here, ρ, $\boldsymbol{U}$, P, E respectively denote the mass density, velocity, pressure and specific total energy of the fluid. Eqs. 1a, 1c, 1d express the conservation of mass, momentum and total energy. The thermodynamic closure is obtained by adding the Equation Of State (EOS) of the form $P = P(\rho,\varepsilon)$, where the specific internal energy, ε, is related to the specific total energy by $\varepsilon = E - \frac{1}{2}\|\boldsymbol{U}\|^2$. We note that volume variation Eq. (1b) which is also named Geometric Conservation Law (GCL), is equivalent to the local kinematic equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X} = \boldsymbol{U}(\boldsymbol{X}(t),t), \quad \boldsymbol{X}(0) = \boldsymbol{x}, \tag{2}$$

where $\boldsymbol{X}$ is a point located on the control volume surface, $S(t)$, at time $t > 0$ and $\boldsymbol{x}$ corresponds to its initial position. We note that the case of Cartesian or cylindrical geometry can be combined by introducing the pseudo Cartesian frame (O,X,Y), equipped with the orthonormal basis $(\boldsymbol{e}_X,\boldsymbol{e}_Y)$, through the use of the pseudo radius $\mathscr{R}(Y) = 1 - \alpha + \alpha Y$, where $\alpha = 1$ for cylindrical geometry and $\alpha = 0$ for Cartesian geometry. We remark that Y corresponds to the radial coordinate in the cylindrical case meaning that we assume rotational symmetry about X-axis, refer to Fig. 1. We note that if we refer to standard cylindrical coordinates, (Z,R), then X corresponds to Z and Y to R. In this framework, the volume V is obtained by rotating the area A about the X-axis. Thus, the volume element, $\mathrm{d}V$, writes $\mathrm{d}V = \mathscr{R}\,\mathrm{d}A$, where $\mathrm{d}A = \mathrm{d}X\mathrm{d}Y$ is the area element in the pseudo Cartesian coordinates. Note that we have omitted the factor 2π due to the integration in the azimuthal direction, namely we consider all integrated quantities to be defined per unit radian.

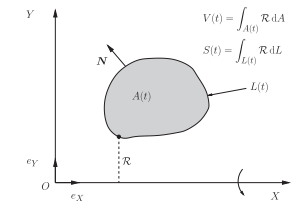


Fig. 1. Notation related to the pseudo Cartesian geometry.

The surface S, which bounds the volume V, is obtained by rotating, L, the boundary of the area A, about the X-axis. Thus, the surface element, $\mathrm{d}S$, writes $\mathrm{d}S = \mathscr{R}\,\mathrm{d}L$, where $\mathrm{d}L$ is the line element along the perimeter of A.

In view of subsequent spatial discretization, we shall express the volume integrals associated with the divergence and gradient operators using the Green formula. We recall that, in the pseudo Cartesian frame, the divergence operator writes

$$\nabla\cdot\boldsymbol{U} = \frac{\partial u}{\partial X} + \frac{1}{\mathscr{R}}\frac{\partial}{\partial Y}(\mathscr{R}v) = \frac{\partial u}{\partial X} + \frac{\partial v}{\partial Y} + \alpha\frac{v}{\mathscr{R}} = \frac{1}{\mathscr{R}}\left[\frac{\partial}{\partial X}(\mathscr{R}u) + \frac{\partial}{\partial Y}(\mathscr{R}v)\right]$$

where (u,v) are the components of the vector $\boldsymbol{U}$. The gradient operator writes as usual

$$\nabla P = \frac{\partial P}{\partial X}\boldsymbol{e}_X + \frac{\partial P}{\partial Y}\boldsymbol{e}_Y.$$

By replacing the volume integral form of the divergence operator by its surface integral form and by employing the previous notations one deduces the Green formula in the pseudo Cartesian framework as

$$\int_V \nabla\cdot\boldsymbol{U}\,\mathrm{d}V = \int_L \boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L. \tag{3}$$

where $\boldsymbol{N}$ is the unit outward normal associated with the contour L. To derive the surface integral form of the gradient operator, we use the vector identity $\boldsymbol{U}\cdot\nabla P = \nabla\cdot(P\boldsymbol{U}) - P\nabla\cdot\boldsymbol{U}$, which holds for any vector $\boldsymbol{U}$. The integration of this identity over the volume V leads to

$$\int_V \boldsymbol{U}\cdot\nabla P\,\mathrm{d}V = \int_L P\boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L - \int_A P\nabla\cdot\boldsymbol{U}\mathscr{R}\,\mathrm{d}A.$$

Assuming a constant $\boldsymbol{U}$ vector, we finally get

$$\int_V \nabla P\,\mathrm{d}V = \int_L P\boldsymbol{N}\mathscr{R}\,\mathrm{d}L - \alpha\boldsymbol{e}_Y\int_A P\,\mathrm{d}A, \tag{4}$$

since for a constant $\boldsymbol{U}$ vector, we have $\nabla\cdot\boldsymbol{U} = \frac{\alpha}{\mathscr{R}}\boldsymbol{U}\cdot\boldsymbol{e}_Y$. We have expressed the volume integral of the gradient operator as a function of a surface integral plus a source term, which ensures the compatibility with the surface integral form of the divergence operator. This approach leads to a discretization which is known as *Control Volume formulation* (CV). An alternative approach to define the surface integral form of the gradient operator is obtained by setting

$$\int_V \nabla P\,\mathrm{d}V = \int_A \nabla P\mathscr{R}\,\mathrm{d}A = \overline{\mathscr{R}}\int_A \nabla P\,\mathrm{d}A.$$

Here, we have used the mean value theorem, hence $\overline{\mathscr{R}}$ is defined as the averaged pseudo radius $\overline{\mathscr{R}} = \frac{1}{|A|}\int_A \mathscr{R}\,\mathrm{d}A$, where $|A|$ is the surface

of the area A. We remark that in the case of Cartesian geometry $\overline{\mathscr{R}} = 1$ since $\alpha = 0$. Finally, applying the Green formula, we get

$$\int_V \nabla P\,\mathrm{d}V = \overline{\mathscr{R}}\int_L P\boldsymbol{N}\,\mathrm{d}L. \tag{5}$$

We recover the Cartesian definition of the gradient operator weighted by the averaged pseudo radius. This alternative approach leads to the so-called *Area-Weighted formulation* (AW). We point out that, in this case, the compatibility between the surface integrals of the divergence and gradient operators is lost. Finally let us remark that formulae (5) and (4) coincide in the case of the Cartesian geometry since $\alpha = 0$ and $\overline{\mathscr{R}} = 1$.

3. Compatible cell-centered Lagrangian scheme

We develop a sub-cell force-based discretization over a domain $\mathscr{D}$ which is paved using a collection of polygonal cells without gap or overlaps. Such discretization has been introduced in [7,8]. Using the previous results and particularly the gradient operator definition given by (4), we rewrite the set of Eq. (1) in the control volume formulation over the moving polygonal cell $\Omega_c(t)$ as

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \int_{\partial\Omega_c(t)} \boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L = 0, \tag{6a}$$

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_c + \int_{\partial\Omega_c(t)} P\boldsymbol{N}\mathscr{R}\,\mathrm{d}L = \alpha A_c P_c\boldsymbol{e}_Y, \tag{6b}$$

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}E_c + \int_{\partial\Omega_c(t)} P\boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L = 0. \tag{6c}$$

Here, A_c is the area of the cell $\Omega_c(t)$ and m_c its constant mass. For any fluid variable ϕ, ϕ_c denotes its mass density average, i.e. $\phi_c = \frac{1}{m_c}\int_{\Omega_c(t)} \rho\phi\,\mathrm{d}V$. The area-weighted formulation is obtained using (5) for the gradient operator definition. In comparison to the control volume formulation, the previous system only differs in the momentum equation. Using the notations previously introduced, the area-weighted formulation of the momentum equation writes

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_c + \overline{\mathscr{R}}_c\int_{\partial\Omega_c(t)} P\boldsymbol{N}\,\mathrm{d}L = 0, \tag{7}$$

where the cell averaged pseudo radius is $\overline{\mathscr{R}}_c = \frac{1}{A_c}\int_{A_c} \mathscr{R}\,\mathrm{d}A$. We point out that, in the case of Cartesian geometry $\overline{\mathscr{R}}_c = 1$ for all c, therefore the area-weighted formulation coincides with the control volume formulation. Moreover recalling that $m_c = V_c\rho_c$ and $\overline{\mathscr{R}}_c = V_c/A_c$ implies that (7) can be rewritten as

$$\mu_c\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_c + \int_{\partial\Omega_c(t)} P\boldsymbol{N}\,\mathrm{d}L = 0, \tag{8}$$

where $\mu_c = A_c\rho_c = m_c\overline{\mathscr{R}}_c$ denotes the Cartesian inertia. Consequently (8) has the same form as the momentum equation written in Cartesian geometry although the Cartesian inertia is not a Lagrangian mass (i.e it is not constant in time).

We have written a set of semi-discrete evolution equations for the cell-centered variables $\left(\frac{1}{\rho_c},\boldsymbol{U}_c,E_c\right)$, whose thermodynamic closure is given by the EOS, $P_c = P(\rho_c,\varepsilon_c)$, where $\varepsilon_c = E_c - \frac{1}{2}\|\boldsymbol{U}_c\|^2$. The motion of the grid is ruled by the discrete trajectory equation written at each point: $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}_p = \boldsymbol{U}_p(\boldsymbol{X}_p(t),t), \boldsymbol{X}_p(0) = \boldsymbol{x}_p$, where $\boldsymbol{X}_p$ denotes the position vector of point p and $\boldsymbol{U}_p$ its velocity. Let us note that by setting $\alpha = 0$ in the previous set of equations we recover the same system as in Cartesian geometry [3]. In the following we determine the numerical fluxes and the nodal velocity used to move the grid.

3.1. Geometric conservation law

Introducing $V_c = \int_{\Omega_c(t)} \mathscr{R}\,\mathrm{d}A$ the measure of the volume obtained by rotation of the polygonal cell Ω_c about X-axis, Eq. (6a) writes as the GCL

$$\frac{\mathrm{d}}{\mathrm{d}t}V_c - \int_{\partial\Omega_c(t)} \boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L = 0. \tag{9}$$

Likewise in the case of Cartesian geometry, we use the fact that V_c is a function of the position vector $\boldsymbol{X}_p$ of point $p \in \mathscr{P}(c)$, where $\mathscr{P}(c)$ denotes the set of points of the Lagrangian cell Ω_c. The cylindrical corner area vector, refer to Fig. 2 is given by

$$\mathscr{A}_{pc}\boldsymbol{N}_{pc} = \frac{1}{2}\left[\frac{\mathscr{R}_{p^-} + 2\mathscr{R}_p}{3}(\boldsymbol{X}_p - \boldsymbol{X}_{p^-}) + \frac{\mathscr{R}_{p^+} + 2\mathscr{R}_p}{3}(\boldsymbol{X}_{p^+} - \boldsymbol{X}_p)\right]\times\boldsymbol{e}_z$$

where $\mathscr{A}_{pc}$ is the corner area that can be computed knowing that $\boldsymbol{N}_{pc}^2 = 1$. Noticing that the half-edge outward normals are given by $L_{pc}^-\boldsymbol{N}_{pc}^- = \pm\frac{1}{2}(\boldsymbol{X}_{p^+} - \boldsymbol{X}_p)\times\boldsymbol{e}_z$, we rewrite the previous equation as

$$\mathscr{A}_{pc}\boldsymbol{N}_{pc} = \frac{\mathscr{R}_{p^-} + 2\mathscr{R}_p}{3}L_{pc}^-\boldsymbol{N}_{pc}^- + \frac{\mathscr{R}_{p^+} + 2\mathscr{R}_p}{3}L_{pc}^+\boldsymbol{N}_{pc}^+. \tag{10}$$

As noticed by Whalen in [9], the corner area vector is the fundamental geometric object that uniquely defines the time rate of change of the cell volume as

$$\frac{\mathrm{d}}{\mathrm{d}t}V_c = \sum_{pc\in\mathscr{P}(c)} \mathscr{A}_{pc}\boldsymbol{N}_{pc}\cdot\boldsymbol{U}_p. \tag{11}$$

This last result yields the definition of the discrete divergence operator over cell Ω_c as follows:

$$(\nabla\cdot\boldsymbol{U})_c = \frac{1}{V_c}\frac{\mathrm{d}}{\mathrm{d}t}V_c = \frac{1}{V_c}\sum_{pc\in\mathscr{P}(c)} \mathscr{A}_{pc}\boldsymbol{N}_{pc}\cdot\boldsymbol{U}_p. \tag{12}$$

We claim that we have completely defined the volume flux in terms of the corner area vector and the nodal velocity, moreover this derivation is compatible with the mesh motion.

3.2. Sub-cell force-based discretization

Let us discretize momentum and total energy equations by means of sub-cell forces. To this end we use the partition of each polygonal cell Ω_c into sub-cells Ω_{pc}, where $p \in \mathscr{P}(c)$ (see Fig. 2). The sub-cell force that acts from sub-cell onto point is defined as

$$\boldsymbol{F}_{pc} = \int_{\partial\Omega_{pc}\cap\partial\Omega_c} P\boldsymbol{N}\mathscr{R}\,\mathrm{d}L. \tag{13}$$

We also use the sub-cell based partition to approximate the total energy flux as

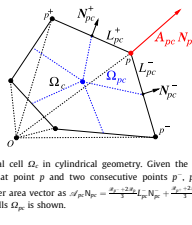


Fig. 2. Polygonal cell Ω_c in cylindrical geometry. Given the half-edge outward normals $L_{pc}^+\boldsymbol{N}_{pc}^+$ at point p and two consecutive points p^-, p^+ one defines the cylindrical corner area vector as $\mathscr{A}_{pc}\boldsymbol{N}_{pc} = \frac{\mathscr{R}_{p^-}+2\mathscr{R}_p}{3}L_{pc}^-\boldsymbol{N}_{pc}^- + \frac{\mathscr{R}_{p^+}+2\mathscr{R}_p}{3}L_{pc}^+\boldsymbol{N}_{pc}^+$. The partition into sub-cells Ω_{pc} is shown.

$$\int_{\partial\Omega_c} P\boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L = \sum_{pc\in\mathscr{P}(c)}\left(\int_{\partial\Omega_{pc}\cap\partial\Omega_c} P\boldsymbol{N}\mathscr{R}\,\mathrm{d}L\right)\cdot\boldsymbol{U}_p = \sum_{pc\in\mathscr{P}(c)}\boldsymbol{F}_{pc}\cdot\boldsymbol{U}_p.$$

Substituting the previous results into system (6c) yields

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \sum_{pc\in\mathscr{P}(c)} \mathscr{A}_{pc}\boldsymbol{N}_{pc}\cdot\boldsymbol{U}_p = 0, \tag{14a}$$

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_c + \sum_{pc\in\mathscr{P}(c)} \boldsymbol{F}_{pc} = \alpha P_c A_c\boldsymbol{e}_Y, \tag{14b}$$

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}E_c + \sum_{pc\in\mathscr{P}(c)} \boldsymbol{F}_{pc}\cdot\boldsymbol{U}_p = 0. \tag{14c}$$

We have expressed the numerical fluxes in terms of the corner area vector, the sub-cell force and the nodal velocity. The last two remain to be determined to complete the discretization. This task is achieved by investigating the thermodynamic consistency and the conservation of the sub-cell force-based discretization [4]. To ensure a local entropy inequality, it is sufficient to postulate the following form for the sub-cell force:

$$\mathrm{F}_{pc} = \mathscr{A}_{pc}P_c\mathrm{N}_{pc} - \mathrm{M}_{pc}(\mathrm{U}_p - \mathrm{U}_c). \tag{15}$$

Here M_{pc} is a sub-cell based 2×2 matrix such that: M_{pc} is symmetric, and, M_{pc} is positive semi-definite. The physical dimension of M_{pc} corresponds to an area times a density times a velocity. We remark that entropy production within cell c is directly governed by the general form of the sub-cell matrix M_{pc} and the velocity jump between the nodal and the cell-centered velocity, $\Delta\boldsymbol{U}_{pc} = \boldsymbol{U}_p - \boldsymbol{U}_c$. Finally total energy conservation is ensured provided that for all point p

$$\sum_{c\in\mathscr{C}(p)} \mathrm{F}_{pc} = \boldsymbol{0}. \tag{16}$$

We remark that this last equation is the same condition than the one obtained in Cartesian geometry for any compatible cell-centered or staggered sub-cell based discretization. Moreover under this condition, and, up to the boundary terms and the radial source term contributions, momentum is conserved over the entire domain. This result is remarkable in the sense that it is written under the same form regardless the geometry.

The last unknowns of the scheme, namely the sub-cell matrix M_{pc} and the node velocity $\boldsymbol{U}_p$, are obtained thanks to a node-centered Riemann solver.

3.3. Node-centered Riemann solver

The node-centered solver that provides the grid velocity is obtained as a consequence of total energy conservation. Substituting the sub-cell force (15) into (16) gives for all point p

$$\mathrm{M}_p\mathrm{U}_p = \sum_{c\in\mathscr{C}(p)} (\mathscr{A}_{pc}P_c\mathrm{N}_{pc} + \mathrm{M}_{pc}\mathrm{U}_c), \tag{17}$$

where M_p is the sum of the corner matrices around node p, which is defined as $\mathrm{M}_p = \sum_{c\in\mathscr{C}(p)}\mathrm{M}_{pc}$. We construct the natural extension of the Cartesian cell-centered scheme [3] to cylindrical geometry by defining the corner matrix as

$$\mathrm{M}_{pc} = Z_{pc}^-\mathscr{R}_{pc}^-L_{pc}^-\left(\mathrm{N}_{pc}^-\otimes\mathrm{N}_{pc}^-\right) + Z_{pc}^+\mathscr{R}_{pc}^+L_{pc}^+\left(\mathrm{N}_{pc}^+\otimes\mathrm{N}_{pc}^+\right), \tag{18}$$

where $\mathscr{R}_{pc}^+ = \frac{1}{3}(\mathscr{R}_{p^+} + 2\mathscr{R}_p)$. We recall that Z_{pc}^+ are the generalized non-linear corner impedances given by $Z_{pc}^+ = \rho_c[a_c + \Gamma_c|(\mathrm{U}_p - \mathrm{U}_c)\cdot\mathrm{N}_{pc}^+|]$, where a_c is the isentropic sound speed and Γ_c is a material dependent parameter, which is given by $\frac{\gamma+1}{2}$ in case of a gamma gas law. Note that this formula is the two-dimensional extension of the 2-shock swept mass flux defined for one-dimensional approximate Riemann problem initially proposed by Dukowicz [10] for shock wave. We also mention that we recover the acoustic approximation by setting $\Gamma_c = 0$. One can easily check that this definition leads to a symmetric positive definite M_{pc} matrix. Therefore, M_p is also symmetric positive definite and thus always invertible, which defines a unique nodal velocity $\boldsymbol{U}_p$ by inverting Eq. (17). Let us mention that this solver preserves the spherical symmetry in the case of a one-dimensional spherical flow computed on an equal angle polar grid.

The high-order extension of our control volume discretization, both in time and space, is obtained by using the Generalized Riemann Problem (GRP) methodology in the acoustic approximation (see [4] for the details). Moreover an extension of this cell-centered Lagrangian scheme in area-weighted formulation is also available [4]. For multi-species computation one simply considers the isopressure, iso-temperature closure model. Each fluid is characterized by its mass fraction C_k and during the Lagrangian phase, the concentration of each fluid evolves following the trivial equation $\frac{\mathrm{d}}{\mathrm{d}t}C_f = 0$ (refer to [1]).

4. Rezone and remap in cylindrical geometry

As mentioned in [1] ReALE consists in modifying the rezone and remap phases of an ALE code assuming that the Lagrangian scheme can handle polygonal mesh. The cell-centered Lagrangian scheme previously described in its control volume or area-weighted version is well suited for this purpose. Therefore it is adopted as the first phase of our ReALE algorithm. The extension of the rezone and remap phases is presented in the following subsections.

4.1. Rezone phase through Voronoi machinery

In cylindrical geometry the simulation is performed on an actual 2D mesh. Any notion of mesh symmetry is therefore to be considered in the plane (Z,R). Consequently, the rezone phase is assumed to operate on this plane behaving "as a Cartesian plane". Therefore neither the generator displacement nor the Voronoi machinery is modified compared to [1]. However because ReALE cornerstone lays in the generator displacement and for the sake of clarity we recall these steps.

Let Ω_c^n and Ω_c^{n+1} denotes the Lagrangian cells at time t^n and $t^{n+1} = t^n + \Delta t$, where Δt is the current time step. The position vector of the generator of the Lagrangian cell Ω_c^n is denoted G_c^n (see Fig. 3). We will define the new position of the generator at time t^{n+1}. First, we compute a Lagrangian-like displacement of the generator by setting

$$\mathrm{G}_c^{n+1,\mathrm{lag}} = \mathrm{G}_c^n + \Delta t\mathrm{U}_c, \tag{19}$$

where $\boldsymbol{U}_c$ is the "pseudo-Lagrangian" velocity of the generator within the cell. This velocity is computed so that the generator remains located in the new Lagrangian cell. To this end we define this velocity to be the average of the velocities of the points of the cell, namely $\mathrm{U}_c = \frac{1}{\mathscr{R}_c^{n+\frac{1}{2}}}\sum_{pc\in\mathscr{P}(c)}\mathrm{U}_p^{n+\frac{1}{2}}$. Here $\mathrm{U}_p^{n+\frac{1}{2}}$ is the time-centered velocity of point p between times t^n and t^{n+1}. Any other formula could be used, as instance by weighting the point velocity by the distance between G_c^n and X_c^n. Let us introduce the centroid of the Lagrangian cell $\mathrm{X}_c^{n+1} = \frac{1}{|\Omega_c^{n+1}|}\int_{\Omega_c^{n+1}}\mathrm{X}\,\mathrm{d}V$, where $|\Omega_c^{n+1}|$ denotes the volume of the cell Ω_c^{n+1}. The updated position of the generator is defined by mean of a convex combination between the new Lagrangian-like position, $\mathrm{G}_c^{n+1,\mathrm{lag}}$ and the centroid X_c^{n+1} of the Lagrangian cell

$$\mathrm{G}_c^{n+1} = \mathrm{G}_c^{n+1,\mathrm{lag}} + \omega_c\left(\mathrm{X}_c^{n+1} - \mathrm{G}_c^{n+1,\mathrm{lag}}\right), \tag{20}$$

where $\omega_c \in [0;1]$ is a parameter that remains to determine.With this convex combination, the updated generator lies in between its Lagrangian position at time t^{n+1} and the centroid of the Lagrangian cell Ω_c^{n+1}.We note that for $\omega_c = 0$ we get a Lagrangian-like

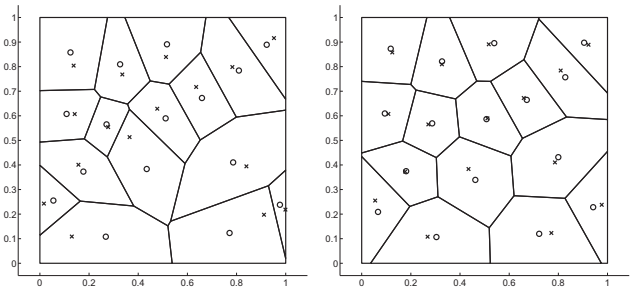


Fig. 3. Examples of Voronoi tesselation with cell generators $\boldsymbol{G}_c$ (×) and cell centroids $\boldsymbol{X}_c$ (○).

motion of the generator whereas for $\omega_c = 1$ we obtain a centroidal-like motion, which tends to produce a smooth mesh.[1] We compute ω_c requiring that the generator displacement satisfies the principle of material frame indifference, that is for pure uniform translation or rotation we want ω_c to be zero.To this end, we construct ω_c using invariants of the right Cauchy–Green strain tensor associated to the Lagrangian cell Ω_c between times t^n and t^{n+1}.Let us recall some general notions of continuum mechanics to define this tensor.First, we define the Cartesian deformation gradient tensor $\mathsf{F} = \frac{\partial \boldsymbol{X}^{n+1}}{\partial \boldsymbol{X}^n}$, where $\boldsymbol{X}^{n+1} = (X^{n+1}, Y^{n+1})^t$ denotes the vector position of a point at time t^{n+1} that was located at position $\boldsymbol{X}^n = (X^n, Y^n)^t$ at time t^n.The Cartesian deformation gradient tensor is the Jacobian matrix of the map that connects the Lagrangian configurations at time t^n and t^{n+1}.The right Cauchy–Green strain tensor, $\mathsf{C} = \mathsf{F}^t\mathsf{F}$, is a 2×2 symmetric positive definite tensor.We notice that this tensor reduces to the unitary tensor in case of uniform translation or rotation. It admits two positive eigenvalues, λ_1 and λ_2 with the convention $\lambda_1 \leqslant \lambda_2$.These can be viewed as the rates of expansion in a given direction during the transformation.To determine ω_c, we first construct the cell-averaged value of the deformation gradient tensor, F_c, and then the cell-averaged value of the Cauchy–Green tensor by setting $\mathsf{C}_c = \mathsf{F}_c^t\mathsf{F}_c$.Noticing that the two rows of the F matrix correspond to the gradient vectors of the X and Y-coordinates, we can set $\mathsf{F}^t = [\boldsymbol{\nabla}_n X^{n+1}, \boldsymbol{\nabla}_n Y^{n+1}]$, where for any functions $\psi = \psi(\boldsymbol{X}^n)$, we have $\boldsymbol{\nabla}_n\psi = (\frac{\partial\psi}{\partial X^n}, \frac{\partial\psi}{\partial Y^n})^t$. With these notations, one defines the cell-averaged value of the gradient of the ψ function over the Lagrangian cell Ω_c^n

$$(\boldsymbol{\nabla}_n\psi)_c = \frac{1}{|\Omega_c^n|}\int_{\Omega_c^n} \boldsymbol{\nabla}_n\psi \, d\mathcal{V} = \frac{1}{|\Omega_c^n|}\int_{\partial\Omega_c^n} \psi \mathrm{N}\, d\mathcal{S} \simeq \frac{1}{|\Omega_c^n|}\sum_{p=1}^{|\mathcal{P}(c)|} \frac{1}{2}\left(\psi_p^n + \psi_{p+1}^n\right) L_{p,p+1}^n \mathrm{N}_{p,p+1}^n \tag{21}$$

where $\psi_p^n = \psi(X_p^n)$ and $L_{p,p+1}^n \mathrm{N}_{p,p+1}^n$ is the unit outward normal to the edge $[X_p^n, X_p^{n+1}]$. In the previous equation, we have first used the Green formula then an approximation of the integral using the trapezoidal rule on a polygonal cell. Applying (21) to $\psi = X^{n+1}$ and $\psi = Y^{n+1}$ we get a cell-averaged expression of the gradient tensor F and, consequently deduce the cell-averaged value of the tensor C_c.

[1] This latter case is equivalent to perform one Lloyd iteration [11].

Knowing this symmetric positive definite tensor in each cell, we compute its real positive eigenvalues $\lambda_{1,c}$, $\lambda_{2,c}$ and define the parameter $\omega_c = \frac{1-\alpha_c}{1-\alpha_{\min}}$ where $\alpha_c = \frac{\lambda_{1,c}}{\lambda_{2,c}}$ and $\alpha_{\min} = \min_c \alpha_c$. We emphasize the fact that for uniform translation or rotation $\lambda_{1,c} = \lambda_{2,c} = 1$ and $\omega_c = 0$, therefore the motion of the generator is quasi Lagrangian and we fulfill the material frame indifference requirement. For other cases, ω_c smoothly varies between 0 and 1. Once the new generator position G_c^{n+1} is computed one constructs the corresponding Voronoi mesh using the Voronoi machinery. This mesh needs a last treatment as this Voronoi mesh may have arbitrary small edges. Such edges can drastically and artificially reduce the time step, and, more important can lead to a lack of robustness. Even if in theory such faces could be kept, we prefer to remove/clean them, see [1].

4.2. Remap phase by exact-intersection

The remapping phase is a conservative interpolation of physical variables from the Lagrangian polygonal mesh at the end of the Lagrangian step onto the new polygonal mesh after the rezone step. The remapping phase must provide valid physical variables to the Lagrangian scheme, and conservation of mass, momentum and total energy must be ensured. Moreover at least a second-order accuracy remapping has to be constructed. In ReALE the rezoned mesh may have a different connectivity than the Lagrangian mesh. Therefore the remapping phase of ReALE is based on exact-intersection of *a priori* two different polygonal meshes. Primary variables are cell-centered density, velocity and specific total energy. Conservative quantities are cell-centered mass, momentum and total energy. First piecewise linear representations of cell-centered variables ρ_c, $\rho_c\boldsymbol{U}_c$, $\rho_c E_c$ are constructed on the Lagrangian mesh. Then a slope limiting process [12] is performed to enforce physically justified bounds. This phase does not change compared to the Cartesian geometry. Then conservative quantities, namely mass, momentum and total energy, are obtained by integration of these representations in cylindrical geometry. Moreover for multi-species computation each mass fraction is remapped.

4.2.1. Control volume based remap

In control volume formulation, volume integrations are performed using the true cylindrical volume, $V = \int_\Omega \mathscr{R}\, d\boldsymbol{A}$, that is to say the measure of the volume obtained by rotation of the surface Ω about the Z-axis. New conservative quantities are calculated by integration over polygons of intersection of new (rezoned) and old (Lagrangian) meshes. Let us consider one non empty polygon resulting from the intersection between an old cell Ω_d^{old} and a new cell Ω_d^{new} namely $\Omega_{cd} = \Omega_c^{\text{old}} \bigcap \Omega_d^{\text{new}}$. Then the mass embedded into this polygon is obtained by integration over Ω_{cd} of the piecewise linear limited representation of cell-centered density $\rho_c(\boldsymbol{X})$

$$\Delta m_{cd} = \int_{\Omega_{cd}} \rho_c(\mathrm{X})\mathscr{R}\, d\boldsymbol{A}. \tag{22}$$

Due to the linear representation of $\rho_c(\boldsymbol{X})$, the previous equation exhibits the integrals of $\mathscr{R}$, $\mathscr{R}^2$, $\mathscr{R}Z$ which are reduced by the Green theorem to the boundary integrals, and subsequently evaluated from the coordinates of the Ω_{cd} region vertices. As a consequence the mass in any new cell $\Omega_d^{\text{new}} = \bigcup_c \Omega_{cd}$ is simply obtained by summation

$$m_d^{\text{new}} = \int_{\Omega_d^{\text{new}}} \rho(\mathrm{X})\mathscr{R}\, d\boldsymbol{A} = \int_{\bigcup_c \Omega_{cd}} \rho(\mathrm{X})\mathscr{R}\, d\boldsymbol{A} = \sum_{c\setminus\Omega_{cd}\neq\emptyset} \Delta m_{cd}. \tag{23}$$

Momentum and total energy are calculated likewise. Finally, primary variables in Ω_d are recovered by division by new volume V_d^{new} (for density) or new mass m_d^{new} (for momentum and energy).

To use the multi-species EOS, we need to remap the concentrations of the F fluids from the Lagrangian grid onto the rezoned one. To this end, we first compute the mass of fluid f in the Lagrangian cell Ω_c^{old}: $m_{f,c} = \int_{\Omega_c^{\text{old}}} \rho C_f \mathscr{R}\, d\boldsymbol{A}$. We note that $m_c^{\text{old}} = \sum_{f=1}^F m_{f,c}$ since $\sum_{f=1}^F C_{f,c} = 1$. Then, the mass of each fluid is conservatively interpolated onto the rezoned grid following the methodology previously described. We denote its new value by $m_{f,c}^{\text{new}}$. At this point we notice that $m_c^{\text{new}} \neq \sum_{f=1}^F m_{f,c}^{\text{new}}$, this discrepancy comes from the fact that our second-order remapping does not preserve linearity due to the slope limiting. Hence, we define the new concentrations $C_{f,c}^{\text{new}} = \frac{m_{f,c}^{\text{new}}}{m_c^{\text{new}}}$ and impose the renormalization $C_{f,c}^{\text{new}} \leftarrow \frac{C_{f,c}^{\text{new}}}{\sum_{f=1}^F C_{f,c}^{\text{new}}}$ so that $\sum_{f=1}^F C_{f,c}^{\text{new}} = 1$. We point out that this renormalization does not affect the global mass conservation.

4.2.2. Area-weighted based remap

The difference between control volume and area-weighted formulation lays in the form of the momentum equation. As previously mentioned Eq. (7) has the same form as in Cartesian geometry modulo the presence of the Cartesian inertia $\mu_c = \overline{m}_c\overline{\mathscr{R}}_c$. Consequently the remapping of the momentum equation in area-weighted cylindrical geometry is performed as in Cartesian geometry. Then the momentum embedded into $\Omega_{cd} = \Omega_c^{\text{old}} \bigcap \Omega_d^{\text{new}}$ is obtained by integration of the piecewise linear limited representation of cell-centered momentum $(\rho\boldsymbol{U})_c(\boldsymbol{X})$

$$\Delta \mathrm{W}_{cd} = \int_{\Omega_{cd}} (\rho\mathrm{U})_c(\mathrm{X})\mathscr{R}\, d\boldsymbol{A} = \overline{\mathscr{R}}_c \int_{\Omega_{cd}} (\rho\mathrm{U})_c(\mathrm{X}) d\mathrm{X} d\mathrm{Y},$$

where the integration is performed over the Cartesian volume $d\mathrm{X}d\mathrm{Y}$. The momentum in a new cell Ω_d^{new} is given by

$$\mathrm{W}_d^{\text{new}} = \int_{\Omega_d^{\text{new}}} (\rho\mathrm{U})_c(\mathrm{X})\mathscr{R}\, d\boldsymbol{A} = \int_{\bigcup_c \Omega_{cd}} (\rho\mathrm{U})_c(\mathrm{X})\mathscr{R}\, d\boldsymbol{A} = \sum_{c\setminus\Omega_{cd}\neq\emptyset} \Delta \mathrm{W}_{cd}.$$

The new velocity in cell Ω_d^{new} is finally given by

$$\mathrm{U}_d^{\text{new}} = \frac{\mathrm{W}_d^{\text{new}}}{\mu_d} = \frac{\mathrm{W}_d^{\text{new}}}{m_d^{\text{new}}\overline{\mathscr{R}}_d^{\text{new}}},$$

where m_d^{new} has been remapped using the true cylindrical volume and $\overline{\mathscr{R}}_d^{\text{new}}$ has been recomputed on the new area A_d^{new}.

5. Numerical tests

In this section we present the numerical results obtained by the cylindrical cell-centered ReALE code based on CHIC code, [3]. Let us remind that any vector is written in the (Z,R) space and that multi-species test cases are run with concentration equations. The first test is the well-known Sedov test case; it is used as a sanity check as no physical vorticity is expected to occur and therefore reconnection-based methods are not required. The second is a helium bubble shock interaction in cylindrical geometry, it is run in order to show the predictive capabilities of ReALE technique. This test generates vorticity which is a classical cause of failure for Lagrangian schemes. For a fixed-connectivity ALE code, it usually leads to a conflict between the Lagrangian motion with a tendency to tangle the mesh and, the mesh-regularization motion with a tendency to avoid bad quality cells. Such a conflict produces a stagnation of the mesh that reconnection technique is intended to cure [1]. Experimental results of the shock/bubble interaction are compared to the simulations. The last test problem is the rise of a light bubble under gravity for which the same type of vortex motion is expected. As no mesh symmetry is supposed to be preserved, we run the code in its control volume formulation for the last two test cases. Only the Sedov problem is run in area-weighted formulation to show the ability of the code to handle this formulation.

5.1. Sedov problem

Let us consider the Sedov blast wave problem with spherical symmetry. This problem models an intense explosion in a perfect gas with a diverging shock wave. The computational domain is $\Omega = [0,1.2] \times [0,1.2]$. The initial conditions are characterized by $(\rho_0, P_0, \boldsymbol{U}_0) = (1, 10^{-6}, \boldsymbol{0})$ for a perfect gas with polytropic index set to $\gamma = \frac{5}{3}$. We set an initial delta-function energy source at the origin prescribing the pressure in the cell containing the origin as $P_{ori} = (\gamma - 1)\rho_0 \frac{\mathscr{E}_0}{V_{ori}}$, where V_{ori} denotes the volume of the cell and $\mathscr{E}_0$ is the total amount of released energy. Choosing $\mathscr{E}_0 = 0.425536$, the solution consists of a diverging shock whose front is located at radius $R = 1$ at time $t = 1$. The peak density reaches the value 6. Symmetry boundary conditions are applied on the axis. The initial mesh is a degenerate Voronoi mesh obtained from 50×50 uniformly distributed generators and four more generators on the corners of the domain. This test does not need ALE, and *a fortiori* ReALE, technique; pure Lagrangian schemes usually perform well. However this is used to assess the validity of ReALE approach. We present the density and mesh in Fig. 4 left-panel. Moreover density is presented as a function of the cell radius for any cell against the exact solution (straight line) in the right-panel of Fig. 4. The final Lagrangian mesh presents expanded cells in the rarefaction wave and compressed ones after the shock wave. On this sanity check ReALE technique in cylindrical geometry is able to produce a smooth mesh and accurate results.

5.2. Helium bubble shock interaction

The computational domain is $\Omega = [0;0.65] \times [0;0.089]$ which represents a cylinder of diameter 0.178 and initial length 0.65. The spherical helium bubble is represented as a disk defined by the coordinates of its center $(Z_c, R_c) = (0.320, 0)$ and its radius $R_b = 0.0225$ (see Fig. 5). We prescribe wall boundary conditions at each boundary except at $Z = 0.65$, where we impose a piston-like boundary condition defined by the inward velocity $\boldsymbol{V}^\star = (u^\star, 0)$. The incident shock wave is defined by its Mach number, $M_s = 1.25$. The bubble and the air are initially at rest. The initial data for helium are $(\rho_1, P_1) = (0.182, 10^5)$, its molar mass is $\mathscr{M}_1 = 5.269 \times 10^{-3}$ and its polytropic index is $\gamma_1 = 1.648$. The initial

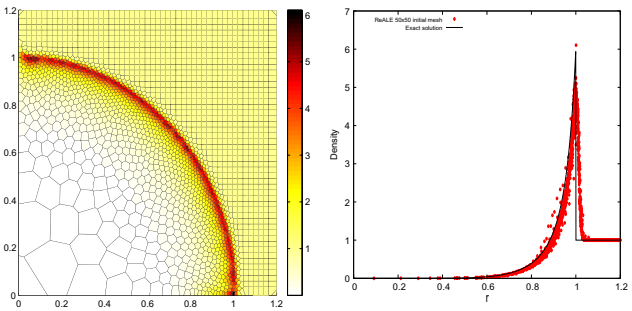


Fig. 4. Sedov problem in cylindrical geometry at time $t = 1.0$ for ReALE – left: mesh and density – right: density as a function of radius for all cells *vs* the exact solution (line).

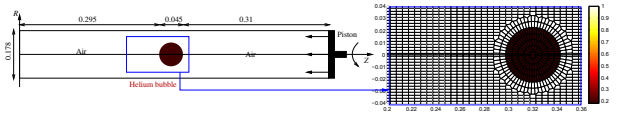


Fig. 5. Helium bubble shock interaction. A right piston compresses some air at rest by sending a $M = 1.25$ shock wave that passes through an helium bubble in cylindrical geometry – left: initial bubble, the computational domain is $\Omega = [0;0.65] \times [0;0.089]$ that is reflected against the Z-axis. Right: zoom on the initial mesh and density that are reflected against the z-axis.

data for air are $(\rho_2, P_2) = (1, 10^5)$, its molar mass is $\mathscr{M}_2 = 28.963 \times 10^{-3}$ and its polytropic index is $\gamma_2 = 1.4$. Specific internal energies are $\varepsilon_1 = 8.4792 \times 10^5$ and $\varepsilon_2 = 2.5 \times 10^5$. Using the Rankine–Hugoniot relations, we find that the velocity of the piston is given by $u^\star = -140.312$. The incident shock velocity is $D_s = -467.707$. The incident shock wave hits the bubble at time $t_i = 657.463 \times 10^{-6}$. The stopping time for our computation is $t_{end} = t_i + 1594 \times 10^{-6} = 2251.463 \times 10^{-6}$. The mesh is built with a set of 4856 generators designed to produce a Voronoi mesh that has a mesh line which exactly matches the bubble boundary, see Fig. 5. In all figures the top part is the actual computational domain that is mirrored around the Z-axis for visualization purposes. In Fig. 6 are displayed density and mesh around the bubble for five intermediate times and the final time of the simulation, namely $t_a = t_i + 20 \times 10^{-6}$, $t_c = t_i + 145 \times 10^{-6}$, $t_d = t_i + 223 \times 10^{-6}$, $t_f = t_i + 600 \times 10^{-6}$ and $t_g = t_i + 1594 \times 10^{-6}$. These correspond to five shadow-photographs of experimental results from [13] (Fig. 8 of page 53) that we reproduced in Fig. 6 (right-panels). Let us note that the final time has a different color scale and that the visualization window follows the bubble. We observe a quite good agreement with the experimental results even for this coarse mesh; the timing of the shock waves and the shape of the deformed bubble fit the shadow-graphs of the experimental results. Of great importance is the fact that the bubble detaches from the Z-axis at t_f and more clearly at t_g, this can be also guessed from the experimental shadow-graphs. Finally in Fig. 7 are displayed the density waves present in the full domain at intermediate times $t_b = t_i + 82 \times 10^{-6}$, $t_c = t_i + 145 \times 10^{-6}$, $t_d = t_i + 223 \times 10^{-6}$, and $t_e = t_i + 1007 \times 10^{-6}$. The dark zones are the inside bubble and the air that has not been yet attained by the initial shock wave. Multiple reflections and refractions can be observed in the density wave patterns.

5.3. Rise of a light bubble under gravity

This problem consists in the rise of a light bubble in a heavy gas bubble under gravity [14]. The statement of the problem is sketched in Fig. 8. The computational domain is $\Omega = [-15,20] \times [0,15]$. This domain is split into three regions filled with air (ideal gas EOS with $\gamma = 1.4$) at rest. One defines for each point (Z,R) its radius $\mathrm{R} = \sqrt{R^2 + Z^2}$ and angle θ so that $Z(\boldsymbol{R},\theta) = \boldsymbol{R}\cos\theta$. The initial data are given in Zone I (inside the bubble), Zone II (transition layer) and Zone III (exterior) by

Zone I	Zone II	Zone III
$\boldsymbol{R} \leqslant R_1$	$R_1 < \boldsymbol{R} < R_2$	$\boldsymbol{R} \geqslant R_2$
$\varepsilon_1 = 3 \times 10^3$	$\varepsilon_2(\boldsymbol{R})$	$\varepsilon_3 = 15.6$
$p_1 = 0.6$	$p_2(\boldsymbol{R},\theta)$	$p_3(\boldsymbol{R},\theta) = 0.6e^{-Z(\boldsymbol{R},\theta)/\Lambda}$

$R_1 = 6.6$ is the radius of the light bubble, $R_2 = 8.5$ is the radius of the transition layer towards the atmosphere, $\Lambda = 63.7$ is the inhomogeneity parameter for the atmosphere. In Zone II a linear transition is applied between the values of p and ε of Zones I and III

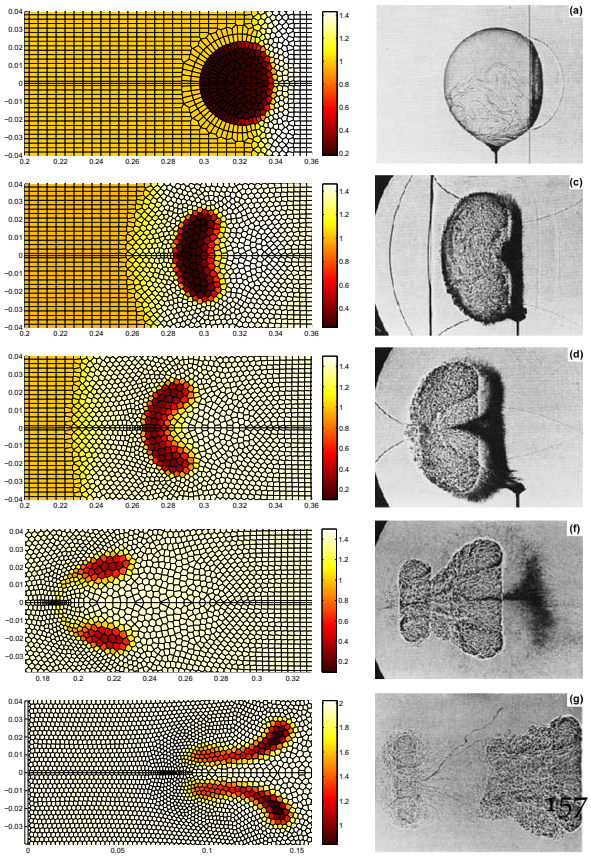


Fig. 6. $M = 1.25$ shock interaction with a spherical helium bubble – mesh and density – from top to bottom: $t_a = t_i + 20 \times 10^{-6}$, $t_c = t_i + 145 \times 10^{-6}$, $t_d = t_i + 223 \times 10^{-6}$, $t_f = t_i + 600 \times 10^{-6}$ and $t_g = t_i + 1594 \times 10^{-6}$ where $t_i = 657.463 \times 10^{-6}$ is the time of the shock/bubble interaction. Left: ReALE in cylindrical geometry. Right: Experimental results from [13].

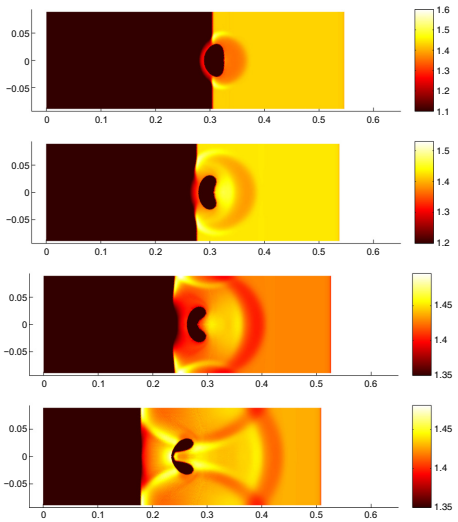


Fig. 7. $M = 1.25$ shock interaction with a spherical helium bubble — density waves in the domain — from top to bottom: $t_a = t_i + 82 \times 10^{-6}$, $t_c = t_i + 145 \times 10^{-6}$, $t_d = t_i + 223 \times 10^{-6}$, $t_e = t_i + 1007 \times 10^{-6}$ where $t_i = 657.463 \times 10^{-6}$ is the time of the shock/bubble interaction. ReALE in cylindrical geometry.

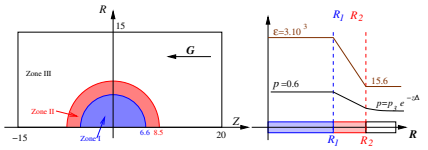


Fig. 8. Rise of a light bubble under gravity. The light bubble (Zone I) has a radius of R_1 and a transition layer is initialized between R_1 and R_2 (Zone II). The rest of the domain $\boldsymbol{R} > R_2$ is some air at rest (Zone III) where $R = \sqrt{R^2 + Z^2}$. Gravity is oriented in the Z direction. The pressure and internal energy profiles are sketched in the right panel.

$$p_2(R, \boldsymbol{\theta}) = (1 - \alpha(R))p_1 + \alpha(R)p_3(R, \boldsymbol{\theta}),$$
$$\varepsilon_2(R) = (1 - \alpha(R))\varepsilon_1 + \alpha(R)\varepsilon_3,$$

with $\alpha(\boldsymbol{R}) = (\boldsymbol{R} - R_1)/(R_2 - R_1)$. Gravity is set downward the Z direction with magnitude $G = 9.8 \times 10^{-2}$ see Fig. 8.The bubble rises in the Z direction because of the density gradients and velocity. In its motion it further deforms into a classical mushroom shape. The final time is $t_{final} = 14$. The initial mesh is made of a total of 1901 cells split into roughly 1200 quadrangles outside Zone II and 653 polygonal cells inside refer to Fig. 9 top-middle panel. Walls boundary conditions are assumed everywhere besides for the Z-axis where symmetry boundary condition is applied. In Fig. 9 left column are plotted the density and mesh for the time moments $t_0 = 0$, $t_1 = 1$, $t_2 = 8$ and $t_3 = 14$. As for the bubble/shock problem the visualization is performed after reflection against the Z-axis. The colored vorticity and vector velocity fields are

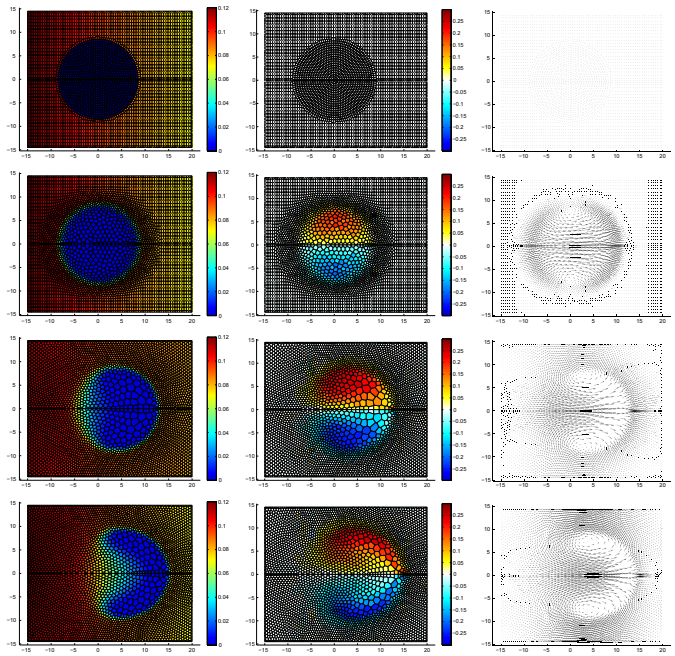


Fig. 9. Rise of a light bubble under gravity — ReALE in cylindrical geometry — left column: density and mesh — middle column: vorticity and mesh — right column: veolcity vectors — time $t_0 = 0$, $t_1 = 1$, $t_2 = 8$, $t_3 = 14$.

shown in Fig. 9 middle and right columns respectively. As expected the bubble rises upwards the Z direction. It adopts a mushroom shape as can be seen in Fig. 9 at $t_3 = 14$. Time $t_1 = 1$ shows how the ReALE technology starts to adapt the mesh while waves are emanating from the bubble. At time $t_2 = 8$ the bubble starts to deform while the cells ahead the tip of the bubble are highly compressed; this process is pursued up to final time. Vorticity and velocity vector plots confirm that the fluid undergoes a vortex-like motion. Such a vortex-like motion has a natural tendency to highly stretch cells in classical ALE simulation without reconnection leading to inaccuracy or failure of the simulation. Contrarily ReALE is able to undergo such motion while allowing cells to change neighborhood. These ReALE results are in agreement with the numerical results provided in [14]. In this paper the authors use a different numerical method that leads to a non smooth polygonal mesh as shown in Fig. 4.19 of page 111. ReALE technique form this point of view seems superior as our mesh keeps a general good geometrical quality.

6. Conclusion and perspectives

In this paper we investigate the extension to cylindrical geometry of the recently developed Reconnection Arbitrary-Lagrangian–Eulerian (ReALE) technology [1]. This extension is fairly obvious; indeed, the cell-centered Lagrangian scheme was already available in cylindrical geometry using a control volume or area-weighted formulation. Moreover, the rezone technology using Voronoi machinery with moving generators introduced in [1] can be used likewise. The last part of our ReALE code, namely the remapping part, is more demanding as it must utilize a control volume based exact-intersection of *a priori* two different polygonal meshes

provided by the Lagrangian and rezone phases. In the control volume formulation true volume integrals are used to remap mass, momentum and total energy whereas in area-weighted formulation the momentum is remapped as in Cartesian geometry. Multi-fluid is treated with concentration equations that must be remapped likewise.

We show that the extension of ReALE to cylindrical geometry produces good results on numerical test cases. First we run the Sedov problem as a sanity check. Then we simulate an helium bubble shock interaction problem. We compare our multi-species simulation against experimental shadow-graphs proving the validity and accuracy of the ReALE technology in cylindrical geometry. The last problem is the rise of a light bubble under gravity that presents vortex-like motion. Unlike ReALE, a classical fixed-connectivity ALE code usually presents difficulties to capture such a motion.

In the near future we plan to investigate the association of ReALE with interface reconstruction in planar and cylindrical geometries. Moreover we will investigate the possible extension of ReALE to 3D.

Acknowledgments

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. The authors thank A. Solovjov for allowing to use his code for Voronoi mesh generation. The authors thank M. Kucharik, J. Dukowicz, F. Adessio, H. Trease, G. Ball, A. Barlow, P. Váchal, V. Ganzha, B. Wendroff, J. Campbell, D. Burton V. Tishkin, A. Favorskii, V. Rasskazova, N. Ardelyan, S. Sokolov for stimulating discussions over many years. The first author acknowledges the help of S. Galera and J. Breil in using CHIC ReALE code.

References

[1] Loubère R, Maire P-H, Shashkov M, Breil J, Galera S. ReALE: a reconnection-based arbitrary-Lagrangian–Eulerian method. JComput Phys 2010;229:4724–61.
[2] Du Q, Faber V, Gunzburger M. Centroidal Voronoi tesselations: applications and algorithms. SIAM Review 1999;41:637–76.
[3] Maire P-H. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. J Comput Phys 2009;228:2391–425.
[4] Maire P-H. A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry. J Comput Phys 2009;228:6882–915.
[5] Caramana EJ, Burton DE, Shashkov MJ, Whalen PP. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. J Comput Phys 1998;146:227–62.
[6] Adessio FL, Carroll DE, Dukowicz KK, Johnson JN, Kashiwa BA, Maltrud ME, et al. CAVEAT: a computer code for fluid dynamics problems with large distortion and internal slip. Technical report LA-10613-MS, Los Alamos National Laboratory; 1986.
[7] Maire P-H. A unified sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grid. Int J Numer Meth Fluids 2010, doi:10.1002/fld.2328.
[8] Maire P-H. Contribution to the numerical modeling of inertial confinement fusion, private communication.
[9] Whalen P. Algebraic limitations on two dimensionnal hydrodynamics simulations. J Comput Phys 1996;124:46–54.
[10] Dukowicz JK. A general non-iterative Riemann solver for Godunov's method. J Comput Phys 1985;61:119–37.
[11] Du Q, Emelianenko M, Gunzburger M. Convergence of the Lloyd algorithm for computing centroidal Voronoi tesselations. SIAM J Numer Anal 2006;44:102–19.
[12] Barth TJ. Numerical methods for gasdynamic systems on unstructured meshes. In: Kroner D, Ohlberger M, Rohde C, editors. An introduction to recent developments in theory and numerics for conservation laws. Proceedings of the international school on theory and numerics for conservation laws, lecture notes in computational science and engineering. Berlin: Springer; 1997. p. 195–284.
[13] Haas J-F, Sturtevant B. Interaction of weak-shock waves. J Fluid Mech 1987;181:41–76.
[14] Sofronov I, Rasskazova V, Nesterenko L. The use of nonregular nets for solving two-dimensional non-stationary problems in gas dynamics. In: Yanenko N, Shokin Y, editors. Numerical methods in fluid dynamics. Moscow: MIR; 1984. p. 82–121.

2.4 ALE codes comparison

In 2007-2008 with Richard Liska and Pavel Váchal (from the Czech connection in Prague) and J. Breil, S. Galera and P.-H. Maire (CELIA, university of Bordeaux) we have investigated the behaviors of our three ALE codes. At this time the codes' description was

CHIC code uses the cell-centered Lagrangian method, condition number rezoning and cell-centered swept region remapping.

PALE standing for Prague ALE. A staggered Lagrangian compatible Lagrangian scheme with Winslow rezoner for this study [8] and partial subcell swept region remapper.

ALE INC(ubator) code uses the staggered Lagrangian method, the Reference Jacobian rezoning and a full subcell swept region remapping.

The goal was to run these codes in their nominal configuration [9] and compare their general behaviors.

As an illustration we reproduce here the Sedov blast wave problem [169] in Cartesian 2D geometry. In the paper we have also run sanity checks (1D Sod shock tube) and shock/bubble interaction test cases. In Fig.2.17 are presented the results of the three ALE codes and an Eulerian scheme based on Lax and Liu scheme [175],[176] for the density variable for the final time $t_{final} = 1$. The top/middle rows correspond to a 30×30 initial Cartesian mesh, the bottom row corresponds to a 60×60 mesh. The top row presents the meshes colored by density. We plot in middle and bottom rows the cell density (for all cells) as a function of the cell radius *versus* the exact solution which consists of a shock wave located at $r = 1$ at $t_{final} = 1$ and an exponential type of density function before the diverging shock (travelling then towards to the right of the pictures) the peak of which is at $\rho_{max} = 6$. The three Lagrangian codes produce decent results, the cell-centered being the most accurate one (see the maximum peak as instance). The two staggered Lagrangian codes perform alike and the Eulerian scheme is the least performing of the four methods. Notice, that all Lagrangian methods are performing for this test much better than the Eulerian method as the moving Lagrangian mesh concentrates more cells close to the circular shock.

The paper [20] is reproduced in the following pages.

8. PALE also enjoys Reference Jacobian rezoning, condition number smoothing.
9. In other words not trying to tweak all fixes and parameters that may help.

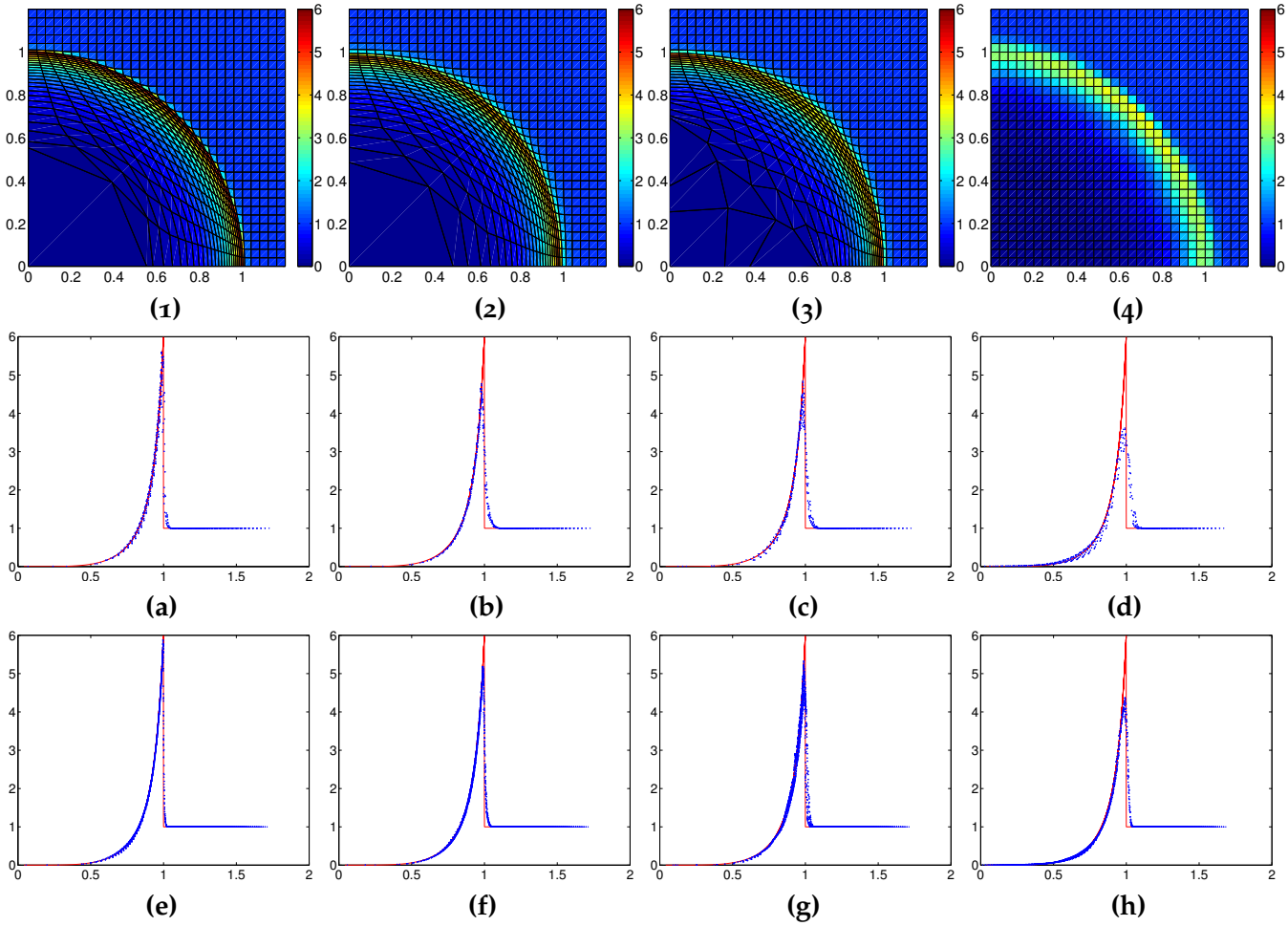


Figure 2.17 – *Sedov problem — Density at* $t = 1$ — **(a)** *: CHIC* 30×30 *cells* — **(b)** *: ALE INC* 30×30 *cells* — **(c)** *: PALE* 30×30 *cells* — **(d)** *: Eulerian scheme* 30×30 *cells* — **(e)** *: CHIC* 60×60 *cells* — **(f)** *: ALE INC* 60×60 *cells* — **(g)** *: PALE* 60×60 *cells* — **(h)** *: Eulerian scheme* 60×60 *cells.*

Proceedings of Symposia in Applied Mathematics

Comparison of Staggered and Cell-Centered Lagrangian and ALE Hydrodynamical Methods

Richard Liska, Raphaël Loubère, Pierre-Henri Maire, Jérôme Breil, Stéphane Galera, and Pavel Váchal

Abstract. Three Arbitrary Lagrangian-Eulerian codes are compared on a set of test problems. Code CHIC is based on cell-centered Lagrangian scheme while codes ALE INC and PALE are based on staggered Lagrangian scheme. Pure Lagrangian methods are first tested on Sod and Sedov problems. The full ALE methods are then compared on two more advanced problems, namely triple point and shock bubble interaction problems, which cannot be treated by pure Lagrangian method due to severe distortion of moving Lagrangian mesh.

1. Introduction

Lagrangian framework for Euler equations, describing dynamics of compressible fluid, offers several advantages compared to standard Eulerian one. In Lagrangian case the computational mesh is moving together with the simulated fluid with zero mass flux between computational cells while Eulerian method employs fixed static computational mesh with advection flux between cells. Clearly Lagrangian methods are superior for problems which involve changing computational domain with moving boundary conditions for which Eulerian setup would be very difficult. For some problems, e.g. those involving shear flow or vorticity, moving Lagrangian mesh suffers from mesh distortion which can lead to failure of pure Lagrangian method. The problems with mesh distortion can however be treated by the ALE (Arbitrary Lagrangian Eulerian) method [**HAC74**] which, either regularly or when needed, smooths the distorted mesh and remaps the conserved quantities to the new smoothed mesh.

1991 *Mathematics Subject Classification.* Primary 35L65, 65M06; Secondary 76M20.

Key words and phrases. Arbitrary-Lagrangian-Eulerian method, moving mesh method, conservative interpolation, remapping, rezoning, Lagrangian hydrodynamics, staggered scheme, cell-centered scheme.

This work has been supported in part by EGIDE through the Project-Hubert-Curien Barrande number 13908PL and by the Czech Ministry of Education projects MEB 020730 and MSM 6840770022.

©0000 (copyright holder)

2. Numerical methods

The Euler equations for compressible gas dynamics written in Lagrangian coordinates are

$$\begin{aligned} \rho\frac{d\,\tau}{d\,t} + \operatorname{div}\mathbf{u} &= 0, \\ \rho\frac{d\,\mathbf{u}}{d\,t} + \mathbf{grad}\,p &= 0, \\ \rho\frac{d\,e}{d\,t} + \operatorname{div}(p\,\mathbf{u}) &= 0, \end{aligned} \tag{2.1}$$

where ρ is density, $\tau = 1/\rho$ is specific volume, $\mathbf{u}$ is velocity, p is pressure, $e = \epsilon + \mathbf{u}^2/2$ is specific total energy and ϵ is specific internal energy. The system of Euler equations which express the conservation of mass, momentum and energy is closed by the equation of state $p = p(\rho, \epsilon)$.

The traditional standard Lagrangian methods, such as compatible method [**CBSW98**], use the staggered discretization with scalar quantities (density, internal energy, pressure) defined inside the computational cells and vector quantities (velocity and position) defined at the mesh nodes. The compatible method is based on forces acting on the node which replace the pressure gradient in the momentum equation. The internal energy is then updated by the work done by the forces which is directly derived from total energy conservation. The subzonal pressure forces [**CS98**] are used to prevent hourglass type of cell motion. The staggered method adds the necessary dissipation at compression regions by means of artificial viscosity [**CSW98, CS01**] applied also in the form of forces.

Recently several cell-centered Lagrangian methods with all quantities defined in the cells have been proposed [**DM05, MABO07, MB08**]. These methods are finite volume schemes for which the pressures and velocities on mesh edges (which are needed in the numerical fluxes) are obtained from Riemann solver at nodes of the mesh. The obtained nodal velocities are used to move the mesh. The dissipation for cell-centered schemes is being added through the Riemann solver and the entropy inequality is satisfied.

This paper employs an indirect ALE method which consists of three phases. The first phase is one or several Lagrangian steps described above. Rezoning, which is the second phase, performs computational mesh smoothing and untangling to improve the quality of the Lagrangian mesh while trying to keep the smoothed mesh close to the Lagrangian one. In the third phase the conservative quantities are remapped from the Lagrangian mesh to the smoothed mesh.

The simplest rezoning method is Winslow smoothing [**Win63**]. A better rezoning method is condition number smoothing [**Knu00**]. Even more advanced is the Reference Jacobian method [**KMS02**].

The remapping phase performs conservative interpolation of conservative quantities from the old Lagrangian mesh to the new rezoned mesh. The conservative quantities are first linearly reconstructed with limited slopes on the cells of the old mesh. This piecewise linear reconstruction is then integrated over intersections of the old cells with the new cell to get integral average over the new cell. As the new smoothed mesh and old mesh do not differ much, this exact integration can be replaced by approximate integration over the regions swept by the cell edges moving from the old mesh to the new one [**KSW03**].

Here we compare the performance of three Lagrangian and ALE codes: CHIC code developed at CELIA laboratory at Bordeaux university, ALE INC(ubator) code developed at Los Alamos National Laboratory and maintained at Toulouse university and PALE (Prague ALE) developed at Czech Technical University in Prague. The CHIC code uses the cell-centered Lagrangian method, condition number rezoning and cell-centered swept region remapping. The ALE INC code uses the staggered Lagrangian method, the Reference Jacobian rezoning and a full subcell swept region remapping [**LS05**]. The PALE code uses the staggered Lagrangian method, Winslow rezoning and a partial subcell swept region remapping.

3. Numerical results

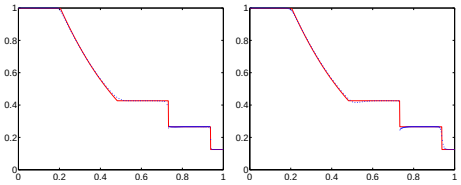
For the comparison we have selected several test cases including Sod shock tube problem, Sedov blast wave problem, triple point problem and shock bubble interaction problem.

The first two tests, Sod and Sedov problems, have known exact solution and can be computed by pure Lagrangian method as the moving Lagrangian mesh does not degenerate. On these problems we compare the Lagrangian methods.

The exact solutions for the other two test problems are not known. These tests cannot be handled by pure Lagrangian methods as the moving mesh degenerates (some computational cells become non-convex or even inverted). The latter two test problems can be formulated in Eulerian framework and we obtain their reference numerical solutions by high order Eulerian numerical method on very fine computational mesh. The results of the staggered and cell-centered Lagrangian methods computed on rather coarse meshes are compared with reference Eulerian solutions.

3.1. Sod problem. The classical Sod problem is a 1D Riemann shock tube. Its solution consists of a left moving rarefaction, a contact discontinuity and a right moving shock. The discontinuity is initially located at 0.5. The domain is filled with an ideal gas at rest with $\gamma = 1.4$. The density/pressure values on the left side of the discontinuity are 1.0/1.0, while those on the right side are 0.125/0.1. We simulate this problem on domain $\Omega = [0;1]\times[0;y_{\max}]$ (where $y_{\max}$ is chosen so that the cells are initially squares) using 200×1 cells up to the final time $t_{\text{final}} = 0.25$.

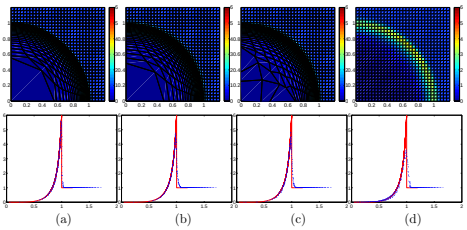
Results for the Sod problem computed by pure Lagrangian methods are presented in Fig. 1 together with exact solution of this Riemann problem plotted by solid line. Fig. 1(a) shows the result by cell-centered CHIC and Fig. 1(b) results by staggered ALE INC and PALE, which are the same. The shock wave on the right is better resolved by the cell-centered method. Its resolution by staggered method can be improved by adjusting the artificial viscosity parameters, however here in all tests we use the standard parameters without any adjustments. The contact discontinuity in the middle is resolved better by the staggered methods with jump in one cell, while cell-centered result is slightly diffused on the top left of the contact, which is typical for Godunov-like methods. On the other hand staggered method produces worse undershoot on the right of the contact. At the tail of the rarefaction wave on the left the staggered method produces an undershoot which is typical for this method, while the cell-centered method diffuses the tail of the rarefaction. At the head of the rarefaction both methods diffuse the jump in gradient, with cell-centered method producing slightly better resolution.



(a) (b)

Figure 1. Density for Sod problem at $t = 0.25$ for 200 cells – (a) CHIC; (b) ALE INC and PALE give the same result. Exact solution by solid line.

3.2. Sedov problem. Sedov problem [**Sed59**] describes the evolution of a blast wave in a point-symmetric explosion. The total energy of the explosion is concentrated in one cell c at the origin and has magnitude $E_{\text{total}} = 0.244816$. The specific internal energy of this cell is defined as $\epsilon(c) = E_{\text{total}}/V(c)$. Initial density is equal to 1, pressure is 10^{-6} (except in cell c at the origin), $\gamma = 1.4$ and gas is initially at rest. We solve this problem on region $\Omega = [0;1.2]\times[0;1.2]$ with 30×30 and 60×60 cells. At final time $t_{\text{final}} = 1.0$ the exact solution is a cylindrically symmetric diverging shock whose front is at radius $r = \sqrt{x^2+y^2} = 1$ and has a density peak $\rho = 6.0$.



(a) (b) (c) (d)

Figure 2. Density for Sedov problem at $t = 1$ for 30×30 cells – (a) CHIC; (b) ALE INC; (c) PALE; (d) Euler LL. Bottom row is the scatter plot of density in each cell versus radius (its distance from origin) with exact solution by solid line.

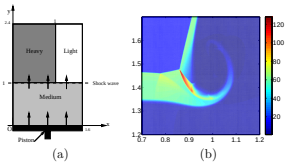
Mesh	30 × 30	60 × 60
CHIC	0.14	0.07
ALE INC	0.15	0.08
PALE	0.16	0.11
Euler LL	0.44	0.23

TABLE 1. Convergence table of relative $L1$ errors for Sedov problem.

We compute this problem not only by our three Lagrangian/ALE codes, but for comparison also by Eulerian Lax and Liu (LL) positive scheme [**LL98**] at the same resolution on the static computational mesh. The results are presented in Fig. 2, where the upper row shows density colormaps with computational mesh and the lower row shows the scatter plots of density versus radius $r = \sqrt{x^2 + y^2}$ in each cell together with the exact solution of the problem by [**KT07**]. As the exact solution is available, we can compute the errors of the numerical solution for objective comparison. The relative $L1$ errors $e_{L1} = \int |\rho_n - \rho_e| \mathrm{d}V / \int |\rho_e| \mathrm{d}V$ (ρ_n is numerical density and ρ_e the exact density) of both meshes are compared in Tab. 1. The table shows the best results by CHIC, followed by ALE INC and PALE. The Eulerian methods are much worse than all the Lagrangian ones which is clearly visible also in Fig. 2. This demonstrates another advantage of Lagrangian methods which for some problems are able to concentrate more cells in interesting areas, here around the diverging shock wave.

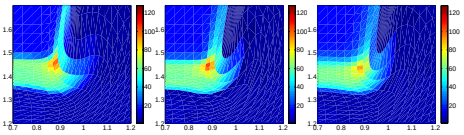
3.3. Triple point problem. The triple point is the point where three contact discontinuities meet. A planar shock wave generated by moving piston impacts the triple point and creates a vortex of heavy material into the lighter one.

The setup of the problem is in Fig. 3(a). The solution domain $\Omega = [0; 1.6] \times [0; 2.4]$ is divided into three regions: the medium region $[0; 1.6] \times [0; 1]$ on the bottom with $(\rho_m, p_m, \mathbf{u}_m) = (4, 4/3, (0, 1))$, the heavy region $[0; 1] \times [1; 2.4]$ on the top left with $(\rho_h, p_h, \mathbf{u}_h) = (16, 10^{-6}, (0, 0))$ and the light region $[1; 1.6] \times [1; 2.4]$ on the top right with $(\rho_l, p_l, \mathbf{u}_l) = (1, 10^{-6}, (0, 0))$. The gas constant is $\gamma = 5/3$ everywhere. The problem is simulated until final time $t_{\text{final}} = 0.8$ on two meshes with 40×60 and 80×120 cells.



(a) (b)

FIGURE 3. Triple point problem – (a) initial setup; (b) density of reference Euler LL solution with 1000×1000 cells at $t = 0.8$.



(a) (b) (c)

FIGURE 4. Density for triple point problem at $t = 0.8$ for 40×60 cells – (a) CHIC; (b) ALE INC; (c) PALE.

Mesh	40×60	80×120
CHIC	0.14	0.08
ALE INC	0.14	0.10
PALE	0.16	0.10

TABLE 2. Convergence tables of relative $L1$ density deviations for triple point problem.

For this problem the exact solution is not known, so we compute the reference solution by the Eulerian positive LL scheme on the fine mesh with 1000×1000 cells on the domain $[0.2; 1.6] \times [0.8; 2.1]$. Density of the reference solution is presented in Fig. 3(b) and is used to compute the relative $L1$ deviations of numerical solution from the reference solution in Tab. 2. From the table we can conclude, that the best results are achieved by CHIC and the worst by PALE, which can be visually confirmed also from density colormaps presented in Fig. 4 (compare the maximum density value in the middle and the ridge going left-up from this maximum). In Fig. 4 one can also distinguish different meshes of the three codes which are related both to different Lagrangian methods and different rezoning methods used by our codes. This problem cannot be solved by pure Lagrangian method due to severe mesh distortion.

3.4. Shock bubble interaction problem. In the shock bubble interaction problem a piston-generated planar shock wave passes through a square-shaped bubble of low density and generates a diverging lens shaped deformation of the bubble.

The initial setup is presented in Fig. 5. The domain $\Omega = [-1; 1] \times [0; 0.5]$ is filled with air $(\rho_a, p_a, \mathbf{u}_a = (1, 1, (0, 0))$. A squared bubble $[0.25; 0.35] \times [0; 0.1]$ within this tube is filled with low density gas at rest with $(\rho_b, p_b, \mathbf{u}_b = (0.1, 1, (0, 0))$. A piston at the left end of the tube moving with velocity $\mathbf{u}_p = (2, 0)$ sends a planar shock wave into the tube which interacts with the bubble. The problem is simulated until final time $t_{\text{final}} = 0.5$ on two meshes with 100×25 and 200×50 cells.

For this problem the exact solution is not known, so we compute the reference solution by the Eulerian positive WENO scheme [**JS96**] on the fine mesh with 1100×250 cells on the domain $[-1.2; 1] \times [0; 0.5]$. The density and the vorticity curl $\mathbf{u}$ of the reference solution are presented in Fig. 7. The reference solution is

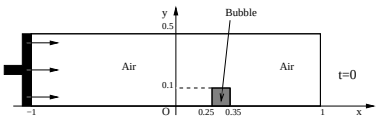
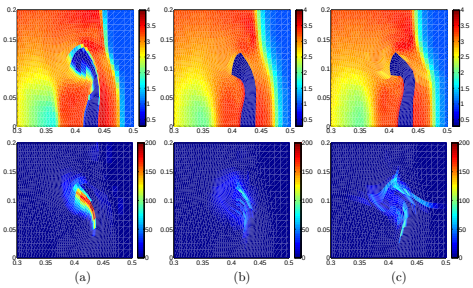


FIGURE 5. Initial setup for shock bubble interaction problem.



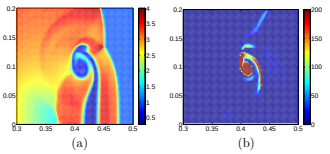
(a) (b) (c)

FIGURE 6. Density in upper row and vorticity in lower row for shock bubble interaction problem at $t = 0.5$ for 200×50 cells – (a) CHIC; (b) ALE INC; (c) PALE.

Mesh	100x25	200x50
CHIC	0.036	0.021
ALE INC	0.062	0.036
PALE	0.061	0.035

TABLE 3. Convergence tables of relative $L1$ density deviations for shock bubble interaction problem.

used to compute the relative $L1$ deviations of numerical solution from the reference solution in Tab. 3. From the table we can conclude, that the best results are achieved by CHIC. The deviations of results by ALE INC and PALE are very close. The numerical results of the three ALE codes are presented in Fig. 6 by density colormap in the upper row and vorticity colormap in the lower row. The contact discontinuity at the bubble surface is resolved best by ALE INC and PALE which



(a) (b)

FIGURE 7. Reference Euler WENO solution with 1100×250 cells at $t = 0.5$ for shock bubble interaction problem – (a) density; (b) vorticity.

employ the staggered Lagrangian method. The nice, really jump-like resolution of contacts is a typical feature of staggered methods (see also contact resolution for the Sod problem in Fig. 1). The contact is diffused in Eulerian reference solution in Fig. 7(a), which is typical for Eulerian methods as their fluxes include convective terms transporting mass, momentum and energy between computational cells. The surface of the upper part of bubble by cell-centered CHIC is diffused, however as mentioned above its overall precision is better. This problem cannot be solved by pure Lagrangian method due to severe mesh distortion.

It is well known that Lagrangian methods have trouble with resolution of vortexes as they distort the Lagrangian moving mesh. This is why we present vorticity results. The maximum vorticity for the Eulerian reference solution shown in Fig. 7(b) is 600, while our colormap ends at 200, which is the maximum vorticity of the CHIC results. Maximum vorticity for ALE INC and PALE is even much lower. Cell-centered CHIC is able to compute results with vorticity, while staggered ALE INC and PALE seems for kill almost all the vorticity in the solution. Killing of vorticity by the staggered method is certainly influenced by the employed artificial viscosity which tends to damp vorticity [**CL06**]. Vorticity damping might be also influenced by the rezone/remap step of the ALE method.

4. Conclusion

We have compared three ALE codes on a set of test problems. First they have been compared in the pure Lagrangian mode on Sod and Sedov problems for which exact solutions exist. Then we have compared them on two more tests involving triple point and shock bubble interaction problems which cannot be treated by pure Lagrangian method and the full ALE methodology has to be employed. We were pleasantly surprised by very good performance of the recently developed cell-centered Lagrangian method based code CHIC, whose results are often superior to results of traditional staggered methods being used for many years.

Acknowledgments

The authors would like to thank Milan Kuchařík for developing most parts of the PALE code and for fruitful discussions.

References

[CBSW98] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen, *The construction of compatible hydrodynamics algorithms utilizing conservation of total energy*, J. Comput. Phys. **146** (1998), no. 1, 227–262.

[CL06] E. J. Caramana and R. Loubère, *"curl-q": A vorticity damping artificial viscosity for Lagrangian hydrodynamics calculations*, J. Comput. Phys. **215** (2006), no. 2, 385–391.

[CS98] E. J. Caramana and M. J. Shashkov, *Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures*, J. Comput. Phys. **142** (1998), no. 2, 521–561.

[CS01] J. C. Campbell and M. J. Shashkov, *A tensor artificial viscosity using a mimetic finite difference algorithm*, J. Comput. Phys. **172** (2001), no. 2, 739–765.

[CSW98] E. J. Caramana, M. J. Shashkov, and P. P. Whalen, *Formulations of artificial viscosity for multi-dimensional shock wave computations*, J. Comput. Phys. **144** (1998), no. 1, 70–97.

[DM05] B. Despres and C. Mazeran, *Lagrangian gas dynamics in two dimensions and Lagrangian systems*, Arch. Rational Mech. Anal. **178** (2005), no. 3, 327–372.

[HAC74] C. W. Hirt, A. A. Amsden, and J. L. Cook, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, J. Comput. Phys. **14** (1974), no. 3, 227–253.

[JS96] G. S. Jiang and C. W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys. **126** (1996), no. 1, 202–228.

[KMS02] P. Knupp, L. G. Margolin, and M. Shashkov, *Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods*, J. Comput. Phys. **176** (2002), no. 1, 93–128.

[Knu00] P. Knupp, *Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I – a framework for surface mesh optimization*, Int. J. Numer. Meth. Engrg. **48** (2000), no. 3, 401–420.

[KSW03] M. Kuchařík, M. Shashkov, and B. Wendroff, *An efficient linearity-and-bound-preserving remapping method*, J. Comput. Phys. **188** (2003), no. 2, 462–471.

[KT07] J. R. Kamm and F. X. Timmes, *On efficient generation of numerically robust Sedov solutions*, Tech. Report LA-UR-07-2849, Los Alamos National Laboratory, 2007.

[LL98] P. D. Lax and X. D. Liu, *Solution of two dimensional Riemann problem of gas dynamics by positive schemes*, SIAM J. Sci. Comput. **19** (1998), no. 2, 319–340.

[LS05] R. Loubere and M. Shashkov, *A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods*, J. Comput. Phys. **209** (2005), no. 1, 105–138.

[MABO07] P-H. Maire, R. Abgrall, J. Breil, and J. Ovadia, *A cell-centered Lagrangian scheme for two-dimensional compressible flow problems*, SIAM J. Sci. Comput. **29** (2007), no. 4, 1781–1824.

[MB08] P-H. Maire and J. Breil, *A second order cell-centered Lagrangian scheme for two-dimensional compressible flow problems*, Int. J. Numer. Meth. Fluids **56** (2008), no. 8, 1417–1423.

[Sed59] L. I. Sedov, *Similarity and dimensional methods in mechanics*, Academic Press, New York, 1959.

[Win63] A. M. Winslow, *Equipotential zoning of two-dimensional meshes*, Tech. Report UCRL-7312, Lawrence Livermore National Laboratory, 1963.

Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, 115 19 Prague 1, Czech Republic
E-mail address: liska@siduri.fjfi.cvut.cz

Institut de Mathématiques de Toulouse (IMT), Université Paul-Sabatier, Toulouse, France
E-mail address: raphael.loubere@math.univ-toulouse.fr

Centre d'Etudes Laser Intenses et Applications (UMR CELIA), Université Bordeaux I et CEA/CESTA, France
E-mail address: maire@celia.u-bordeaux1.fr

Centre d'Etudes Laser Intenses et Applications (UMR CELIA), Université Bordeaux I et CEA/CESTA, France
E-mail address: breil@celia.u-bordeaux1.fr

Centre d'Etudes Laser Intenses et Applications (UMR CELIA), Université Bordeaux I et CEA/CESTA, France
E-mail address: galera@celia.u-bordeaux1.fr

Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, 115 19 Prague 1, Czech Republic
E-mail address: vachal@galileo.fjfi.cvut.cz

2.5 Multi-material treatment

Dealing with more than one material requires several new developements in a staggered ALE code.

The staggered Lagrangian scheme should be extended to treat mixed cells. Several techniques are well documented, see as instance [177]. One simple technique is based on concentration equations. The multi-material flow is considered as a multi-component mixture of miscible fluids. Each fluid is characterized by its concentration being a passive scalar used to define the location of an interface within mixed cells. The equation of state of the mixture of materials must be determined usually by an *ad hoc* assumption like pressure/temperature equilibrium. Another approach based on volume of fluid (VOF) [178] method which introduces a Lagrangian tracking of material interfaces is often prefered when no mixing between materials is expected. Consequently mixed cells are present in the computational domain. These consists of the collocation of two or more materials within the same cell. The time evolution of a mixed cell is obtained using a closure model that computes a mixed thermodynamic state function of the thermodynamic states of each material taking into account the volume fractions (i.e. the rates of presence). Several closure models have been developed as instance pressure/temperature equilibrium or relaxation [179, 180, 181, 182, 183], subcell dynamics formulations [184] and other comparable techniques [185, 186, 187, 188]. As already seen each mixed cell must be split into pure materials knowing the volume fractions fields. Generally a reconstruction of the interface between two materials in mixed cell is performed using a straight segment following the well-known Youngs' method [189, 190] (or also known after [191] as PLIC *Piecewise Linear Interface Calculation* the extension of SLIC method *Simple Line Interface Calculation* [192]).

Some requirements could also be added to the rezone method, as instance some interfaces between materials can be preserved during the rezone phase. However most of the time the rezoning phase is kept as it is.

The remapping phase should adapt to the mixing model employed in the Lagrangian phase. In the case of concentration equations each concentration must be consistently remapped as instance by using some scaling as in [177]. On the other hand in the case of a VOF method remapping is made material by material and usually demands the use of an exact intersection between the rezoned mesh and the Lagrangian mesh with reconstructed interfaces in mixed cells.

Due to their strict conservation of materials, volume-of-fluid (VOF) methods using interface reconstruction are widely used. However the effective management and capture of interfaces is essential for accurate and reliable simulation of multi-material and multi-phase flows. As already said VOF methods do not explicitly track the interface between materials, but rather advect volume fractions which prescribe the material composition of each cell of the mesh. When the interface between materials is needed, the interface is recreated based on the material volume fraction in the cell and its surrounding cells [52, 191, 193, 194]. A common problem impacting these reconstruction methods is their dependence on a specified material ordering, i.e. if more than two materials are present in a cell, the reconstruction may depend on the sequence in which the materials are processed. As an illustration we present in Fig. 2.18 the nested dissections obtained by Youngs' method in one mixed cell with three materials : two different orders of treatment produce different final interfaces (see panels (d) and (h)). This is undesirable as it may improperly locate materials within the cell. Moreover this may result in material being incorrectly fluxed into neighbouring cells.
In 2005 with some colleagues from the Los Alamos National Laboratory we have put some effort to develop an order-independent interface reconstruction method for general grids.

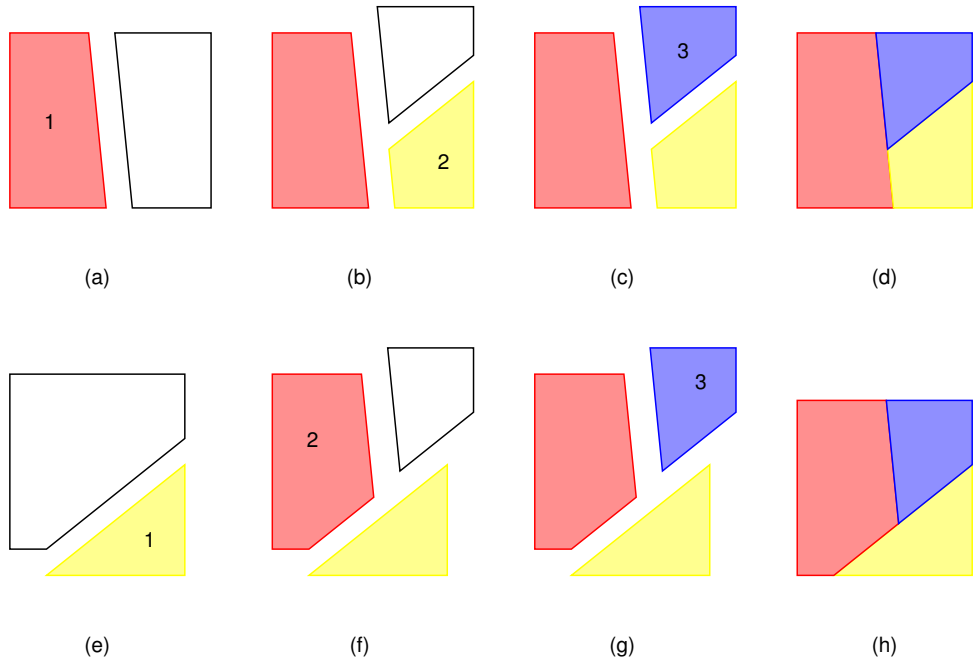


FIGURE 2.18 – *Nested dissection interface reconstruction for three materials (a) the first material is removed leaving a smaller available polygon, (b) the second material is removed from the available polygon, (c) the remaining available polygon is assigned to material 3, (d) the resulting partitioning of the computational cell. (e)-(g) show the same procedure but the materials are processed in a different order leading to a different reconstruction (h).*

2.5.1 Interface reconstruction techniques using Power Diagram

In a set of two articles [13, 17] respectively entitled *Material order independent interface reconstruction using power diagrams* and *A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations* with my colleagues from Los Alamos S.P. Schofield, R.V. Garimella and M.M. Francois we have developed a method that can reconstruct a multi-material interface with no dependence on material ordering. The method is very general : it works on unstructured grids, accommodates an arbitrary number of materials and extends naturally to three dimensions. The method does not assume a topology for the material regions, i.e. a layer structure or triple point configuration. Furthermore, all of the material regions created are convex. Notice that at the same time V. Dyadechko with M. Shashkov from Los Alamos have developed an alternative and concurrent technique called Moment of Fluid method (MOF) [195]. This technique has been tested in 3D [196] and in [197] a comparative study between different methods including the power diagram method from [13, 17] and MOF has been carried out.

The method proposed in [13] consists of two steps :

First, the relative location of the materials within the mixed cell is approximated ;

Second, a power diagram is used to split the mixed cell into pure polygons the surfaces of which fulfill the volume fractions of the materials.

Relative location of the materials within the mixed cell. The method utilizes a particle attraction model or approximate centroid calculation to infer the relative location of the materials in the cell. In the first step of the method, a number of particles representing the materials are placed in multi-

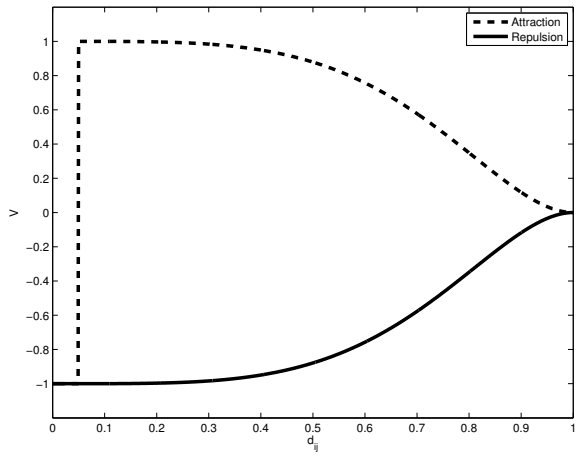


Figure 2.19 – *Particle attraction and repulsion "forces" used in the model.*

material cells and any pure or mixed neighboring cells. A particle, P_i, has a position, $\mathbf{x}_i$, velocity $\mathbf{V}_i = \frac{d\mathbf{x}_i}{dt}$ and material $m(i)$, and is constrained to stay within the cell in which it is initially placed. Taking inspiration from molecular dynamics [198, 199] and smoothed particle hydrodynamics [200, 201], we evolve the particle positions according to "forces" based on the particles' relative locations and materials. The positions of the particles are updated through time integration of a set of ordinary differential equations,

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{V}_i \tag{2.8}$$

$$\mathbf{V}_i = \sum_{j:\, m(j)=m(i)} \mathbf{V}_{\mathbf{att}}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{j:\, m(j)\neq m(i)} \mathbf{V}_{\mathbf{rep}}(\mathbf{x}_i, \mathbf{x}_j)$$

where $\mathbf{V}_{\mathbf{att}}$ and $\mathbf{V}_{\mathbf{rep}}$ are the prescribed attractive and repulsive "forces" in the direction $\mathbf{x}_j - \mathbf{x}_i$. Particles of the same material attract each other until they are very close, at which point they start to repel each other. Particles of different materials repel each other. In our tests, the particles start at random locations within their cell, but they can be initialized using other means such as their relative locations in a cell at a previous time step.
The particle-particle "forces" (plotted in Fig. 2.19) are prescribed as

$$\mathbf{V}_{\mathbf{att}}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} -1, & d_{ij} < \delta \\ 1 - 2d_{ij}^4 + d_{ij}^8, & \delta \leq d_{ij} \leq 1 \\ 0, & d_{ij} > 1 \end{cases} \tag{2.9}$$

$$\mathbf{V}_{\mathbf{rep}}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} -(1 - 2d_{ij}^4 + d_{ij}^8), & d_{ij} \leq 1 \\ 0, & d_{ij} > 1 \end{cases} \tag{2.10}$$

$$\tag{2.11}$$

where $d_{ij} = \frac{||\mathbf{x}_i - \mathbf{x}_j||}{2.5h}$ is the distance between points scaled by an interaction distance, taken to be 2.5 times the characteristic mesh size h, and $\delta = 0.05$. Unlike a traditional mechanical model, the "forces" here actually prescribe the instantaneous velocities of the particles. In a cell, $\mathcal{C}_i$, the number of particles, $N(\mathcal{C}_i)$ is

$$N(\mathcal{C}_i) = \left\lfloor N_p \times \frac{\|\mathcal{C}_i\|}{A_0} \right\rfloor \tag{2.12}$$

where N_p is a prescribed constant (usually around 30), $\|\mathcal{C}_i\|$ is the area of the cell, A_0 is a reference cell area for the grid (for example on a uniform Cartesian grid, $A_0 = h^2$ where h is the grid spacing) and $\lfloor a \rfloor$ is the floor function giving the greatest integer less than or equal to a. Each particle has a designated material type, corresponding to a material present in the cell. Each material that is present in the cell is represented by the same number of particles, $N(\mathcal{C}_i)/N_m^i$, where N_m^i is the number of materials present in the cell. We found that making the number of particles representing each material proportional to the volume fraction of the material often leads to unsatisfactory results. If the volume fraction is small, the material will be represented by only a few particles, which are not sufficient to provide a reliable estimate of the location of the material within the cell. In addition, we found that for unstructured, general polygonal grids, making the number of particles proportional to the area of the cell was important. Otherwise, the particles tend to cluster in regions of the mesh with a concentration of smaller cells.

Once the particles are distributed, the particle model is run. Since the model prescribes instantaneous velocities and not true forces, the particles may remain in perpetual motion unless the system is forced to cool. The velocities determine the kinetic energy of the system which in turn defines the temperature. The velocity of each particle is rescaled at each time step to decrease the kinetic energy of the system and force the particles to settle into a final configuration. At time step n in the time integration of Equation 2.8, the kinetic energy of all the particles is

$$KE^n = \sum_i \frac{1}{2} ||\mathbf{V}_i||^2 \tag{2.13}$$

After the system is sufficiently agitated, typically after 5 to 10 time steps, we force the kinetic energy to decrease as,

$$KE^{n+1} \leq \alpha KE^n \tag{2.14}$$

where $0 < \alpha < 1$. In practice, α is set to be $0.7 - 0.9$. If $KE^{n+1} \geq KE^n$, all the particle velocities are scaled as

$$\mathbf{V}_i' = \sqrt{\alpha \frac{KE^n}{KE^{n+1}}} \mathbf{V}_i. \tag{2.15}$$

To speed up the calculation, we use a variable time step with a new Δt calculated after each time step as

$$\Delta t = \frac{0.1}{2||\mathbf{V}_{max}||}, \tag{2.16}$$

where $||\mathbf{V}_{max}|| = \max_i ||\mathbf{V}_i||$ where $\mathbf{V}_i$ is as defined in Equation 2.8. The positions are then updated as

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \ \mathbf{V}_i'. \tag{2.17}$$

If a particle goes outside the cell, it is placed back in the cell by repositioning it to the center of the triangle formed by the old position, the new position, and the center of the cell. If that fails, the particle is kept in its old position.

The particles are allowed to evolve for a number of time steps until the average kinetic energy of each particle has dropped below a specified stopping criteria. The particle model exhibits rapid convergence to the particle clusters, usually requiring under 20 time steps to converge to approximately the final positions. We have conducted statistical tests that show the model displays little sensitivity to the random initial particle positions, with standard deviations in the final material locations typically less than 5 percent of the mesh spacing [202].

This particle technique is able to produce the results in Fig. 2.20 where in a mixed cell, for each material randomly initialised particles (the number of which depend on the volume fraction) approximately gather around the material centroid. This particle method can produce approximate centroids in 2D and 3D.

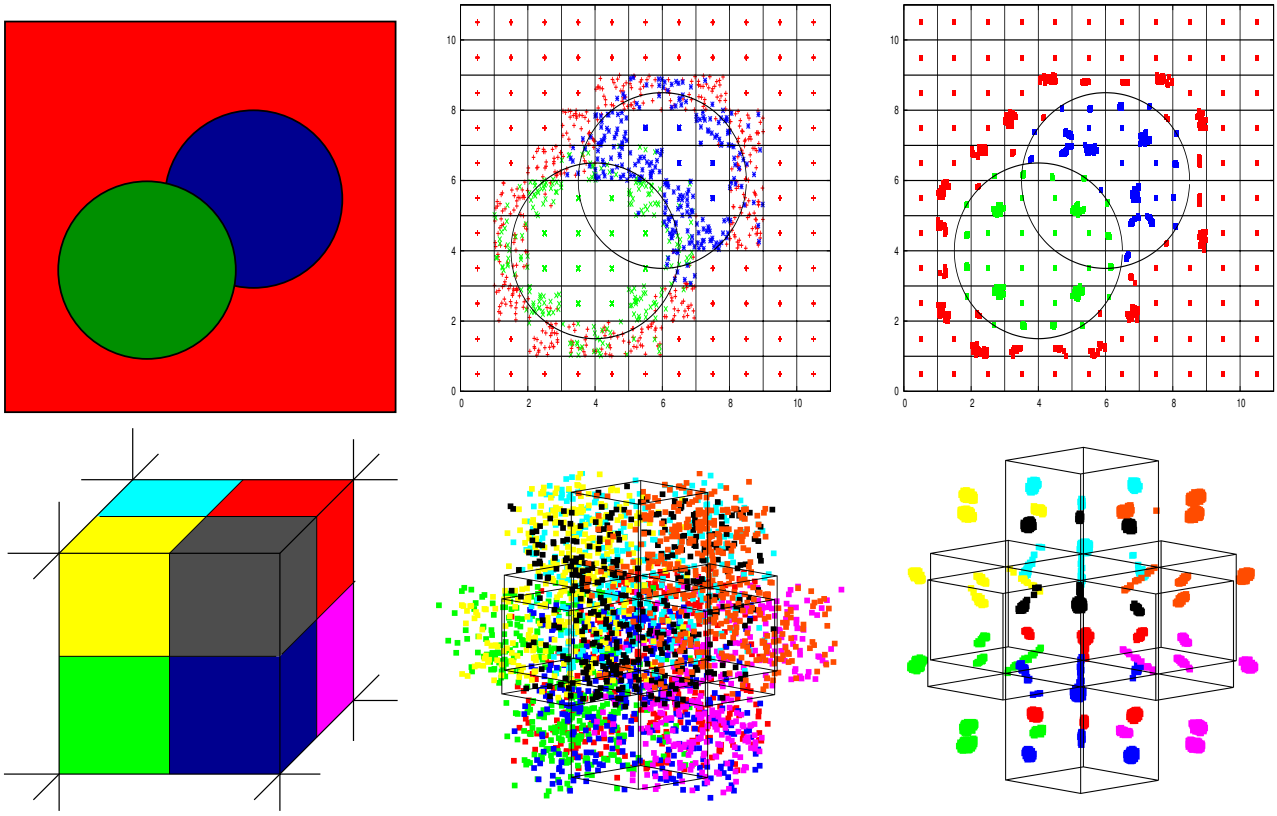


Figure 2.20 – *Numerical results obtained with the particle attraction model to infer the relative location of the materials in mixed cells. Top panels : two enlaced disks (left panel) are described by particles which are randomly distributed with mixed cells (middle). At convergence of the partcle model (right panel) particles do represent an accurate approximation of the material centroids — Bottom panels : approximation for 3D eight materials case (zoom on the central 8 materials mixed cell on the left panel), initial distribution of particles in the neighborhood of the central mixed cell (middle panel) and final location of particles in the enighborhood (right panel).*

An alternative technique in [13] is the direct calculation of an approximate center of mass of each material in a subset of the mesh around the cell being reconstructed.

Given the volume fractions of materials in cells in a mesh, our task is to determine the relative locations of materials in a multi-material cell. To do this, we must ideally recover the characteristic function for each material in the domain. While it is possible to reconstruct the characteristic function in 1D [203], no method (other than interface reconstruction itself) exists to do this in higher dimensions. Therefore, we make a simplifying assumption that a smooth function, called the volume fraction function, exists for each material and that its pointwise cell-centered values are given by cell-wise volume fraction data. This smooth function represents the distribution of material in the mesh cells and in that sense, it can be considered analogous to a density distribution function for the material. However, we should note that the volume fraction function is not a clearly defined mesh-independent continuous function like the density function. Swartz [204] describes it as the

function that quantifies the relative amount of a material present in a small window that moves around in a domain with a sharp interface. Defined this way, it is clear that the volume fraction function steepens as the size of the window (or in other words, the mesh size) gets smaller and the gradient of the function blows up as the window size goes to zero. Nevertheless, for a given mesh, we will treat the volume fraction function like a smooth, density distribution function.

We then compute a piecewise linear approximation for this smooth volume fraction function using standard methods used in higher-order finite-volume methods [205]. Finally, continuing the analogy with the density function, we compute the center of mass of the materials in cells from the linear reconstruction as described below.

Consider a mesh on which we have cell-centered values f_i of a function $f(\mathbf{x})$. In each cell $\mathcal{C}_i$, we reconstruct a linear approximation, $\tilde{f}_i(\mathbf{x})$, of the function such that

$$\tilde{f}_i(\mathbf{x}) = f_i + \nabla f \cdot (\mathbf{x} - \mathbf{x}_c(\mathcal{C}_i)), \tag{2.18}$$

where $\mathbf{x}_c(\mathcal{C}_i)$ is the centroid of the cell. ∇f is the gradient of the function that we wish to approximate and it is considered to be constant within the cell. The gradient may be computed either by a Green-Gauss [205] or a least-squares technique [191]. On structured and unstructured grids, we use all vertex and edge connected neighbors in the gradient computation. For a least-squares technique, the same mesh cells are used in the computation with each entry weighted by the inverse of the squared distance between the centroid of the cell being reconstructed and the centroid of the neighboring cell as described in [191]. The computed gradient is limited using Barth-Jesperson-type limiter [206] to preserve local bounds on the volume fraction function. The limiter is calculated using all vertex connected neighbors. The limited gradient is indicated by $\boldsymbol{\delta} = \phi \nabla f$ with $\phi \in (0, 1]$. Then, the approximate center of mass of the function $f(\mathbf{x})$ over the domain Ω_i as approximated by the function $\tilde{f}_i(\mathbf{x})$ is given by

$$\bar{\mathbf{x}} = \frac{\int_{\Omega_i} \mathbf{x} \tilde{f}_i(\mathbf{x}) d\Omega}{\int_{\Omega_i} \tilde{f}_i(\mathbf{x}) d\Omega} = \frac{1}{\|\Omega_i\| f_i} \int_{\Omega_i} \mathbf{x}(f_i + \boldsymbol{\delta} \cdot (\mathbf{x} - \mathbf{x}_c(\mathcal{C}_i))) d\Omega \tag{2.19}$$

where $\|\Omega_i\|$ is the area of the domain Ω_i.

The obvious choice of domain Ω_i for integrating this equation is the cell, $\mathcal{C}_i$ and this works well for structured meshes. The calculation of equation 2.19 for a polygon may be done with the application of Stokes' theorem in the plane, for details see [207]. However, for unstructured meshes, we have found that integrating over the cell domain induces a strong bias in the orientation of the reconstructed interface based on the cell geometry. In order to eliminate this effect, we integrate instead over the smallest square, $S(\mathcal{C}_i) \supseteq \mathcal{C}_i$, whose center coincides with the centroid of the cell, $\mathbf{x}_c(\mathcal{C}_i)$ and encloses the computational cell.

For two materials, this choice of integration domain is equivalent to a gradient-based method when using a power diagram interface reconstruction. In a power diagram based reconstruction of a two material cell, the interface normal depends only on the direction of the vector pointing from one material locator to the other. For two materials, m and n, with material locators $\mathbf{x}_m$ and $\mathbf{x}_n$ and volume fractions f_m and $1 - f_m$ respectively, the normal to the interface between them given by the power diagram reconstruction will be

$$\begin{aligned} \mathbf{x}_m - \mathbf{x}_n &= \frac{1}{\|S(\mathcal{C}_i)\|} \left(\frac{1}{\bar{f}_m} + \frac{1}{1 - \bar{f}_m} \right) \int_{y_0}^{y_1} \int_{x_0}^{x_1} \mathbf{x} \left(\boldsymbol{\delta} \cdot (\mathbf{x} - \mathbf{x_c}) \right) \, dx \, dy \\ &= \frac{\Delta^2}{12} \left(\frac{1}{\bar{f}_m (1 - \bar{f}_m)} \right) \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} \end{aligned}$$

where $S(\Omega_i) = [x_0, x_1] \times [y_0, y_1]$ and $\Delta = x_1 - x_0 = y_1 - y_0 = \sqrt{||S(\mathcal{C}_i)||}$. That is, the normal is a positive constant times the gradient. Hence, the interface normal will be the computed gradient. In addition, this choice of integration domain makes the calculation of equation 2.19 trivial and provides a better initial reconstruction for a starting point to the interface smoothing procedures.

Power diagram Using the relative location of materials obtained from the particle method or from a piecewise linear reconstruction of the volume fraction function, the interface is reconstructed using a weighted Voronoi diagram, known as a power diagram, such that the required volume fractions are exactly matched [208, 209].
In [13] we propose several static test cases to show the behavior of this new method. We reproduce in the following Fig. 2.21 the static four material interface reconstruction on structured and unstructured meshes using (a) particles and power diagrams, (b) approximate centroids and power diagrams, and in (c), (d) Youngs' method [190, 189] with two different material orderings. Notice that Youngs' method can not represent quadruple point, therefore no material ordering will provide an acceptable and accurate result.

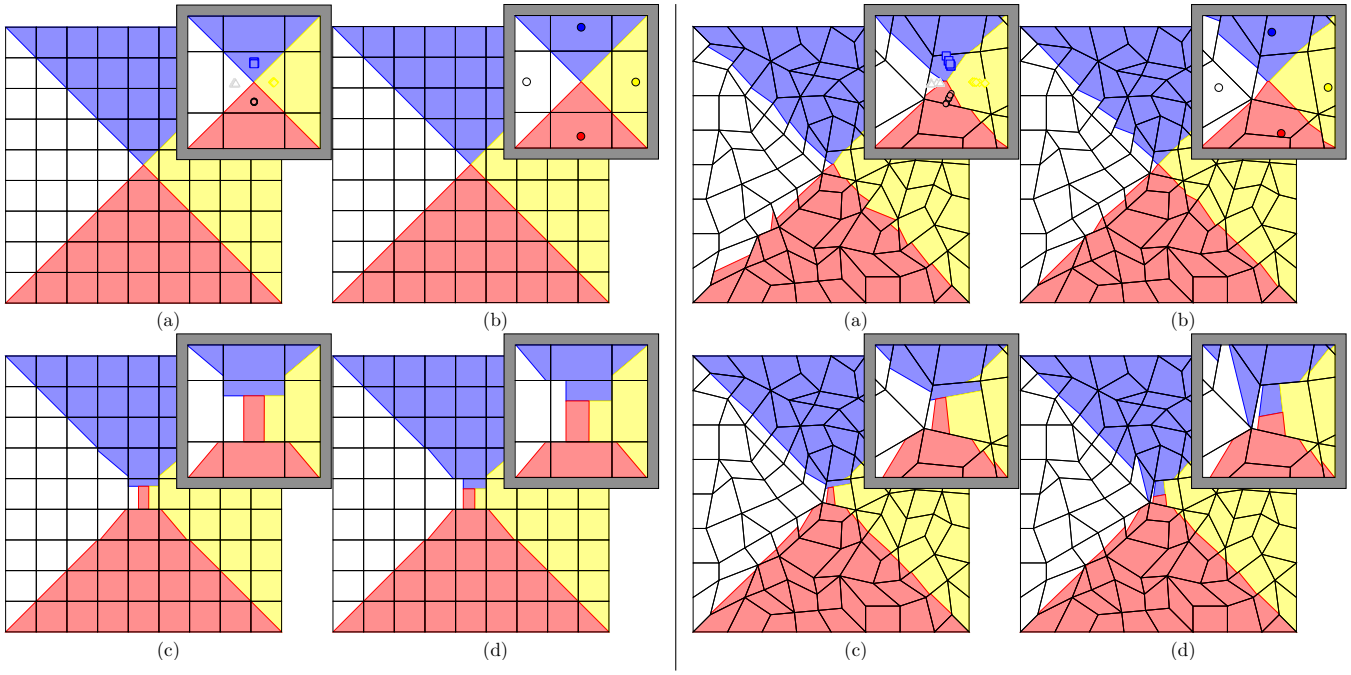
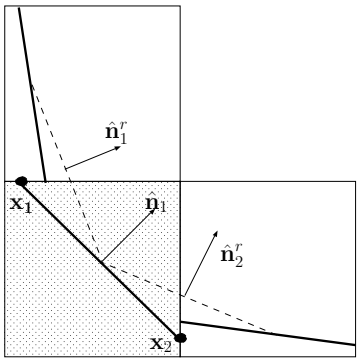
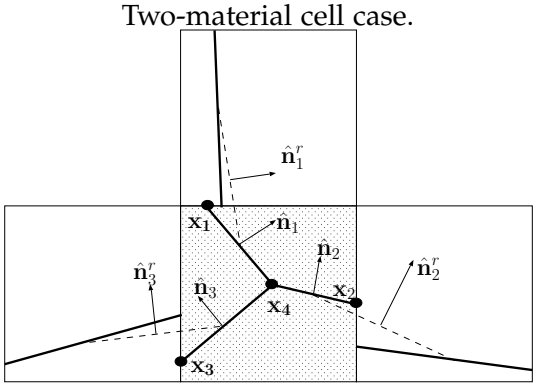


Figure 2.21 – *Numerical results from paper [13]. Four material interface reconstruction on structured and unstructured meshes using (a) particles and power diagrams (b) approximate centroids and power diagrams (c), (d) Youngs' method with two different material orderings. The insets show the four material cell at the center of the mesh. The converged particles locations for the center cell are also shown in the inset in (a). The approximate centers of mass for the center cell are shown in (b).*

Neither the particle model nor the approximate center of mass method when combined with a power diagram-based reconstruction exactly reproduces a straight line, indicating that both methods are only first-order accurate in this sense. Consequently in [17] the interfaces are improved by minimizing an objective function that smoothes interface normals while enforcing convexity and volume constraints for the pure material subcells.



Two-material cell case.



Three material cell case.

Consider a 2D cell with N_m materials, N_s interface segments and N_p interface points. The smoothing procedure repositions the cell's N_p interface points so that it minimizes the discrepancy between the normal of each of its interface segments and normals of reference interface segments in neighboring cells (separating the same materials). The constraints imposed on this process are that the volume fractions of the materials in the cells must be matched exactly and that all the pure material subcells remain convex. Naturally, interface points on the boundary of the cell must remain on the boundary and interior points must remain strictly inside the cell. The local objective function for smoothing in a particular cell is written as :

$$F_i(\mathbf{s}) = \sum_{j=1}^{N_s} \sum_{k=1}^{(N_r)_j} \|\hat{\mathbf{n}}_j(\mathbf{s}) - \hat{\mathbf{n}}_k^r\|^2.$$

As an illustration the figure to the left presents the definition of reference normals.

In order to numerically validate this approach we have proposed in [17] a set of test cases : the diagonal translation of a four material disk and a four material vortex test in an incompressible reversible velocity field. Convergence tests from the article show the second-order accuracy of the proposed method.
As instance one presents in Fig. 2.22 the translation of a four material disk In this figures one compares our second-order approach with Youngs' method using two different prescribed material orders. The four material vortex test case's results are presented in Fig. 2.23 where our first-order method is also presented.

The paper [13] is reproduced in the following pages.

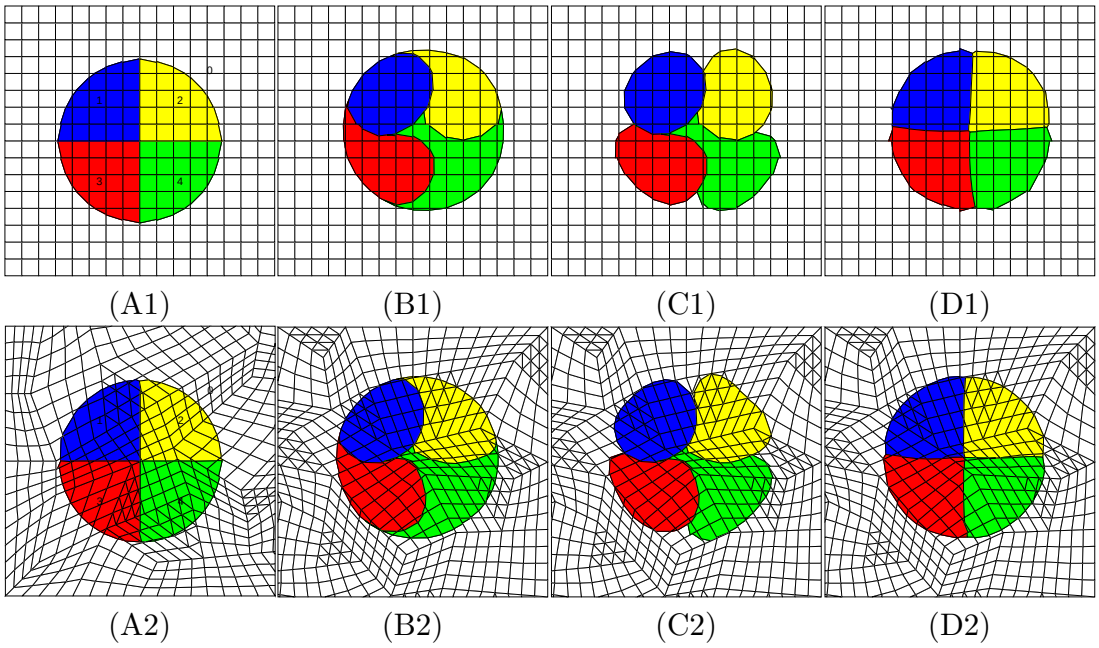


Figure 2.22 – *Numerical results from paper [17]. Final configuration of the four material circle shown in (A1) with material numbers after diagonal translation with a velocity of* $(1,1)$ *at time* $t = 0.5$ *using the interface reconstruction methods : (B1) Youngs' with material order* $(0,1,2,3,4)$ *(C1) Youngs' with material order* $(1,2,3,4,0)$ *(D1) our second order method. (A2)-(D2) show the same results on an unstructured mixed triangle and quadrilateral grid.*

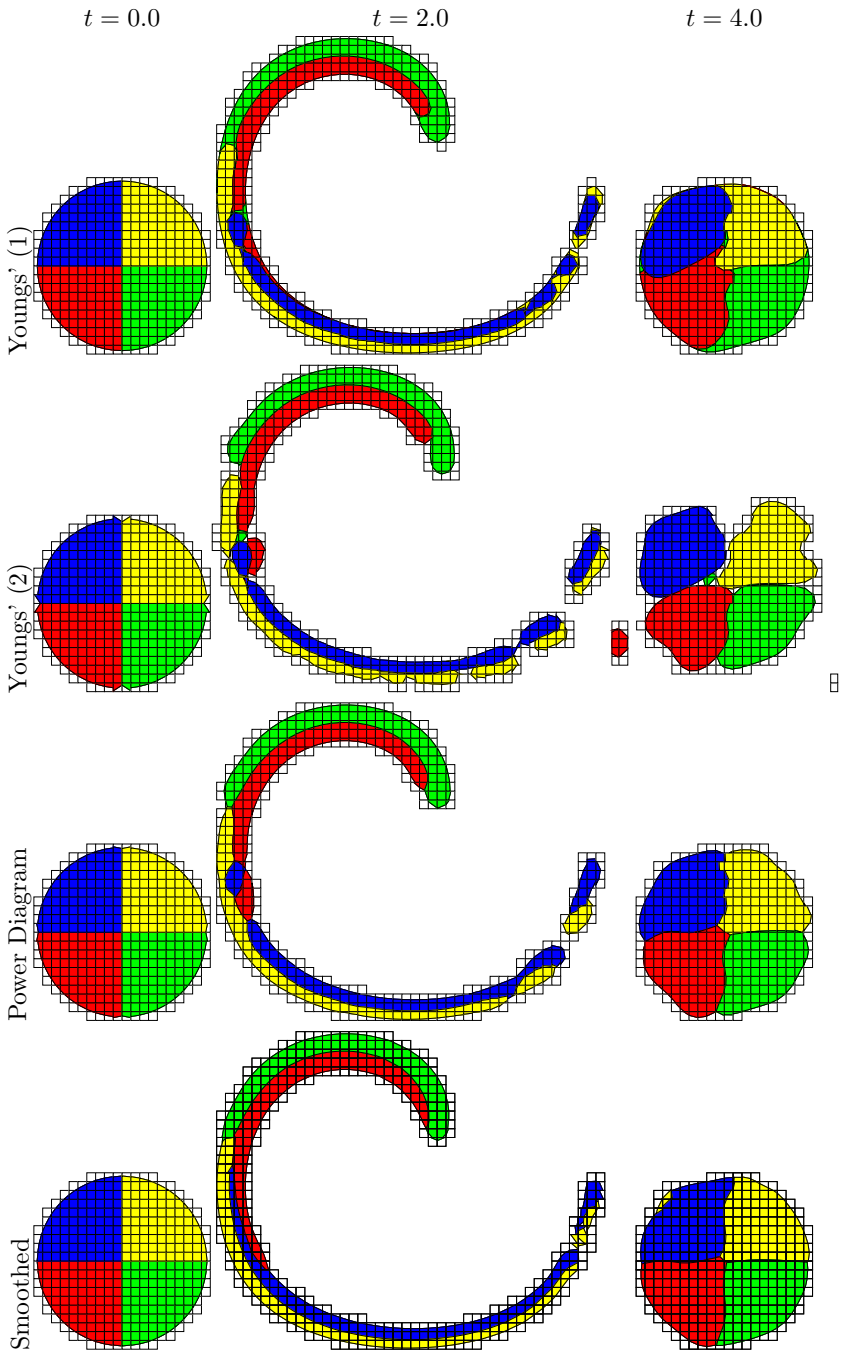


FIGURE 2.23 – *Numerical results from paper [17]. Material interface configuration for the four material vortex test initialy $t = 0$ (left column) at maximum stretch $t = 2$ (middle column) and at complete reversal time $t = 4$ (right column) run on a 64×64 grid. The material numbers are 0 for the white material, 1 for blue, 2 for yellow, 3 for red and 4 for green. For the method Youngs' (1), the material ordering was $(0, 1, 2, 3, 4)$. For method Youngs' (2), the order was $(1, 2, 3, 4, 0)$. Power diagram corresponds to the firts-order method from [13] whereas smoothed corresponds to the second-order method from [17].*

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:643–659
Published online 27 June 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1544

Material order-independent interface reconstruction using power diagrams‡

Samuel P. Schofield[1,*,†], Rao V. Garimella[1], Marianne M. Francois[2] and Raphaël Loubère[3]

[1]*Mathematical Modeling and Analysis (T-7), Los Alamos National Laboratory, Los Alamos, NM, U.S.A.*
[2]*Continuum Dynamics (CCS-2), Los Alamos National Laboratory, Los Alamos, NM, U.S.A.*
[3]*Mathematical Institute of Toulouse UMR-5219, Toulouse, France*

SUMMARY

We have developed a new, multi-material, piecewise linear interface reconstruction method that correctly locates the position of each material in the mesh cell and matches the required volume fractions with no material ordering required. This is different from other volume tracking interface reconstruction methods in which an improper material ordering may result in materials being incorrectly located within the cell. The new method utilizes a type of weighted Voronoi diagram, known as a power diagram, to reconstruct the interface from approximate material locations derived either from a particle model or quadrature formula. It works on structured and general polygonal grids, for an arbitrary number of materials and can be naturally extended to three dimensions. Published in 2007 by John Wiley & Sons, Ltd.

Received 12 January 2007; Revised 30 April 2007; Accepted 12 May 2007

KEY WORDS: volume of fluid; interface reconstruction; material ordering; power diagram

1. INTRODUCTION

The effective management and capture of interfaces is essential for accurate and reliable simulation of multi-material and multi-phase flows. Due to their strict conservation of materials, volume-of-fluid (VOF) methods using interface reconstruction are widely used and will be the focus of this article. VOF methods do not explicitly track the interface between materials, but rather advect volume fractions which prescribe the material composition of each cell of the mesh. When the interface between materials is needed, the interface is recreated based on the material volume fraction in the cell and its surrounding cells [1–4].

A common problem impacting these reconstruction methods is their dependence on a specified material ordering, i.e. if more than two materials are present in a cell, the reconstruction may depend on the sequence in which the materials are processed. This is undesirable as it may improperly locate materials within the cell. In a finite volume implementation, this may result in material being incorrectly fluxed into neighbouring cells.

In this article, we propose and demonstrate a method that can reconstruct a multi-material interface with no dependence on material ordering. The method is very general: it works on unstructured grids, accommodates an arbitrary number of materials and extends naturally to three dimensions. The method utilizes a particle attraction model or approximate centroid calculation to infer the relative location of the materials in the cell. Using that information, the interface is reconstructed using a weighted Voronoi diagram, known as a power diagram, such that the required volume fractions are matched.

*Correspondence to: Samuel P. Schofield, T-7 MS B284, Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.
†E-mail: sams@lanl.gov
‡This article is a U.S. Government work and is in the public domain in the U.S.A.

Contract/grant sponsor: US Department of Energy; contract/grant number: DE-AC52-06NA25396
Contract/grant sponsor: DOE Advanced Simulation and Computing (ASC) program

Published in 2007 by John Wiley & Sons, Ltd.

WILEY InterScience® DISCOVER SOMETHING GREAT

2. INTERFACE RECONSTRUCTION IN VOLUME-OF-FLUID METHODS

The VOF method, originally developed by Hirt and Nichols [5], advects the fractional volumes of each fluid in the cell to track materials in a flow simulation. The volume fraction, $f_m(\mathscr{C}_i)$, of a material, m, in a cell, $\mathscr{C}_i$ of volume $\|\mathscr{C}_i\|$ is defined as

$$f_m(\mathscr{C}_i) = \frac{A_m(\mathscr{C}_i)}{\|\mathscr{C}_i\|} \tag{1}$$

where $A_m(\mathscr{C}_i)$ is the volume of the material in the cell. Early VOF methods used a simple interface that was defined to be a coordinate axis aligned line within each cell that partitioned the cell into the correct volume fractions. This is often referred to as the simple line interface calculation (SLIC) due to Noh and Woodward [6]. This interface structure was natural when combined with directionally split advection.

Youngs [7, 8] extended the method to permit the material interface to have an arbitrary orientation within the cell. In Youngs' method, the outward normal of the interface separating a material within a computational cell is taken to be the negative gradient of the volume fraction function, estimated using the volume fractions of that material in the neighbouring cells. On unstructured grids, the gradient of the volume fraction function is calculated with either a Green–Gauss formula [9] or a least-squares technique [3]. On either structured or unstructured grids, the interface is then defined by locating a line having the prescribed normal that cuts off the correct volume of material from the computational cell. This and other methods that allow an arbitrarily oriented linear interface within a cell are referred to as piecewise linear interface calculation (PLIC) methods. In general, the methods are first order and the interfaces will be discontinuous at cell boundaries. However, there are extensions that make the reconstruction second order using a local optimization [10] or interface smoothing [11, 12].

While PLIC methods work well for two-material cells, the method must be extended to reconstruct the interfaces between materials in computational cells with three or more materials present. In the 'onion-skin' approach, each material interface is assumed to separate two materials and consists of a single line segment with both endpoints on the computational cell boundary. This form of reconstruction works for simple layer structures only [7, 13]. A more general approach is

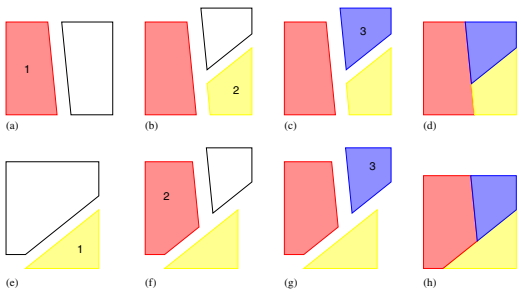


Figure 1. Nested dissection interface reconstruction for three materials: (a) the first material is removed leaving a smaller available polygon; (b) the second material is removed from the available polygon; (c) the remaining available polygon is assigned to material 3; (d) the resulting partitioning of the computational cell; and (e)–(g) show the same procedure but the materials are processed in a different order leading to a different reconstruction (h).

'nested dissection' [14], where each material is separated from the others in a specified order. This process is illustrated in Figure 1. In Figure 1(a) the first material is separated from the remaining materials leaving a pure material region and an available polygon. Then in Figure 1(b), the second material is removed from the available polygon according to its calculated normal and required volume. Finally, in Figure 1(c), the remaining available polygon is assigned to the last material. This same process can be used for an arbitrary number of materials. Unfortunately, the result of this method depends on the order in which the materials are processed. In Figures 1(e)–(g), the reconstruction of the cell with the same volume fractions and interface normals is performed, but with the materials processed in a different order, leading to a different reconstruction as is seen by comparing Figures 1(d) and (h).

The effects of material order dependency are further illustrated in Figure 2. With the correct material ordering used in Figure 2(b), the interface reconstructed by a first-order PLIC method, is close to the correct configuration shown in Figure 2(a). In this example, the reconstruction would be identical to the correct configuration using a second-order reconstruction method. Using an incorrect ordering results in substantial degradation of the interface as shown in the reconstructions in Figures 2(c) and (d).

The calculation of the interface normal is also affected by the presence of multiple materials. In Youngs' method, the gradient of the volume fraction function gives the normal of the interface. In two-material simulations, the orientation of the interface is independent of which material is used to calculate the normal since the volume fractions satisfy the relationship $f_1 = 1 - f_2$ which gives $\nabla f_1 = -\nabla f_2$. However, for multi-material simulations this simple relationship between the volume fractions does not hold. Since the gradient or normal direction is calculated using the

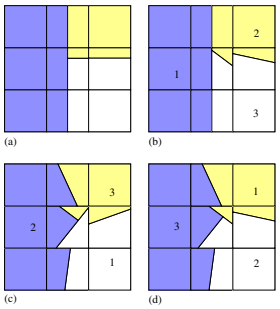


Figure 2. (a) Correct reconstruction and (b)–(d) first-order PLIC-based reconstruction using different material orderings. The numbers designate the order in which the materials were processed.

volume fraction data in a neighbourhood of the cell being reconstructed, it may happen that in a two-material cell, $\mathscr{C}_i$, $\nabla f_1(\mathscr{C}_i) \neq -\nabla f_2(\mathscr{C}_i)$ due to the presence of another material in the neighbourhood used for the gradient calculation. Two different reconstructions are possible depending on whether $\nabla f_1(\mathscr{C}_i)$ or $\nabla f_2(\mathscr{C}_i)$ is chosen as the direction for the interface normal. This introduces a material order dependence in even two-material cells. This is shown in Figures 2(c) and (d) as the interface of the two-material cell on the right side of the mesh has a different normal depending on which material was processed first.

These incorrect reconstructions may adversely impact the material advection in the simulation. If the advection routine is based on fluxing volumes through the edges of the cell, the improper material ordering may result in a material being fluxed early or not at all. For example, if the flow is moving diagonally towards the top right in Figure 2(c), the white material will move into the top right cell prematurely. This can lead to a breakup of the interface, a phenomenon known as 'flotsam and jetsam'.

Selecting a global ordering can be problematic as the appropriate ordering for one region of the mesh may be quite wrong for another. To remedy this, there has been some work on deriving the material order. The geometrically derived material priority by Mosso and Clancy [15] is based on the assumption of a layer structure and works by approximating the local centre of mass of each material; then, based on the relative locations along a line, it selects an ordering. A similar approach was developed by Benson [16]. However, both methods can fail in the presence of a triple point.

In order to eliminate the material order dependence in multi-material interface reconstruction, we have developed a novel method consisting of two steps:

1. Relative locations of materials in multi-material cells are inferred using a particle model or quadrature formula.

2. Using these locations, the interface is reconstructed using a form of weighted Voronoi diagram known as a power diagram.

The method is completely general, working on general polygonal grids with an arbitrary number of materials in each cell. In addition, it can be naturally extended to three dimensions. In contrast to existing SLIC or PLIC methods, all materials are processed simultaneously and, as such, have no material order dependency. Furthermore, unlike methods such as the triple point method of Choi and Bussmann [13], no additional extensions are required to accommodate an arbitrary number of materials within the cell. The reconstruction will automatically give either the appropriate layer structure or multiple triple point configuration.

3. DETERMINATION OF MATERIAL LOCATIONS

We describe two methods for determining the approximate location of each material in the cell:

1. A particle model, where particles representing the materials evolve according to an attraction/repulsion model.
2. A method of approximately calculating the centre of mass of a material in a neighbourhood of the cell.

3.1. Material location with a particle attraction–repulsion model

In the first step of the method, a number of particles representing the materials are placed in multi-material cells and any pure or mixed neighbouring cells. A particle, P_i, has a position, $\mathbf{x_i}$, velocity $\mathbf{v_i} = \mathrm{d}\mathbf{x}_i/\mathrm{d}t$ and material $m(i)$, and is constrained to stay within the cell in which it is initially placed.

Taking inspiration from molecular dynamics [17, 18] and smoothed particle hydrodynamics [19, 20], we evolve the particle positions according to 'forces' based on the particles' relative locations and materials. The positions of the particles are updated through time integration of a set of ordinary differential equations,

$$\frac{\mathrm{d}\mathbf{x_i}}{\mathrm{d}t} = \mathbf{V_i}$$
$$\mathbf{V_i} = \sum_{j:m(j)=m(i)} \mathbf{V_{att}}(\mathbf{x_i}, \mathbf{x_j}) + \sum_{j:m(j)\neq m(i)} \mathbf{V_{rep}}(\mathbf{x_i}, \mathbf{x_j}) \tag{2}$$

where $\mathbf{V_{att}}$ and $\mathbf{V_{rep}}$ are the prescribed attractive and repulsive 'forces' in the direction $\mathbf{x_j} - \mathbf{x_i}$. Particles of the same material attract each other until they are very close, at which point they start to repel each other. Particles of different materials repel each other. In our tests, the particles start at random locations within their cell, but they can be initialized using other means such as their relative locations in a cell at a previous time step.

The particle–particle 'forces' (plotted in Figure 3) are prescribed as

$$\mathbf{V_{att}}(\mathbf{x_i}, \mathbf{x_j}) = \begin{cases} -1, & d_{ij} < \delta \\ 1 - 2d_{ij}^4 + d_{ij}^8, & \delta \leqslant d_{ij} \leqslant 1 \\ 0, & d_{ij} > 1 \end{cases} \tag{3}$$

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

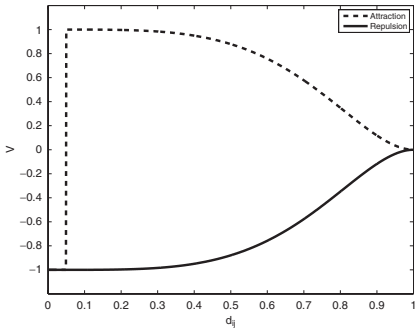


Figure 3. Particle attraction and repulsion 'forces' used in the model.

$$\mathbf{V_{rep}}(\mathbf{x_i}, \mathbf{x_j}) = \begin{cases} -(1 - 2d_{ij}^4 + d_{ij}^8), & d_{ij} \leqslant 1 \\ 0, & d_{ij} > 1 \end{cases} \tag{4}$$

where $d_{ij} = \|\mathbf{x_i} - \mathbf{x_j}\|/2.5h$ is the distance between points scaled by an interaction distance, taken to be 2.5 times the characteristic mesh size h, and $\delta = 0.05$. Unlike a traditional mechanical model, the 'forces' here actually prescribe the instantaneous velocities of the particles.

In a cell, $\mathscr{C}_i$, the number of particles, $N(\mathscr{C}_i)$ is

$$N(\mathscr{C}_i) = \left\lfloor N_\mathrm{p} \times \frac{\|\mathscr{C}_i\|}{A_0} \right\rfloor \tag{5}$$

where N_p is a prescribed constant (usually around 30), $\|\mathscr{C}_i\|$ is the area of the cell, A_0 is a reference cell area for the grid (for example, on a uniform Cartesian grid, $A_0 = h^2$ where h is the grid spacing) and $\lfloor a \rfloor$ is the floor function giving the greatest integer less than or equal to a. Each particle has a designated material type, corresponding to a material present in the cell. Each material that is present in the cell is represented by the same number of particles, $N(\mathscr{C}_i)/N_m^i$, where N_m^i is the number of materials present in the cell. We found that making the number of particles representing each material proportional to the volume fraction of the material often leads to unsatisfactory results. If the volume fraction is small, the material will be represented only by a few particles, which are not sufficient to provide a reliable estimate of the location of the material within the cell. In addition, we found that for unstructured, general polygonal grids, making the number of particles proportional to the area of the cell was important. Otherwise, the particles tend to cluster in regions of the mesh with a concentration of smaller cells.

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

Once the particles are distributed, the particle model is run. Since the model prescribes instantaneous velocities and not true forces, the particles may remain in perpetual motion unless the system is forced to 'cool'. The velocity of each particle is rescaled at each time step to decrease the kinetic energy of the system and force the particles to settle into a final configuration. At time step n in the time integration of Equation (2), the kinetic energy of all the particles is

$$\mathrm{KE}^n = \sum_i \frac{1}{2}\|\mathbf{V_i}\|^2 \tag{6}$$

After the system is sufficiently agitated, typically after 5 to 10 time steps, we force the kinetic energy to decrease as,

$$\mathrm{KE}^{n+1} \leqslant \alpha \mathrm{KE}^n \tag{7}$$

where $0 < \alpha < 1$. In practice, α is set to be 0.7–0.9. If $\mathrm{KE}^{n+1} \geqslant \mathrm{KE}^n$, all the particle velocities are scaled as

$$\mathbf{V}_i' = \sqrt{\alpha \frac{\mathrm{KE}^n}{\mathrm{KE}^{n+1}}}\,\mathbf{V}_i \tag{8}$$

To speed up the calculation, we use a variable time step with a new Δt calculated after each time step as

$$\Delta t = \frac{0.1}{2\|\mathbf{V}_\mathrm{max}\|} \tag{9}$$

where $\|\mathbf{V}_\mathrm{max}\| = \max_i \|\mathbf{V}_i\|$ where $\mathbf{V}_i$ is as defined in Equation (2).

The positions are then updated as

$$\mathbf{x_i}^{n+1} = \mathbf{x_i}^n + \Delta t \mathbf{V_i}' \tag{10}$$

If a particle goes outside the cell, it is placed back in the cell by repositioning it to the centre of the triangle formed by the old position, the new position, and the centre of the cell. If that fails, the particle is kept in its old position.

The particles are allowed to evolve for a number of time steps until the average kinetic energy of each particle has dropped below a specified stopping criterion. The particle model exhibits rapid convergence to the particle clusters, usually requiring under 20 time steps to converge to approximately the final positions. We have conducted statistical tests that show the model displays little sensitivity to the random initial particle positions, with standard deviations in the final material locations typically less than 5% of the mesh spacing [21].

Once the velocity of the particle has dropped to a sufficiently low threshold, the positions of the materials in the cell need to be derived from the final location of the particles. However, the particles of a material may form multiple groups. A clustering algorithm is needed to detect the multiple clusters and utilize that information to capture the subcell structure. A naive averaging of the particle positions for each material can yield reasonable results if each material in the cell is accurately described by a single convex polygon, but it will not detect the presence of multiple clusters of particles which may occur when thin filament-like structures are present.

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

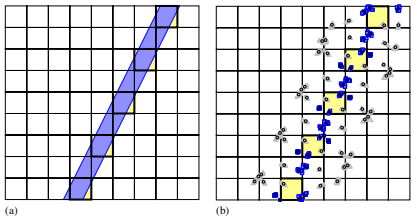


Figure 4. (a) Thin filament structure and (b) converged particle locations from the volume fraction data. The cluster locations are designated with the dark circles. The highlighted cells show where the filament has multiple interfaces within the cell. In these cells, away from the boundary, multiple clusters of particles are detected by the k-means clustering algorithm.

A simplified version of the k-means clustering algorithm [22], is used to detect if the particles of a material form one or two clusters. The algorithm proceeds as follows, for each material:

1. Initialize cluster 1 with the position of any particle of that material.
2. Initialize cluster 2 with the position of the particle of the same material farthest from cluster 1.
3. Assign each particle to the closest cluster.
4. Compute the mean position of all particles in each cluster to obtain new cluster locations.
5. Repeat steps 3 and 4 until the cluster locations converge or one cluster contains no particles.
6. If the two clusters are sufficiently close, that is, the distance between them is less than 10% of the characteristic mesh size, then use the average location of all of the particles of that material instead.

The clustering algorithm converges rapidly and is robust.

Figure 4 shows the example of a narrow filament. The particles reflect the thin filament structure present. The clustering algorithm is able to detect the presence of the multiple clusters within a cell. With the exception of cells at the boundary, multiple clusters are present and detected in all of the cells where the filament has two interfaces within the cell. These cells are indicated as the highlighted cells in Figure 4.

If there are multiple clusters of a material, the volume of material must be partitioned between the clusters. The number of particles in each cluster does not give a reliable means to do this as it can vary depending on the initial conditions. Instead, we partition the material volume equally between the clusters as we have no knowledge of the relative volume of the two regions of the material within the mesh cell.

3.2. Material location via approximate centres of mass

An alternative to the particle model is the direct calculation of an approximate centre of mass of each material in a subset of the mesh around the cell being reconstructed. In a region of the layer

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

Ω, assume a material occupies a non-trivial subset $\Omega_m \subseteq \Omega$. The centre of mass of the material in the mesh region, Ω is given as

$$\mathbf{x_m}(\Omega) = \frac{\int_\Omega \mathbf{x}\chi_m(\mathbf{x})\,\mathrm{d}\mathbf{x}}{\int_\Omega \chi_m(\mathbf{x})\,\mathrm{d}\mathbf{x}} \tag{11}$$

where $\chi_m(\mathbf{x})$ is the characteristic function for material m, defined as

$$\chi_m(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega_m \\ 0, & \mathbf{x} \notin \Omega_m \end{cases} \tag{12}$$

However, Ω_m and χ_m are unknown.

If we assume that within each cell the centre of mass of each material contained within that cell is approximately the centroid of the mesh cell, then Equation (11) may be approximated for a region of the mesh larger than the cell of interest. In particular, if we take Ω to be the mesh cell we wish to reconstruct and all of its neighbours, the approximation takes the form

$$\tilde{\mathbf{x}}_m(\mathscr{C}_i) = \frac{\sum_{\mathscr{C}_j \in \mathscr{N}_i} f_m(\mathscr{C}_j)\|\mathscr{C}_j\|\mathbf{x}_\mathbf{c}(\mathscr{C}_j)}{\sum_{\mathscr{C}_j \in \mathscr{N}_i} f_m(\mathscr{C}_j)\|\mathscr{C}_j\|} \tag{13}$$

where $\mathscr{N}_i$ is the set of mesh cells including the cell being reconstructed, $\mathscr{C}_i$, and all of its neighbouring cells. $\mathbf{x}_\mathbf{c}(\mathscr{C}_j)$ is the centroid of mesh cell $\mathscr{C}_j$. Similar formulations for a regional centre of mass derived from the volume fraction data have been used by a number of investigators [15, 16, 23].

These approximate centres of mass will be for the region $\mathscr{N}_i$ and will in general not be contained within the cell $\mathscr{C}_i$. However, it gives some information regarding the relative location of materials in a cell.

4. POWER DIAGRAM-BASED INTERFACE RECONSTRUCTION

Once the materials are located in a cell, the interfaces within the cell, separating the materials, are constructed using a power diagram. A power diagram or Laguerre diagram [24, 25] is a generalization of a Voronoi diagram generated from a set of points, S, each with an associated radius or weight. In this context, the generators will be the points determined either by the particle clusters or the approximate centroids. The Laguerre distance from a point $\mathbf{x} \in \mathbb{R}^n$ to a point mass, $s_i \in S$ with $s_i = (\mathbf{x_i}, w_i)$ is defined as

$$d_\mathrm{L}^2(\mathbf{x}, s_i) = d^2(\mathbf{x}, \mathbf{x_i}) - w_i \tag{14}$$

where $d^2(\mathbf{x}, \mathbf{x_i}) = \sum_{i=1}^n (x - x_i)^2$ is the usual Euclidean distance in $\mathbb{R}^n$. If w_i is replaced with w_i^2 in Equation (14), the resulting distance is called the *power* of the point $\mathbf{x}$ with respect to $\mathbf{x_i}$.

Each cell in the power diagram is the set of points

$$\mathrm{cell}(s_i) = \{\mathbf{x} \in \mathbb{R}^n | d_\mathrm{L}^2(\mathbf{x}, s_i) < d_\mathrm{L}^2(\mathbf{x}, s_j)\ \forall s_j \in S, s_j \neq s_i\} \tag{15}$$

As with Voronoi diagrams, each power diagram cell is convex.

The weight associated with a point generator can be interpreted as the square of the radius of a circle centred at that point. The power bisector (a chordale in Aurenhammer's terminology

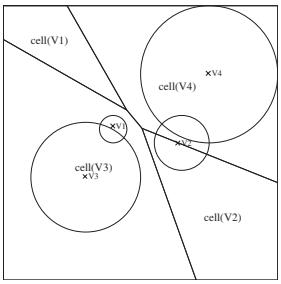


Figure 5. Power diagram with four generators ($V1$–$V4$) and their weight circles. The generators $V1$ and $V2$ do not lie within their corresponding cells.

[24]) between two points $s_i = (\mathbf{x}_i, w_i)$ and $s_j = (\mathbf{x}_j, w_j)$ is the line perpendicular to the segment connecting the points $\mathbf{x}_i$ and $\mathbf{x}_j$ and is located by finding a point, $\mathbf{x}_0$ on that segment such that $d_\mathrm{L}^2(\mathbf{x_0}, s_i) = d_\mathrm{L}^2(\mathbf{x_0}, s_j)$.

An example of power diagram is shown in Figure 5. Clearly, if all point masses have equal weight (or radius), the power diagram reduces to the usual Voronoi diagram. Unlike a Voronoi diagram, a point in the point set generating the diagram does not necessarily lie in the cell to which it corresponds, as is this case for generators $V1$ and $V2$ in Figure 5. Furthermore, the cell corresponding to a given point mass, may be trivial, that is $\mathrm{cell}(s_i) = \emptyset$. In practice, this is not a problem. Imai *et al.* [25] provide a useful lemma that provides a sufficient condition for the power diagram cell of a point to be non-trivial:

> Given the power diagram for a finite set of point masses, $S = \{s_1, \ldots, s_n\}$, $\mathrm{cell}(s_i)$ is non-trivial if s_i lies on a corner of the convex hull of S.

If only three materials are present (i.e. S consists of only three point masses), then the point mass corresponding to each material must necessarily be a corner of the convex hull assuming the three points are not collinear. This ensures that the power diagram in each cell with three materials will always be non-trivial for all choices of weights. If the points are collinear, then the cell is partitioned by two parallel lines which can obviously be made to cut off the appropriate volume fractions. The same argument applies to three and four material cells in 3D. Proving this for larger numbers of materials is more difficult. However, we have not encountered a scenario where we are unable to find a power diagram that matches the required volume fractions.

The power diagram can be constructed in a number of ways. A power diagram may be created through a randomized, incremental algorithm [26], similar to the incremental construction of a Delaunay triangulation. A Voronoi diagram of the point generators may also be efficiently converted into a power diagram [27]. However, we have chosen a simple algorithm that intersects all of the

mutual power bisectors. It has $\mathscr{O}(n^2)$ asymptotic complexity in the number of generator points. For our applications, the number of materials in a cell, corresponding to the maximum possible number of power diagram cells, is small (typically 5 or less), so the asymptotic complexity of the construction algorithm is not a problem. In addition, it has proven to be robust in finite precision arithmetic.

4.1. Matching volume fractions

The volume fractions of the materials in a cell are matched by iteratively adjusting the weights of the point generators, thereby adjusting the area of the power diagram cells once clipped to the mesh cell. This requires the solution of a set of non-linear equations

$$A_m(\omega_1, \ldots, \omega_{N_m}) = \|\mathscr{C}_i\| f_m, \quad m = 1, \ldots, N_m \tag{16}$$

where $A_m(\omega_1, \ldots, \omega_{N_m})$ is the area of the power diagram corresponding to material m after it has been clipped by the bounding mesh cell polygon with area $\|\mathscr{C}_i\|$. $f_m(\mathscr{C}_i)$ is again the volume fraction for material m in cell $\mathscr{C}_i$. The constraint

$$\sum_m^{N_m} A_m(\omega_1, \ldots, \omega_n) = \|\mathscr{C}_i\| \tag{17}$$

reduces the number of equations to $N_m - 1$. Specifically, this is done by forcing one of the weights to be a specified value. This enforces a unique set of weights for the desired power diagram.

A Newton procedure with a finite-difference Jacobian is used to solve Equations (16) and (17). Some caution is required, since the area of each cell is bounded above and below, that is

$$0 \leqslant A_m(\omega_1, \ldots, \omega_n) \leqslant \|\mathscr{C}_i\|, \quad m = 1, \ldots, N_m \tag{18}$$

For extreme values of the weights, some of the power diagram cells will be outside of the mesh cell and as such have zero area once clipped to the mesh cell. Furthermore, the A_m will be flat (that is they have a zero gradient) making the Newton procedure fail. As a result, the Newton procedure needs to adjust for overshoots to make sure it does not end up in this region. This is simply done by reducing the size of the Newton step at each iteration if it exceeds those bounds. We found the procedure to be robust and efficient, typically matching the required volumes to within $10^{-12}\|\mathscr{C}_i\|$ in 3–6 iterations.

For the initial guess, we use equal weights for all the point generators if all of the generators lie within the mesh cell being reconstructed. If any of the generators are outside the cell, as may happen with the approximate centre of mass calculation, the initial weights are assigned such that the power bisectors between all the generators go through the centroid of the cell. This ensures that the initial power diagram will not have any cells outside of the mesh cell.

4.2. Reconstruction fidelity

The use of the power diagram to reconstruct the interface is based on the assumption that we have been able to obtain an approximation to the location of each material in the cell. Indeed, if the point generators for the power diagram are the exact centres of mass of the material subcells, the reconstruction is representative of the actual configuration. In Figures 6 and 7, the power diagram reconstruction of different three-material cells, using the actual material centres of mass

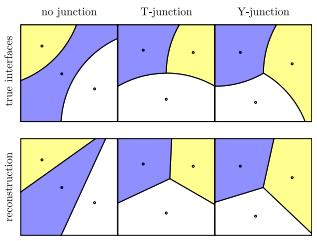


Figure 6. The true material interfaces and their centres of mass are shown in the top row with the power diagram reconstruction using the centres of mass in the bottom row. Here, the radius of curvature of all of the true interfaces is set to $R/h = 1$.

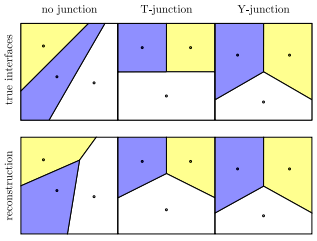


Figure 7. The asymptotic convergence of the power diagram reconstruction using the centres of mass as the radius of curvature of the interfaces becomes large. The power diagrams are unable to recreate the interfaces exactly. Here $R/h = 256$.

as generators, are shown. The power diagram provides a reasonable approximation to the true interface for interfaces with a radius of curvature comparable to the mesh spacing. As the radius of curvature goes to infinity, the interfaces straighten, but the power diagram does not necessarily converge to the true interface as illustrated in Figure 7. Still, the materials retain their relative locations within the cell.

In the following section, we present some static interface reconstruction examples using the power diagram-based reconstruction with point generators derived from the particle model and the approximate centroid calculation.

5. NUMERICAL EXPERIMENTS

To assess the overall performance of the method, a number of numerical experiments were performed.

In Figure 8, a four-material interface reconstruction on a structured grid is shown. As can be seen, our reconstructions respect the symmetry in the problem. The particle model derived material locations shown in Figure 8(a) are completely symmetric and result in a near perfect reconstruction. The reconstruction using approximate centre of mass-derived material locations shown in Figure 8(b) is exact. Observe that the material locations shown in the inset are outside of the centre cell. In the material order-dependent Youngs' method reconstructions in Figures 8(c) and (d), the reconstruction in the centre cell is not symmetric due to the material ordering. In fact, for this example, no material ordering will enable Youngs' method to create a completely symmetric reconstruction.

For the unstructured grid shown in Figure 9, neither power diagram-based reconstruction exactly reproduces a straight line, indicating that the reconstruction cannot be second order [10]. Still, in special treatment was used at the boundary for either the particle method or approximate centroid method. With the correct material ordering, Youngs' method produces a very good reconstruction as shown in Figure 10(a). This motivates the use of a mixed method using Youngs' method for two-material cells and a power diagram-based reconstruction for three or more material cells. These reconstructions are shown in Figures 10(e) and (f). The multi-material cells, shown with bold outlines, used a power diagram-based method while the rest of the interface was reconstructed

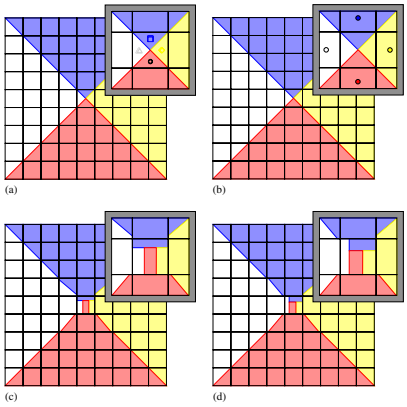


Figure 8. Four-material interface reconstruction using: (a) particles and power diagrams; (b) approximate centroids and power diagrams; and (c), (d) Youngs' method with two different material orderings. The insets show the four-material cell at the centre of the mesh. The converged particles locations for the centre cell are also shown in the inset in (a). The approximate centres of mass for the centre cell are shown in (b).

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

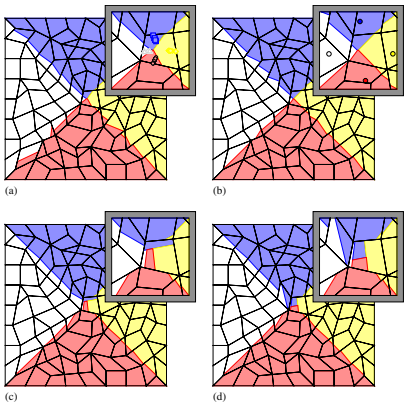


Figure 9. Four-material interface reconstruction on an unstructured grid using: (a) particles and power diagrams; (b) approximate centroids and power diagrams; and (c), (d) Youngs' method with two different material orderings. The insets show the four material cell at the centre of the mesh with the particles for the centre cell in (a) and the approximate centres of mass for the centre cell in (b).

the multi-material centre cell, the power diagram-based reconstructions in Figures 9(a) and (b) are more representative of the structure of the true configuration. In Figures 9(c) and (d), the material order dependency provides significantly different reconstructions for the four-material centre cell. This and the structured example in Figure 8 demonstrate the strength of our method in reconstructing multi-material cells. In two-material cells, Youngs' and related methods do quite well, as is shown in Figures 9(c) and (d) in the regions away from the centre. An effective strategy is to *use existing methods for two-material cells, and a power diagram-based method for cells containing three or more materials*. In Figures 8 and 9, the power diagram reconstruction was used on the entire mesh, which would not typically be done in practice.

Figure 10 shows the reconstruction of a filament-type structure that is not aligned with the grid. The filament is preserved with all the three-material cells showing the proper material positions. The power diagram reconstructions shown in Figures 10(c) and (d) do not reverse the location of the materials relative to the filament as does Youngs' reconstruction with the wrong material ordering as shown in Figure 10(b). The power diagram-based method fails to exactly reproduce a straight line in both Figures 10(c) and (d). The reconstruction in the cells at the top and bottom of the grid in Figures 10(c) and (d) could be improved with better boundary treatment. No

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

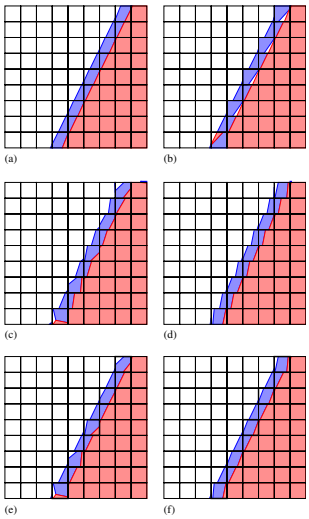


Figure 10. Multi-material interface reconstruction for a filament: Youngs' reconstruction with (a) correct material ordering and (b) incorrect material ordering; (c) particles and power diagram reconstruction in all cells; (d) approximate centroids and power diagram-based reconstruction in all cells; (e) mixed approach using Youngs' method for two-material cells and particles and power diagram reconstruction in three or more material cells; and (f) mixed approach using Youngs' method and approximate centroids and power diagram reconstruction. The three-material cells using the power diagram-based reconstructions are in bold in (e) and (f). Notice the presence of the white material on the right-hand side of the filament in Youngs' reconstruction in (b). This can lead to 'flotsam and jetsam'.

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

using Youngs' method. These reconstructions do not have the material ordering problems as would Youngs' method used on all mesh cells.

6. CONCLUSIONS

We have developed a first-order method for material order-independent reconstruction of multi-material cells. It is based on the determination of the relative locations of each material in a cell and then reconstructing the interface using a power diagram, that once clipped to the mesh cell, matches the desired volume fractions. The method does not assume a topology for the material regions, i.e. a layer structure or triple point configuration. Furthermore, all of the material regions created will be convex.

The performance of the particle model is good. The average particle position converges to within a few per cent of its converged position in under 20 time steps of the attraction–repulsion model. Furthermore, the particles do not show much sensitivity to the random initial conditions used.

The approximate centre of mass approach typically gives better results, and relies on no externally supplied parameters. In addition, while being highly problem dependent, it can be around 50 times faster than the particle method. Neither the particle model nor approximate centre of mass method when combined with a power diagram-based reconstruction exactly reproduces a straight line, indicating both methods are only first-order accurate. While the potential for the particles to detect the presence of a thin filament within the mesh through the determination of multiple particle clusters is intriguing, we recommend the approximate centre of mass approach, particularly when subcell details such as thin filaments do not have to be resolved.

The power diagram-based interface reconstruction maintains the relative location of the materials within the cell and may be useful in other methods utilizing approximate material location information.

As two-material reconstruction algorithms are reliable and well established, this method would be most applicable for reconstruction of only cells containing more than two materials. This reconstruction method is currently being implemented in multi-material flow codes to further understand the combined errors of advection and interface reconstruction.

ACKNOWLEDGEMENTS

The authors would like to thank B. K. Swartz and M. J. Shashkov for their helpful discussions. This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program.

REFERENCES

1. Benson DJ. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:235–394.
2. Benson DJ. Volume of fluid interface reconstruction methods for multi-material problems. *Applied Mechanics Review* 2002; **55**(2):151–165.
3. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
4. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 1999; **31**:567–603.

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

5. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
6. Noh WF, Woodward P. SLIC (Simple line interface calculation). In *Lecture Notes in Physics*, vol. 59, van de Vooren AI, Zandbergen PJ (eds). Springer: Berlin, 1976; 330–340.
7. Youngs DL. Time dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic Press: New York, 1982; 273–285.
8. Youngs DL. An interface tracking method for a 3d Eulerian hydrodynamics code. *Technical Report AWE/44/92/35*, AWRE Design and Math Division, 1984.
9. Barth TJ. Aspects of unstructured grids and finite-volume solvers for Euler and Navier–Stokes equations. *VKI/NASA/AGARD Special Course on Unstructured Grid Methods for Advection Dominated Flows AGARD Publication R-787*, 1995.
10. Pilliod Jr JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics* 2004; **199**:465–502.
11. Garimella RV, Dyadechko V, Swartz BK, Shaskov MJ. Interface reconstruction in multi-fluid, multi-phase flow simulations. *Proceedings of the 14th International Meshing Roundtable*, San Diego, CA. Springer: Berlin, 2005; 19–32.
12. Swartz BK. The second-order sharpening of blurred smooth borders. *Mathematics of Computation* 1989; **52**(186):675–714.
13. Choi BY, Bussmann M. A piecewise linear approach to volume tracking a triple point. *International Journal for Numerical Methods in Fluids* 2007; **53**:1005–1018.
14. Dyadechko V, Shashkov MJ. Multi-material interface reconstruction from the moment data. *Technical Report LA-UR-06-5846*, Los Alamos National Laboratory, Los Alamos, NM, 2006.
15. Mosso S, Clancy S. A geometrically derived priority system for Young's interface reconstruction. *Technical Report LA-CP-95-0081*, Los Alamos National Laboratory, Los Alamos, NM, 1995.
16. Benson DJ. Eulerian finite element methods for the micromechanics of heterogeneous materials: dynamic prioritization of material interfaces. *Computer Methods in Applied Mechanics and Engineering* 1998; **151**: 343–360.
17. Allen MP, Tildesley DJ. *Computer Simulation of Liquids*. Oxford University Press: New York, 1987.
18. Haile JM. *Molecular Dynamics Simulation: Elementary Methods*. Wiley: New York, 1992.
19. Benz W. Smooth particle hydrodynamics: a review. *The Numerical Modelling of Nonlinear Stellar Pulsations: Problems and Prospect*. NATO ASI Series. Series C, Mathematical and Physical Sciences, vol. 302. Kluwer Academic Publishers: Dordrecht, 1989; 269–287.
20. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics—theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**:375–389.
21. Schofield SP, Garimella RV, Francois MM, Loubere R. Multi-material interface reconstruction using particles and power diagrams. *Technical Report LA-UR-06-8740*, Los Alamos National Laboratory, Los Alamos, NM, 2006.
22. Jain AK, Murty MN, Flynn PJ. Data clustering: a review. *ACM Computing Surveys* 1999; **31**(3):264–323.
23. Puckett EG, Saltzman JS. A 3d adaptive mesh refinement algorithm for multimaterial gas dynamics. *Physica D* 1992; **60**:84–93.
24. Aurenhammer F. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing* 1987; **16**(1):78–96.
25. Imai H, Iri M, Murota K. Voronoi diagram in the Laguerre geometry and its applications. *SIAM Journal on Computing* 1985; **14**(1):93–105.
26. Edelsbrunner H, Shah NR. Incremental topological flipping works for regular triangulations. *Algorithmica* 1996; **15**:223–241.
27. Anton F, Mioc D. On the conversion of ordinary Voronoi diagrams into Laguerre diagrams. *The Eleventh Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999; 150–153.

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:643–659
DOI: 10.1002/fld

Paper [17] is reproduced in the following pages.

Journal of Computational Physics 228 (2009) 731–745

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations

Samuel P. Schofield [a,*], Rao V. Garimella [a], Marianne M. Francois [b], Raphaël Loubère [c]

[a] Mathematical Modeling and Analysis (T-7), Los Alamos National Laboratory, Los Alamos, NM 87545, United States
[b] Computational Physics and Methods (CCS-2), Los Alamos National Laboratory, Los Alamos, NM 87545, United States
[c] Mathematical Institute of Toulouse, CNRS, University of Toulouse, France

ARTICLE INFO

Article history:
Received 17 January 2008
Received in revised form 18 September 2008
Accepted 29 September 2008
Available online 11 October 2008

Keywords:
Volume-of-fluid
Interface reconstruction
Multi-material flow
Material-order independence
Linear reconstruction
Centroids
Power diagrams

ABSTRACT

A new, second-order accurate, volume conservative, material-order-independent interface reconstruction method for multi-material flow simulations is presented. First, materials are located in multi-material computational cells using a piecewise linear reconstruction of the volume fraction function. These material locator points are then used as generators to reconstruct the interface with a weighted Voronoi diagram that matches the volume fractions. The interfaces are then improved by minimizing an objective function that smoothes interface normals while enforcing convexity and volume constraints for the pure material subcells. Convergence tests are shown demonstrating second-order accuracy. Static and dynamic examples are shown illustrating the superior performance of the method over existing material-order-dependent methods.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

Multi-material and multi-phase flows occur in a variety of natural phenomena and industrial processes. To accurately model such flows, it is essential to effectively capture and manage material interfaces. Due to their ability to strictly conserve mass, volume-of-fluid (VOF) methods using interface reconstruction are widely used in such simulations [1–4]. Originally developed by Hirt and Nichols [5], VOF methods do not explicitly track interfaces but rather track the volume of each material. When required, the interface position is computed using the volume fraction data. In a flow simulation, the volume fractions are updated by determining the flux of each material into or out of a computational cell although in multi-material compressible simulations, volume fractions may also be modified by mixture models like pressure equilibration [1]. Contemporary schemes use the reconstructed interface to obtain a better approximation to the material fluxes. Poor interface reconstruction directly affects material fluxes and can result in material being transported to the wrong locations and unphysical fragmentation of material.

Early VOF methods used a straight line aligned with a coordinate axis to partition the cell according to the material volume fractions [6]. Youngs [7,8] extended the method to permit the material interface to have an arbitrary orientation within the cell. Such methods, that allow a generally oriented interface within the cell, are referred to as piecewise linear interface calculation (PLIC) methods [3]. In Youngs' method, the outward normal of the interface separating a material from the rest of

* Corresponding author. Tel.: +1 505 606 1484; fax: +1 505 665 5757.
E-mail addresses: sams@lanl.gov (S.P. Schofield), rao@lanl.gov (R.V. Garimella), mmfran@lanl.gov (M.M. Francois), loubere@mip.ups-tlse.fr (R. Loubère).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.09.023

the cell is taken to be the negative gradient of the "volume fraction function." The "volume fraction function" is assumed to be a smooth function whose cell-centered values are given by the cell-wise material volume fractions. The gradient may be computed by finite-difference formulas [8], Green–Gauss formula [9] or a least-squares technique [3]. The interface is then defined as a line with the calculated normal that cuts off the correct volume of material from the computational cell. We will refer to all methods that compute the interface from the gradient of the volume fraction function as gradient-based methods. In general, gradient-based methods for interface reconstruction are only first-order accurate although they may exhibit nearly second-order accurate behavior for rectangular grids. However, there are extensions that make the reconstruction second-order accurate by smoothing the interface normals [10–12]. Finally, there is a class of methods that attempts to reconstruct curved interfaces but these methods are not in widespread use [13].

VOF–PLIC techniques have been successfully used to accurately simulate two-phase (or two-material) flows and free-surface flows in two and three dimensions. However, their application to flows involving three or more materials that come closer than the mesh spacing and even form junctions has been mostly *ad hoc*. Examples of such phenomena are flows of immiscible fluids (e.g. oil–water–gas), inertial confinement fusion (ICF), hypervelocity impact and penetration, dynamic compaction of multi-material powders [2] and detonation shock dynamics with multiple inert and energetic materials [14]. In the oil–water–gas flow example, the three-immiscible fluids can statically or dynamically exhibit a thin film structure along the interface (Fig. 1(a)) or form a contact line (triple point in 2D) at the intersection between the three fluids (Fig. 1(b)). In inertial confinement fusion (ICF), a spherical plastic target filled with deuterium–tritium (solid and gas, see Fig. 2) is compressed by laser-initiated X-rays while located within an experimental chamber called a hohlraum that is usually made of gold or tantalum. The resulting compression and possible fragmentation of the multi-material target imply that thin layers and multi-material junctions are expected in the simulation and must be properly resolved.

The most common extensions of PLIC to cells with more than two materials (multi-material cells) is to process materials one-by-one leading to a reconstruction that is strongly dependent on the order in which the materials are processed. In a strict sense, any cell with more than one material is a multi-material cell. However, we choose to distinguish two-material cells from cells with more than two materials by calling the latter multi-material cells. This is because we have to reconstruct the interface for only one material in a two material cell whereas we have to reconstruct multiple interfaces in a cell with three or more materials. In the "onion-skin" approach, each material interface is assumed to separate two materials and consists of a single line segment with both endpoints on the computational cell boundary. This form of reconstruction works only for simple layer structures [7,15] and even there it may create overlapping layers. A more general and correct approach is the "nested dissection" method [16,17], where each material is separated from the others in a specified order. In the method, a pure polygon (or polyhedron) representing the first material is marked out from the cell, leaving a mixed polygon for the remaining materials. Then, a polygon representing the second material is marked out from the mixed polygon and the process continues until the last material is processed.

With the "correct" material ordering, the interface reconstructed by one of the above methods is close to the correct configuration (Fig. 3(a)). However an incorrect ordering results in substantial degradation of the interface as shown Fig. 3(b) and (c). Sometimes, there may not be an ordering which will give the correct configuration by nested dissection (Fig. 4) or the correct ordering in one part of the domain may be incorrect in a different part. Two-material cells next to multi-material cells may also be affected by the order in which materials are processed as discussed in [18].

Finally, the presence of multiple materials in simulations creates special considerations for second-order accurate methods like LVIRA. In the LVIRA method [10], the interface normal within a cell is updated by minimizing an objective function for each cell, $\mathcal{C}_i$

$$F^{LVIRA} = \sum_{j \in \mathcal{N}_i} (f_m(\mathcal{C}_j) - \tilde{f}_m(\mathcal{C}_j, \mathbf{n}, b))^2 \tag{1}$$

where $\mathcal{N}_i$ are the vertex connected neighbors of cell $\mathcal{C}_i$, $f_m(\mathcal{C}_j)$ is the prescribed volume fraction for material m in cell j, and $\tilde{f}_m(\mathcal{C}_j, \mathbf{n}, b)$ is the volume fraction cut off in cell j by continuing the line with normal $\mathbf{n}$ and line constant b from cell i through cell j. However, in multi-material cells the volume of a cell cut off by extending the interface may not be representative of the error. This is because the procedure fails to account for the presence of an additional material behind the extended line, giving a large error in the volume fraction even when the interface normal is correct as illustrated in Fig. 5.

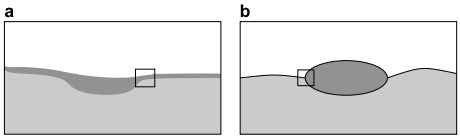


Fig. 1. Example of immiscible fluid interfaces (e.g. oil–water–air): (a) thin film and (b) triple point.

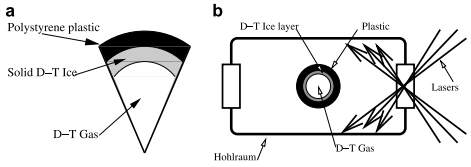


Fig. 2. (a) Sketch of an ICF target. (b) The target in the Holhraum (made of gold or tantalum), lasers enter the chamber generating X-ray emission that ultimately compress the target up to ignition of the deuterium–tritium gas.

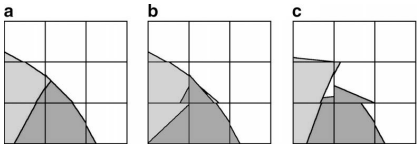


Fig. 3. (a) Reconstruction with the correct order, (b) and (c) reconstructions with an incorrect order.

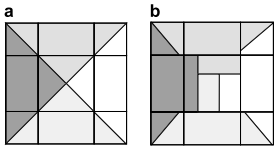


Fig. 4. (a) Four material junction and (b) order dependent reconstruction. In this example, the right configuration cannot be generated by any order of materials in a nested dissection process.

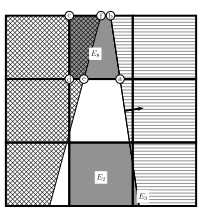


Fig. 5. In LVIRA, the interface normal for the white material in the center cell is adjusted by extending the interface from the center cell to the neighboring cells and minimizing the discrepancy between the volumes it cuts off in the neighboring cell, here labeled E_2, E_3, and E_8, and the prescribed volume fraction for that material in the neighboring cell. However, in computing the volume discrepancy the procedure may not be accounting for another material behind the interface as in $E8$ where the volume cutoff becomes the polygon `abed` instead of `abfe`.

Most importantly, these incorrect reconstructions adversely impact the material advection in flow simulations. In most cases, an improper material ordering may result in materials being advected prematurely (or belatedly) into neighboring cells. This can further lead to small pieces of the material getting separated and drifting away from the bulk of the material, a phenomenon known as "flotsam and jetsam".

To address this problem, there has been some work on deriving the material order automatically. The geometrically derived material priority by Mosso and Clancy [19] first approximates the local center of mass of each material; then, assuming a layer structure, it selects an ordering based on the relative locations of the approximate centroids along a line. A similar approach was developed by Benson [20]. However, both methods are primarily designed to capture layers and may fail for multi-material junctions.

Choi and Bussman [15] have addressed the material ordering problem by developing a method for reconstructing three-material T-junctions in two-dimensions. In a three-material cell, the interface between material 1 and materials 2 and 3 are reconstructed as in a gradient-based method. Then the normal of the interface between materials 2 and 3 are determined by a minimization procedure similar to LVIRA. This method is not truly order-independent since the first material must be chosen *a priori* and it is restricted to three-material cells. Also, the feasibility of applying it to 3D is not clear.

Caboussat et al. [21] also addressed the reconstruction of a triple point using a minimization procedure given by the neighboring normals. However, it is also restricted to three materials.

Bonnell et al. [22] have also described an interface reconstruction method that draws upon ideas of isosurfacing. This method can handle multiple materials in a cell automatically but the method is not guaranteed to match volume fractions exactly.

Previously, we presented a method that reconstructed interfaces in a material-order independent manner using a particle attraction–repulsion model or a regional quadrature formula to locate materials in cells and subdividing the cell using a weighted Voronoi diagram or power diagram [18]. This method is capable of generating a material-order independent partitioning of multi-material cells but is not second-order accurate.

In this article, we present a new second-order accurate VOF–PLIC method that reconstructs a multi-material interface with no dependence on the order in which materials are specified. The method matches volume fractions exactly as required by VOF interface reconstruction methods. The method works for arbitrary number of materials in general polygonal meshes. The reconstruction automatically gives either the appropriate layer structure or configurations with multiple material junctions. Finally, the method ensures that the reconstructed interface partitions the cell into convex material regions, an important consideration for a majority of material advection and remapping schemes. Finally, it reproduces straight line interfaces exactly and reconstructs interfaces with second-order accuracy.

Unlike the particle model method or quadrature formula method of [18], the new technique infers relative locations of materials based on a piecewise linear approximation to the characteristic function of each material. Using the approximate material locations, the cell is partitioned into material regions by a weighted Voronoi diagram while matching the volume fractions. Finally, the interface segments in the multi-material cells are smoothed with respect to their neighbors so that the method reproduces smooth interfaces with second-order accuracy even around multi-material junctions.

The following sections describe the three main steps of the new method, i.e. material location, power diagram reconstruction and smoothing. These are followed by results demonstrating that the method performs better than material order-dependent methods in static, geometric reconstructions and in dynamic advection tests. Convergence tests are presented to demonstrate that the method is second-order accurate.

2. Material location by piecewise linear reconstruction of the volume fraction function

Given the volume fractions of materials in cells in a mesh, our task is to determine the relative locations of materials in a multi-material cell. To do this, we must ideally recover the characteristic function for each material in the domain. While it is possible to reconstruct the characteristic function in 1D [23], no method (other than interface reconstruction itself) exists to do this in higher dimensions. Therefore, we make a simplifying assumption that a smooth function, called the volume fraction function, exists for each material and that its pointwise cell-centered values are given by cell-wise volume fraction data. This smooth function represents the distribution of material in the mesh cells and in that sense, it can be considered analogous to a density distribution function for the material. However, we should note that the volume fraction function is not a clearly defined mesh-independent continuous function like the density function. Swartz [24] describes it as the function that quantifies the relative amount of a material present in a small window that moves around in a domain with a sharp interface. Defined this way, it is clear that the volume fraction function steepens as the size of the window (or in other words, the mesh size) gets smaller and the gradient of the function blows up as the window size goes to zero. Nevertheless, for a given mesh, we will treat the volume fraction function like a smooth, density distribution function.

We then compute a piecewise linear approximation for this smooth volume fraction function using standard methods used in higher-order finite-volume methods [9]. Finally, continuing the analogy with the density function, we compute the center of mass of the materials in cells from the linear reconstruction as described below.

Consider a mesh on which we have cell-centered values f_i of a function $f(\mathbf{x})$. In each cell $\mathcal{C}_i$, we reconstruct a linear approximation, $\tilde{f}_i(\mathbf{x})$, of the function such that

$$\tilde{f}_i(\mathbf{x}) = f_i + \nabla f \cdot (\mathbf{x} - \mathbf{x}_c(\mathcal{C}_i)), \tag{2}$$

where $\mathbf{x}_c(\mathcal{C}_i)$ is the centroid of the cell. ∇f is the gradient of the function that we wish to approximate and it is considered to be constant within the cell. The gradient may be computed either by a Green–Gauss [9] or a least-squares technique [3]. On structured and unstructured grids, we use all vertex and edge connected neighbors in the gradient computation. In Fig. 6, the path used for the Green–Gauss technique is shown with the dotted line. For a least-squares technique, the same mesh cells are used in the computation with each entry weighted by the inverse of the squared distance between the centroid of the cell being reconstructed and the centroid of the neighboring cell as described in [3]. The computed gradient is limited using Barth–Jesperson-type limiter [25] to preserve local bounds on the volume fraction function. The limiter is calculated using all vertex connected neighbors. The limited gradient is indicated by $\boldsymbol{\delta} = \phi\nabla f$ with $\phi \in [0, 1]$. Then, the approximate center of mass of the function $f(\mathbf{x})$ over the domain Ω_i as approximated by the function $\tilde{f}_i(\mathbf{x})$ is given by

$$\bar{\mathbf{x}} = \frac{\int_{\Omega_i} \mathbf{x}\tilde{f}_i(\mathbf{x})d\Omega}{\int_{\Omega_i} \tilde{f}_i(\mathbf{x})d\Omega} = \frac{1}{\|\Omega_i\|\bar{f}_i}\int_{\Omega_i} \mathbf{x}(\bar{f}_i + \boldsymbol{\delta}\cdot(\mathbf{x} - \mathbf{x}_c(\mathcal{C}_i)))d\Omega \tag{3}$$

where $\|\Omega_i\|$ is the area of the domain Ω_i.

The obvious choice of domain Ω_i for integrating this equation is the cell, $\mathcal{C}_i$ and this works well for structured meshes. The calculation of Eq. (3) for a polygon may be done with the application of Stokes' theorem in the plane, for details see [26]. However, for unstructured meshes, we have found that integrating over the cell domain induces a strong bias in the orientation of the reconstructed interface based on the cell geometry. In order to eliminate this effect, we integrate instead over the smallest square, $S(\mathcal{C}_i) \supseteq \mathcal{C}_i$, whose center coincides with the centroid of the cell, $\mathbf{x}_c(\mathcal{C}_i)$ and encloses the computational cell. An example of this domain is shown with the solid line in Fig. 6.

For two materials, this choice of integration domain is equivalent to a gradient-based method when using a power diagram interface reconstruction. In a power diagram based reconstruction of a two material cell, the interface normal depends only on the direction of the vector pointing from one material locator to the other. For two materials, m and n, with material locators $\mathbf{x}_m$ and $\mathbf{x}_n$ and volume fractions f_m and $1 - f_m$, respectively, the normal to the interface between them given by the power diagram reconstruction will be

$$\mathbf{x}_m - \mathbf{x}_n = \frac{1}{\|S(\mathcal{C}_i)\|}\left(\frac{1}{\bar{f}_m} + \frac{1}{1-\bar{f}_m}\right)\int_{y_0}^{y_1}\int_{x_0}^{x_1}\mathbf{x}(\boldsymbol{\delta}\cdot(\mathbf{x} - \mathbf{x}_c))dx\,dy = \frac{\Delta^2}{12}\left(\frac{1}{\bar{f}_m(1-\bar{f}_m)}\right)\begin{pmatrix}\delta_x\\ \delta_y\end{pmatrix}$$

where $S(\Omega_i) = [x_0, x_1] \times [y_0, y_1]$ and $\Delta = x_1 - x_0 = y_1 - y_0 = \sqrt{\|S(\mathcal{C}_i)\|}$. That is, the normal is a positive constant times the gradient. Hence, the interface normal will be the computed gradient. In addition, this choice of integration domain makes the calculation of Eq. (3) trivial and provides a better initial reconstruction for a starting point to the interface smoothing procedures.

For meshes and simulations with special symmetries such as uniform-in-angle grids and problems with cylindrical symmetry [27], the choice of the square integration domain may break symmetry. In such cases, a different integration domain, such as the mesh cell itself, may be preferable.

3. Power diagram based interface reconstruction

Once the materials are located in a cell, the interfaces within the cell, separating the materials, are constructed using a power diagram. A power diagram or Laguerre diagram [28,29] is a generalized Voronoi diagram of a set of points, S, each with an associated weight, ω_i. In this context, the generators are the material locators determined by the method detailed in Section 2.

The power of a point $\mathbf{x} \in \mathbb{R}^n$ with respect to a point mass, $s_i \in S$ with $s_i = (\mathbf{x}_i, \omega_i)$ is defined as

$$\mathrm{pow}(\mathbf{x}, s_i) = d^2(\mathbf{x}, \mathbf{x}_i) - \omega_i^2 \tag{4}$$

where $d^2(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^{n}(x^j - x_i^j)^2$ is the usual Euclidean distance in $\mathbb{R}^n$.

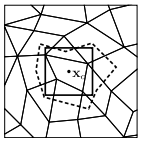


Fig. 6. The dotted line shows the path used for the Green–Gauss computation of the gradient in the center cell. The dark solid line shows the square domain, $S(\Omega_i)$, used for the material location calculation. It is the smallest square that covers the center cell and has the same centroid, $\mathbf{x}_c$, as the center cell.

Each cell of the power diagram is the set of points

$$\mathrm{cell}(s_i) = \{\mathbf{x} \in \mathbb{R}^n | \mathrm{pow}(\mathbf{x}, s_i) < \mathrm{pow}(\mathbf{x}, s_j) \forall s_j \in S, s_j \neq s_i\}. \tag{5}$$

As with Voronoi diagrams, each power diagram cell is convex with a piecewise linear boundary. If all the weights are equal, the power diagram degenerates to a Voronoi diagram. The weight associated with a point generator can be interpreted as the square of the radius of a circle centered at that point. An example power diagram is shown in Fig. 7.

A power diagram can be constructed by a randomized incremental algorithm [30] or by conversion from a Voronoi diagram [31] which may be created using a number of efficient algorithms such as Fortune's sweepline algorithm [32]. However, we use a simple half-space intersection algorithm because it is simple and robust for a small number of materials in each cell. The half-space intersection algorithm computes the power diagram cell for each generator by clipping the polygon by each bisector between that generator and all other generators, starting with the original computational cell.

3.1. Matching volume fractions

The volume fractions of the materials in a cell are matched by iteratively adjusting the weights of the point generators in a power diagram, thereby adjusting the area of the power diagram cells once clipped to the mesh cell. This requires the solution of a set of non-linear equations

$$A_m(\omega_1^2, \ldots, \omega_{N_m}^2) = \|\mathcal{C}_i\|f_m, \quad m = 1, \ldots, N_m \tag{6}$$

where $A_m(\omega_1^2, \ldots, \omega_{N_m}^2)$ is the area of the power diagram corresponding to material m after it has been clipped by the bounding mesh cell polygon with area $\|\mathcal{C}_i\|$. f_m is the volume fraction for material m in cell $\mathcal{C}_i$. The constraint:

$$\sum_{m}^{N_m} A_m(\omega_1^2, \ldots, \omega_n^2) = \|\mathcal{C}_i\| \tag{7}$$

reduces the number of equations to $N_m - 1$. Since for any power diagram

$$A_m(\omega_1^2 + c^2, \ldots, \omega_n^2 + c^2) = A_m(\omega_1^2, \ldots, \omega_n^2) \tag{8}$$

for any real number c, a normalized set of weights must be chosen. Specifically, this is done by forcing one of the weights to be a specified value and varying the others.

A Newton procedure with a finite-difference Jacobian is used to solve Eqs. (6) and (7). At each iteration of the Newton procedure, a power diagram, clipped to the mesh cell is computed to determine the A_m and approximate the Jacobian. Some caution is required, since the area of each cell is bounded above and below, that is

$$0 \leqslant A_m(\omega_1^2, \ldots, \omega_n^2) \leqslant \|\mathcal{C}_i\| m = 1, \ldots, N_m \tag{9}$$

For extreme values of the weights, some of the power diagram cells will be outside of the mesh cell and as such have zero area once clipped to the mesh cell, $\mathcal{C}_i$, under consideration. In such cases, one or more of $A_m(\omega_1^2, \ldots, \omega_n^2)$ will be constant with value 0 or $\|\mathcal{C}_i\|$ and as a function of the weights $(\omega_1^2, \ldots, \omega_n^2)$ it will have a zero gradient making a straightforward Newton procedure fail. As a result, the Newton procedure needs to adjust for overshoots to make sure it does not end up in this region. This is simply done by reducing the size of the Newton step at each iteration if it exceeds those bounds. We found the procedure to be robust and efficient, typically matching the required volumes to within 10^{-12} in 3–6 iterations.

For the initial guess, we use equal weights for all the point generators if all of the generators lie within the mesh cell being reconstructed. If any of the generators are outside the cell the initial weights are assigned such that the power bisectors be-

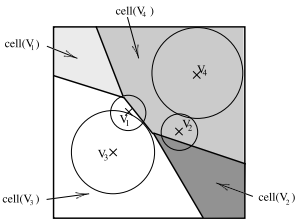


Fig. 7. Power diagram with four generators (V_1, V_2, V_3, and V_4) and their weight circles. The generators V_1 and V_2 do not lie within their corresponding cells.

tween all the generators go through the centroid of the cell. This ensures that the initial power diagram will have non-zero areas for each power diagram cell once clipped to the mesh cell.

4. Interface smoothing

Once an initial reconstruction of the interface is available, we apply an order-independent optimization-based smoothing procedure to improve the approximation of the interface.

Consider a 2D cell with N_m materials, N_s interface segments and N_p interface points. The smoothing procedure repositions the cell's N_p interface points so that it minimizes the discrepancy between the normal of each of its interface segments and normals of reference interface segments in neighboring cells (separating the same materials). The constraints imposed on this process are that the volume fractions of the materials in the cells must be matched exactly and that all the pure material subcells remain convex. Naturally, interface points on the boundary of the cell must remain on the boundary and interior points must remain strictly inside the cell.

The local objective function for smoothing in a particular cell, $\mathcal{C}_i$ is written as

$$F_i(\mathbf{s}) = \sum_{j=1}^{N_s}\sum_{k=1}^{(N_r)_j}\|\hat{\mathbf{n}}_j(\mathbf{s}) - \hat{\mathbf{n}}_k^r\|^2 \tag{10}$$

In the above equation, $\hat{\mathbf{n}}_j$ is the normal of the jth interface segment, $\hat{\mathbf{n}}_k^r$ is the normal of the kth reference segment, and finally, $(N_r)_j = 0$, 1 or 2 is the number of reference segments for the jth interface segment. Also, $\mathbf{s}$ is the vector of optimization variables which includes the x, y coordinates of interior interface points and 1D parameters of points on cell edges.

The reference segments for an interface segment are chosen according to the smoothing procedure laid out by Swartz [24] for smoothing two material interfaces. Given an interface segment in cell $\mathcal{C}_i$, we first find its endpoints that lie on the boundary of the cell. If the endpoint lies on a cell edge, we find the neighboring cell on the other side of the edge. In this neighboring cell, we attempt to find another interface segment that separates the same two materials as the interface segment in $\mathcal{C}_i$ that is under consideration. The reference normal for the interface segment is then the normal of the line segment connecting the midpoints of the two interface segments (see Fig. 8). Using this normal leads to a second-order accurate method for smooth interfaces in Swartz's work. If a neighboring cell does not exist (because the edge is on the domain boundary) or a suitable interface segment does not exist in the neighboring cell, we attempt to find a suitable reference segment in a cell connected to the vertices of the cell edge that the interface endpoint lies on. If the endpoint of an interface segment lies on a vertex, the reference interface segment is chosen from one of the vertex connected neighbors. Thus, the algorithm will pick two, one or zero reference segments for each interface segment. While it is possible that there may be more than two neighboring mixed cells with candidate reference segments, such a situation usually represents noisy volume fraction data or a feature of the flow that is not well resolved; in such situations, no perfect solution exists and therefore, any choice will probably lead to unsatisfactory results.

The formulation of the objective function is illustrated with the help of the examples shown in Fig. 8. The first example (Fig. 8(a)) shows a two-material cell of interest (shaded). Since there is only one interface segment with normal $\hat{\mathbf{n}}_1$ and its endpoints on the cell boundaries, this interface segment has two reference normals. Therefore, the objective function is

$$F_i(\mathbf{s}) = \|\hat{\mathbf{n}}_1(\mathbf{s}) - \hat{\mathbf{n}}_1^r\|^2 + \|\hat{\mathbf{n}}_1(\mathbf{s}) - \hat{\mathbf{n}}_2^r\|^2 \tag{11}$$

and the optimization variables are the 1D parameters of points $\mathbf{x}_1$ and $\mathbf{x}_2$. In the second example (Fig. 8(b)), there are three interface segments with normals $\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2$ and $\hat{\mathbf{n}}_3$ and endpoints $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and $\mathbf{x}_4$. Therefore, the objective function is

$$F_i(\mathbf{s}) = \|\hat{\mathbf{n}}_1(\mathbf{s}) - \hat{\mathbf{n}}_1^r\|^2 + \|\hat{\mathbf{n}}_2(\mathbf{s}) - \hat{\mathbf{n}}_2^r\|^2 + \|\hat{\mathbf{n}}_3(\mathbf{s}) - \hat{\mathbf{n}}_3^r\|^2 \tag{12}$$

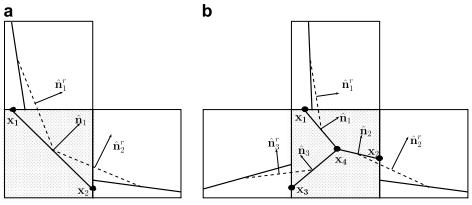


Fig. 8. Definition of reference normals for smoothing in (a) two-material cell and (b) three-material cell. The cell of interest is shaded.

the optimization variables are the 1D parameters of boundary points $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$, and the x, y coordinates of the interior point, $\mathbf{x}_4$.

The volume conservation constraint for the optimization can be expressed in the form of an equality constraint as

$$G_i(\mathbf{s}) = \frac{1}{\|\mathcal{C}_i\|^2}\sum_{m=1}^{N_m}(V_m(\mathbf{x}) - V_m^r)^2 = 0 \tag{13}$$

where $V_m(\mathbf{x})$ is the computed volume and V_m^r is the reference or specified volume of the mth material.

The convexity constraint on the subpolygons is expressed as a constraint on all interior angles of the subpolygons in the cell. The constraint function for convexity, $H_i(\mathbf{s})$ is a step function which is zero if all the interior angles of all material polygons in the cell are greater than 0 but less than or equal to π, and a very large penalty number (10^6) otherwise.

The optimization is then set up to minimize the composite function:

$$F_i^*(\mathbf{s}) = F_i(\mathbf{s}) + \lambda G_i(\mathbf{s}) + H_i(\mathbf{s}) \tag{14}$$

where λ is a penalty parameter. When the penalty parameter is sufficiently large, the constraints are enforced exactly [33].

The procedure minimizes $F_i^*(\mathbf{s})$ in each cell using a multi-dimensional non-linear conjugate gradient minimization procedure [33] with increasing values of the penalty parameters. Since the original reconstruction will satisfy the volume and convexity constraints, the penalty parameter λ is initially set to one, then incrementally increased by multiplying by 10 and smoothing using that value in the objective function until the penalty parameter reaches a desired value (typically 10^5).

For two materials cells the minimum of the objective functional in Eq. (14) may be constructed explicitly. It is the reconstruction corresponding to a single line with a normal taken by averaging the reference normals and matching the volume fraction. The multi-dimensional minimization procedure does not need to be invoked.

Multiple iterations ($N_{global} = 3$ to 5) over all mixed cells are performed so as to minimize the global objective function that can be formed by summing the F^*s over all mixed cells. The optimization converges to a correct solution quickly in all but a few very difficult cases for which convergence is impractically slow. The overall minimization algorithm is given in pseudocode in Fig. 9.

In Fig. 10, the initial power diagram reconstruction is shown along with the material locators for a three material, curved interface. After smoothing, the interface is much improved.

5. Numerical experiments

In this section, we present some static interface reconstruction examples and demonstrate second-order convergence. To demonstrate that our method leads to significant improvements in dynamic simulations, we present results of advection of multi-material configurations on structured and unstructured meshes.

5.1. Interface reconstruction

To demonstrate the performance of the method, two multi-material interface reconstructions are shown in Fig. 11. Both examples are on a domain of $[0, 1] \times [0, 1]$.

The first row in Fig. 11 shows the reconstruction of a triple point. In the center cell, the volume fractions are (0.5, 0.3, 0.2). In the top center and bottom center cells the volume fractions are (0.5, 0.0, 0.5) and (0.5, 0.5, 0.0), respectively. The right center cell has volume fractions of (0.0, 0.6, 0.4).

```
Material order independent smoothing
I^1 := power diagram reconstruction
For n = 1 to N_global
    For each mixed cell C_i
        construct F*(s^0) using I^n
        For k = 0 to 5
            λ := 10^k
            s^{k+1}(C_i) := minimizerF*(s^k; λ)
        End
        I^{n+1}(C_i) = interfaces(F*, s^K)
    End
End
```

Fig. 9. Material order independent interface smoothing algorithm. Here $\mathcal{I}^n$ is the interface reconstruction at global iteration n. $\mathbf{s}^k$ is the vector of optimization variables at each stage of the inner loop which increases the penalty parameter λ. *minimizerF*$^*(\mathbf{s}^k; \lambda)$ returns the variables at the local minimum starting from initial guess $\mathbf{s}^k$. The routine **interfaces**$(F^*, \mathbf{s})$ gives the polygonal decomposition of the cell corresponding to the values of the optimization variables, $\mathbf{s}$.

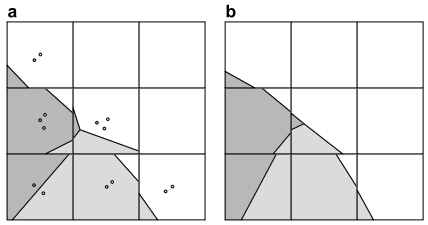


Fig. 10. (a) Interface reconstruction before smoothing showing material locators and (b) interface reconstruction after smoothing.

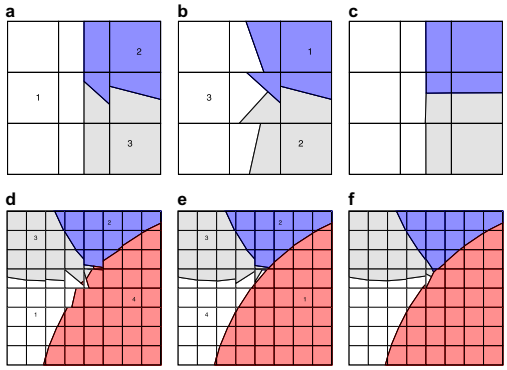


Fig. 11. Static interface reconstruction tests. (a) and (b) Triple point configuration using a nested dissection/gradient-based method with the material ordering indicated in the figure. (c) Triple point configuration using our second order method. (d) and (e) Three circle intersection test using nested dissection method with two different material orderings. (f) The interface reconstruction using our second order method. The convexity restrictions and initial topology of the power diagram prevent the optimization from completely smoothing the interface in the four material cells.

The second row of Fig. 11 shows the reconstruction of a three circle configuration containing two, three and four material cells. The volume fractions are defined by three circles given by (A) radius 0.5385, center (0.3, 1.0) (B) radius 0.7071, center (0.99, 0.97) and (C) radius 0.7071, center (0.99, 0.01) with circle (C) overlaying (B) which overlays (A).

The method correctly reconstructs the triple point in Fig. 11(c), although the white region in the center cell is only marginally convex due to the node at the center of the T-junction. Note in Fig. 11(d) and (e), the effect of multiple materials on a gradient-based method even in the two material cells near the four material center cell.

In Fig. 11(f), the method significantly improves the interface reconstruction in the four material and surrounding cells. However, due to the convexity constraints and the initial topology of the power diagram reconstruction the method is unable to perfectly match the interfaces in the four material cell.

5.2. Convergence

A number of tests have been conducted to demonstrate that the method converges and is second-order accurate. The error in each cell is measured as the symmetric area difference between the approximate and "true" reconstructions. Given two shapes Ω_1 and Ω_2, the symmetric area difference is measured as

$$e = (\|\Omega_1\| - \|\Omega_1 \cap \Omega_2\|) + (\|\Omega_2\| - \|\Omega_1 \cap \Omega_2\|) \tag{15}$$

The overall error, E, in the reconstruction is obtained by summing e over all materials and all cells. The symmetric area difference for a single cell is illustrated in Fig. 12.

The volume fraction initialization uses a sampling method based on adaptive mesh refinement. To compute the volume fractions of materials in a cell, the vertices of the cell are assigned material IDs based on an in-out test. If the vertices of the cell are of the same material, the cell is assigned that material, If not, the cell has multiple materials and is then subdivided into four equal subcells (assuming the cell is a regular quadrilateral). The new points generated by refinement are assigned material IDs based on further in-out tests. The refinement continues in this way until all subcells are pure or the size of a subcell is below the required volume fraction tolerance. The areas of subcells are then added up to compute volume fractions up to the desired tolerance.

Three multi-material examples where chosen for the convergence tests. The problem domain was $[0,1] \times [0,1]$ for all three tests.

- The first test (*circle*) shown in Fig. 13(a) consists of a circle of radius 0.25 with center (0.5, 0.5).
- The second test (*two-arcs*) shown in Fig. 13(b) is a three-material problem with two intersecting circular arcs. The bottom circle has a radius of 0.7 and center (0.5, 0.0). The circle on the right overlays the bottom circle and has radius of 0.5 and center (1.0, 0.5).
- The third test (*curved-filament*) is a curved filament with a varying thickness as shown in (Fig. 13(c)). It is composed of two circles, one with a radius of 0.7 and center (0.5, 0.0) which is partially covered by a circle of radius 0.64 and center (0.5, 0.05).

The coarsest mesh was a regular mesh of 32×32 and the finest mesh was 256×256, which had a characteristic mesh spacing of $h = \frac{1}{256}$.

The errors for these examples along with errors of an exactly second-order accurate scheme are plotted in Fig. 14. From this plot, it can be seen that the scheme reconstructs interfaces with second-order accuracy.

5.3. Advection

Interface reconstruction methods are generally applied in dynamic simulations where the materials, as described by the volume fractions, move over time. In simple, multiple passive scalar transport each material satisfies an advection equation,

$$\frac{d}{dt} f^m(\mathcal{C}_i) = -\frac{1}{\|\mathcal{C}_i\|} \int_{\partial \mathcal{C}_i} \chi_m(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{n} d\mathbf{x} \tag{16}$$

where $\chi_m(\mathbf{x})$ is the characteristic function for material m and $\mathbf{u}$ is the prescribed transport velocity and $\mathbf{n}(\mathbf{x})$ is the unit outward normal on the cell boundary. Furthermore, we assume the velocity field is divergence free and the materials are immiscible but with no surface tension.

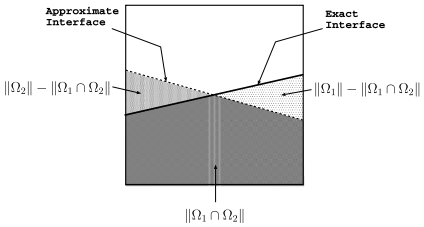


Fig. 12. Illustration of symmetric area difference with exact and approximate straight line interfaces. The darkly shaded part shows the area of overlap between the areas behind the two reconstructions and the remaining shaded parts show the area of non-overlap.

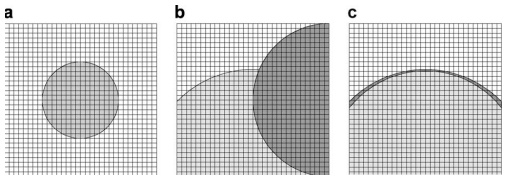


Fig. 13. Test cases used to study convergence – (a) two-material circular interface (*circle*) (b) three-material junction with circular arcs (*two-arcs*) and (c) varying thickness curved filament (*curved-filament*).

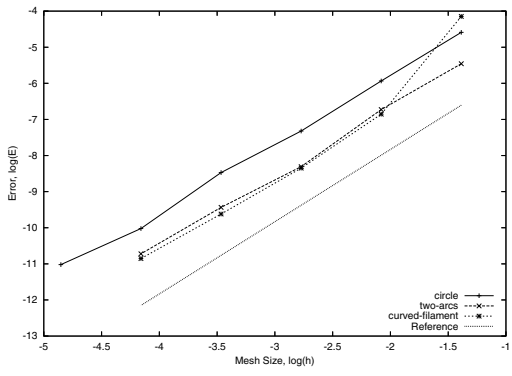


Fig. 14. Convergence plots for the three examples showing second-order accuracy. Here h is the mesh spacing and E the overall error.

A number of material advection schemes exist to simulate Eq. (16). For unstructured grids, an attractive advection scheme is the Lagrangian step plus Eulerian remap approach [34]. In this scheme, the advection step consists of a Lagrangian phase, where the mesh cell is moved in a Lagrangian fashion, followed by a conservative remap, where the cell contents from the Lagrangian step are mapped back to the Eulerian mesh. This approach has been suggested for material advection by a number of authors including [11,35–37].

There are two basic formulations of this method: a forward trajectory remap or a backward trajectory remap. In the forward trajectory remap, the position of the cell is calculated at time t^{n+1} and the interface reconstruction is performed using the updated cell positions. The material interfaces are then intersected with the original Cartesian mesh to determine the updated material volumes.

In the backward trajectory remap method, used in this work, the Eulerian mesh is taken as the cell positions at time t^{n+1}. The position of the cell is then integrated backwards in time to determine its location at the previous time step, t^n, which we refer to as the "preimage" of the cell. The interface reconstruction is performed using the volume fractions at time t^n on the original mesh. The preimage of the cell is then intersected with the interfaces on the Eulerian mesh. This process is illustrated in Fig. 15. The primary advantage of the backward trajectory remap, is that the interface reconstruction is always performed on the same, usually more regular, Eulerian mesh unlike the forward method where the interface reconstruction is performed on the Lagrangian mesh, which may consist of deformed cells even if the original Eulerian mesh was structured.

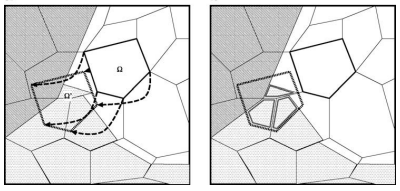


Fig. 15. Backward Lagrangian advection scheme. (a) The "preimage", Ω_f of the center computational cell of the fixed Eulerian mesh, obtained by integrating the nodes backward in time. (b) To determine the cell contents at the new time, the "preimage" is intersected with the interface reconstruction for each material at the previous time.

In addition, the computation of the gradient of the volume fraction function depends on cell geometry which is fixed for the backward scheme, but may change each timestep in the Lagrangian phase using the forward method.

In the first step of the backward trajectory remap method, the position of the cell vertices, $\mathbf{x}_i$, at the previous time step are determined by integrating the ODE backwards in time

$$\frac{d\mathbf{x}_i^n}{dt} = \mathbf{u}(\mathbf{x}, t) t \in [t^{n+1}, t^n] \tag{17}$$

with initial condition

$$\mathbf{x}_i^{n+1}(t^{n+1}) = \mathbf{x}_i \tag{18}$$

using the computed velocities from times t^n and t^{n+1}. We use a fourth order Runge–Kutta method, although many other time integrators are suitable.

Once the location of the vertices at the previous time step are calculated, the preimage of the cell is taken to be the polygon consisting of those vertices connected with straight lines. There are some errors introduced in computing the preimage in this manner. For a solenoidal velocity field ($\nabla \cdot \mathbf{u} = 0$), the area of the preimage will be the same as the area of the original cell. For a general velocity field, the preimage of a line will typically be a curve. Failing to account for the curvature of the edges of the preimage introduces a defect in the cell area estimated to be $\mathcal{O}(h^3 \Delta t)$ in addition to area defects associated with backward time integration of the vertices.

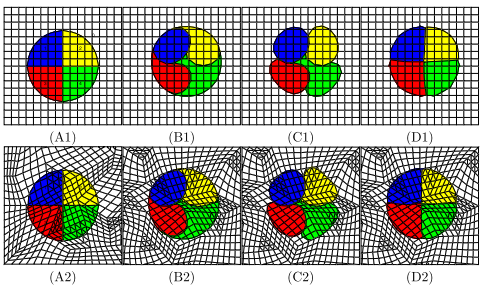


Fig. 16. Final configuration of the four material circle shown in (A1) with material numbers after diagonal translation with a velocity of $(1, 1)$ at time $t = 0.5$ using the interface reconstruction methods: (B1) Youngs' with material order (0, 1, 2, 3, 4) (C1) Youngs' with material order (1, 2, 3, 4, 0) (D1) our second order method. (A2)–(D2) show the same results on an unstructured mixed triangle and quadrilateral grid.

In the second step of the backward trajectory remap method, the preimage of the cell is intersected with the material polygons on the original mesh at time t^n as shown in Fig. 15(b). The areas of each material that lie within the preimage are added to obtain the material contents of the cell. Since the area of the preimage may not exactly match the area of the original cell, the volume fractions are taken with respect to the preimage area. Volume fractions less than 10^{-8} are suppressed.

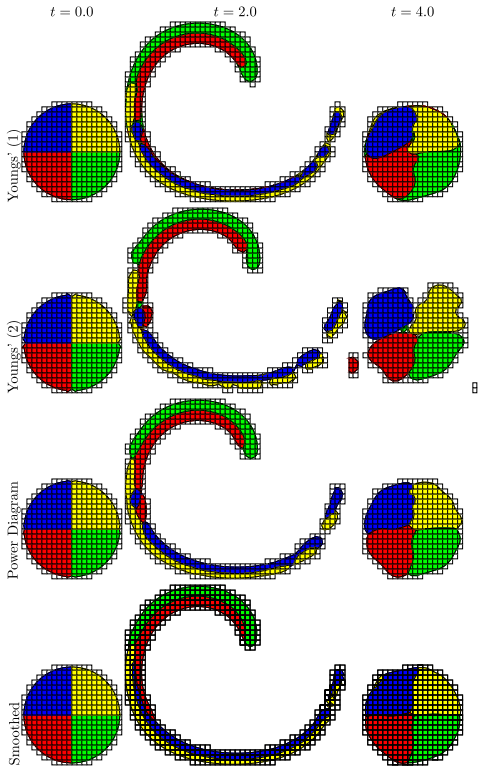


Fig. 17. Material interface configuration for the four material vortex test at maximum stretch time $t = 2.0$ and at complete reversal time $t = 4.0$ run on a 64×64 grid. For the method Youngs' (1), the material ordering was (0, 1, 2, 3, 4). For method Youngs' (2), the order was (1, 2, 3, 4, 0). The initial condition and material numbers are the same as the four material circle shown in Fig. 16 but with the circle center located at (0.5, 0.75).

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program.

References

[1] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (1992) 235.
[2] D.J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Applied Mechanics Review 55 (2) (2002) 151.
[3] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (1998) 112.
[4] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annual Review of Fluid Mechanics 31 (1999) 567.
[5] C.W. Hirt, B.D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201.
[6] W.F. Noh, P. Woodward, Slic (simple line interface calculation), in: A.I. van der Vooren, P.J. Zandbergen (Eds.), 5th International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, 1976, p. 330.
[7] D.L. Youngs, Time dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, New York, 1982, p. 273.
[8] D.L. Youngs, An Interface Tracking Method for a 3d Eulerian Hydrodynamics Code, Technical Report AWE/44/92/35, AWRE Design and Math Division, 1984.
[9] T.J. Barth, Aspects of unstructured grids and finite-volume solvers for Euler and Navier–Stokes equations, in: VKI/NASA/AGARD Special Course on Unstructured Grid Methods for Advection Dominated Flows, vol. R-787, AGARD Publication, 1995.
[10] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, Journal of Computational Physics 199 (2004) 465.
[11] S.J. Mosso, B.K. Swartz, D.B. Kothe, R.C. Ferrell, A parallel, volume-tracking algorithm for unstructured meshes, in: P. Schiano, A. Ecer, J. Periaux, N. Satofuka (Eds.), Parallel Computational Fluid Dynamics: Algorithms and Results Using Advanced Computers, Elsevier, 1997, p. 368.
[12] R.V. Garimella, V. Dyadechko, B.K. Swartz, M.J. Shaskov, Interface reconstruction in multi-fluid, multi-phase flow simulations, in: Proceedings of the 14th International Meshing Roundtable, Springer, San Diego, CA, 2005, pp. 19–32. September.
[13] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, Journal of Computational Physics 183 (2002) 400.
[14] J.E. Guilkey, T.B. Harman, B. Banerjee, An Eulerian–Lagrangian approach for simulating explosions of energetic devices, Computers and Structures 85 (2007) 660.
[15] B.Y. Choi, M. Bussmann, A piecewise linear approach to volume tracking a triple point, International Journal for Numerical Methods in Fluids 53 (2007) 1005.
[16] V. Dyadechko, M. Shashkov, Multi-material interface reconstruction from the moment data, Journal of Computational Physics (2008), doi:10.1016/j.jcp.2007.12.029.
[17] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Journal of Computational Physics 226 (2007) 2096.
[18] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubère, Material order independent interface reconstruction using power diagrams, International Journal for Numerical Methods in Fluids 56 (2008) 643.
[19] S. Mosso, S. Clancy, A Geometrically Derived Priority System for Young's Interface Reconstruction, Technical Report LA-CP-95-0081, Los Alamos National Laboratory, Los Alamos, NM, 1995.
[20] D.J. Benson, Eulerian finite element methods for the micromechanics of heterogeneous materials: dynamic prioritization of material interfaces, Computer Methods in Applied Mechanics and Engineering 151 (1998) 343–360.
[21] A. Caboussat, M.M. Francois, R. Glowinski, D.B. Kothe, J.M. Sicilian, A numerical method for interface reconstruction of triple points within a volume tracking algorithm. Mathematical and Computer Modelling, in press, doi:10.1016/j.mcm.2008.05.009.
[22] K.S. Bonnell, M.A. Duchaineau, D.R. Schikore, B. Hamann, K.I. Joy, Material interface reconstruction, IEEE Transactions on Visualization and Computer Graphics 9 (4) (2003) 500.
[23] A. Harten, ENO scheme with subcell resolution, Journal of Computational Physics 83 (1989) 148.
[24] B.K. Swartz, The second-order sharpening of blurred smooth borders, Mathematics of Computation 52 (186) (1989) 675.
[25] T.J. Barth, Numerical methods for gasdynamics systems on unstructured grids, in lecture notes in computational science and engineering, "an introduction to recent developments in theory and numerics for conservation laws", in: D. Kroner, M. Ohlberger, C. Rohde (Eds.), Proceedings of the International School on Theory and Numerics for Conservation Laws, Freiburg/Littenweiler, October, 20–24, 1997, Springer, Berlin, 1997, 195–285.
[26] S.F. Bockman, Generalizing the formula for areas of polygons to moments, The American Mathematical Monthly 92 (1989) 131.
[27] L. Margolin, M. Shashkov, Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics, Journal of Computational Physics 149 (1999) 389.
[28] F. Aurenhammer, Power diagrams: properties, algorithms and applications, SIAM Journal of Computing 16 (1987) 78.
[29] H. Imai, M. Iri, K. Murota, Voronoi diagram in the Laguerre geometry and its applications, SIAM Journal of Computing 14 (1985) 93.
[30] H. Edelsbrunner, N.R. Shah, Incremental topological flipping works for regular triangulations, Algorithmica 15 (1996) 223.
[31] F. Anton, D. Mioc, On the conversion of ordinary Voronoi diagrams into Laguerre diagrams, in: The Eleventh Canadian Conference on Computational Geometry, Vancouver, Canada, August 1999, pp. 150–153.
[32] S. Fortune, A sweepline algorithms for Voronoi diagrams, Algorithmica 2 (1987) 153.
[33] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, 1999.
[34] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, Journal of Computation Physics 160 (2000) 318.
[35] N. Ashgriz, T. Barbat, G. Wang, A computational Lagrangian–Eulerian advection remap for free surface flows, International Journal for Numerical Methods in Fluids 44 (2004) 1–32.
[36] P. Liovic, M. Rudman, J-L. Liow, D. Lakehal, D. Kothe, A 3D unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction, Computers and Fluids 35 (2006) 1011.
[37] K. Shahbazi, M. Paraschivoiu, J. Mostaghimi, Second order accurate volume tracking based on remapping for triangular grids, Journal of Computational Physics 188 (2003) 100.
[38] H.T. Ahn, M. Shashkov, Geometric algorithms for 3D interface reconstruction, in: M.L. Brewer, D. Marcum (Eds.), Proceedings of the 16th International Meshing Roundtable, Springer, 2008, pp. 405–422.

5.4. Example: diagonal translation of a four material disk

To illustrate the performance of our method in a dynamic example, we choose a simple, five material example that consists of diagonal translation of a four material disk at constant velocity. A four material disk of radius 0.15 is placed at (0.25, 0.25) on a mesh comprising the domain $[0, 1] \times [0, 1]$. The disk is then translated diagonally with a constant velocity of $\mathbf{u} = (1, 1)$ to a final time of $t = 0.5$. While the backward Lagrangian advection scheme has no restrictions on timestep for stability, here the timestep was restricted to emphasize the effect of repeated interface reconstructions. For all simulations, the CFL number, $\nu = \|\mathbf{u}\| \frac{\Delta t}{h}$, was $\frac{1}{8\sqrt{2}}$. The grid spacing, h, is taken as the square root of the area of the smallest grid cell.

In Fig. 16, the final configuration of the circle is shown using different interface reconstruction methods on a 32×32 structured grid and a comparable unstructured, mixed quadrilateral and triangular grid. The material order independent, second-order method (Fig. 16(D1) and (D2)) exhibits no grid artifacts and clearly preserves the structure better than the first order, material order dependent method (Fig. 16(B1),(C1), (B2) and (D2)).

5.5. Example: four material vortex test

A standard volume tracking test case is the vortex consisting of a circle of radius 0.15 centered at (0.5, 0.75) in a $[0, 1] \times [0, 1]$ domain. To demonstrate material ordering issues, we again use a four material circle. The incompressible velocity field is given by the streamfunction

$$\Psi = \cos\left(\frac{\pi}{4}t\right)\frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y) \tag{19}$$

with the velocity field defined to be $(u, v) = (-\frac{\partial \Psi}{\partial y}, \frac{\partial \Psi}{\partial x})$. At time $t = 4$, the material configuration should be identical to the initial condition at time $t = 0$.

Here the time step used was $\Delta t = \frac{1}{2}h$ where $h = \frac{1}{64}$ is the mesh spacing and the simulations were run to a final time of $t = 4.0$. The results are shown in Fig. 17. The material order issues with Youngs' method lead to excessive breakup of the interface. On the 64×64 mesh, the solution computed using the second-order method shows problems at the tail of the material. This is due to a poor initial reconstruction and the difficulty in obtaining reliable reference normals for the smoothing procedure when the material has broken up or is very poorly resolved. Overall, our method shows significantly better results than the material order dependent methods. In addition, the power diagram based reconstruction without the smoothing step also shows very good results.

The total change in material volume is defined as

$$e_{vol} = \sum_{m=1}^{N_m=5} \frac{\|V_m^{final} - V_m^0\|}{V_m^0} \tag{20}$$

where V_m^{final} is the total volume of material m after the final time step, and V_m^0 is the total initial volume of material m. For the methods shown, the change in material volume was for Youngs' method with order 01234, $e_{vol} = 2.54e-3$, for Youngs' method with order 12,340, $e_{vol} = 1.28e-3$, for the power diagram reconstruction, $e_{vol} = 1.35e-4$, and for the material order independent second-order method, $e_{vol} = 3.10e-3$. The volume discrepancy in the backward Lagrangian advection method is largely responsible for the error as all methods exhibit similar errors.

6. Conclusions

We have developed a second-order accurate method for material order independent interface reconstruction of multi-material cells. It is based on the determination of the relative locations of each material in a cell using linear reconstruction of the volume fraction function, reconstructing the interface using a power diagram and smoothing the interface with respect to its neighbors. The method does not assume a topology for the material regions, i.e. a layer structure or triple point configuration. Furthermore, all of the material regions created are convex which is a requirement for most advection and remapping schemes.

The extension of the smoothing algorithm to three dimensions is challenging. However, extending the material location by linear reconstruction is straightforward as it only relies on computation of the gradient. In addition, the construction of the power diagram to three dimensions only requires the ability to intersect polyhedrons with a halfspace, a procedure detailed in [38]. For the power diagram based reconstruction, the volume fraction matching procedure is dimension independent.

This reconstruction method is currently being implemented in multi-material flow codes to further understand the combined errors of advection and interface reconstruction.

Acknowledgments

The authors would like to thank B.K. Swartz for helpful discussions.

$\sim$

This chapter was devoted to the description of some of our investigations of the rezone and remap phases of and ALE code. Rezone has been extended to allow mesh reconnection and gave rise to the so-called ReALE method. Finally we have presented our contribution to interface reconstruction methods which are able to deal with more than two materials in multi-material fluid flows.

The next chapter presents some of our investigations which are not genuinely related to ALE *per se*. This chapter deals with very high-order MOOD schemes, kinetic schemes and interface reconstruction technique in an Eulerian context.

Other more or less related investigations

3

Contents

In this chapter we present some topics that we have treated which are more or less related to Lagrangian and ALE numerical schemes. I came across these projects thanks to colleagues and collaborators, and I would like to take the opportunity to thank them for feeding me with their idea. An exhaustive description of the context, the existing methods and the details of our approaches is available in the published papers and are not rephrased here. Instead we justify why these topics of research have been initiated and emphasize some difficulties that have been resolved and also some others which are still to be solved. Moreover several numerical results from our papers will be reproduced in order to show how and why these investigations led to numerical methods and simulation codes that are of interest for the community.

More precisely this chapter is organized in the following sections :

A section devoted to the Multi-dimensional Optimal Order Detection (MOOD) method. This method has been developed during the three years of a PhD (2009-2012) S. Diot shared with S. Clain (university Do Minho, Guimaraes Portugal). The Multidimensional Optimal Order Detection (MOOD) method which has been developed and improved in set of three successive papers [29, 32, 33] and also a set of proceedings [30, 31]. This method is a very high order finite volume method based on polynomial reconstruction based on *a posteriori* polynomial degree decrementing which plays the role of a classical limitation. This provides a different manner of considering how, where and when limitation is needed.

A section dedicated to the presentation of an ultra Fast Kinetic Scheme (FKS) [39]. This work has been done with G. Dimarco (IMT) following one of his idea. This consists of a new ultra efficient numerical method for solving kinetic equations in the case of the BGK relaxation operator. The scheme is based on a splitting technique between transport and collision. The key

idea is to solve the collision part on a grid and then to solve exactly the transport linear part by following the characteristics backward in time. The main difference between the method proposed and semi-Lagrangian methods is that here we do not need to reconstruct the distribution function at each time step. This allows to tremendously reduce the computational cost of the method and it permits to compute solutions of full six dimensional kinetic equations on a single processor laptop machine.

A section devoted to the interface reconstruction technique within the Finite Volume with Characteristic Flux scheme [210]. This scheme is an Eulerian finite volume scheme based on characteristic decomposition of the flux between neighbor cells. A two material extension has been proposed by J.-P Braeunig *et al* in [211, 212], this extension employs a VOF approach with a SLIC interface reconstruction method. However we have shown that some inconsistency in the interface reconstruction method led to poor advection results. Our contribution proposed an improvement of this so-called Natural Interface Positioning (NIP) method [211, 212]. Our modification, called Enhanced Natural Interface Positioning has been published in [18] and its extension to deal with more than two materials is to be published in a forthcoming paper [41].

3.1 Very high order finite volume scheme : The Multidimensional Optimal Order Detection method (MOOD)

In this work we solve the advection equation and the Euler system of hydrodynamics with a method built on the basics of the Finite Volume (FV) numerical scheme. FV method considers piecewise constant values of the variables per cell (*i.e* mean values) and computes their evolution in time. As already known FV method using constant states is only first-order accurate in space. Higher order accurate methods can be obtained as instance using polynomial reconstruction using mean values to evaluate more accurately the flux. Unfortunately some sort of limitation is needed to avoid spurious oscillations near discontinuous profiles (shock wave or contact discontinuity as instance). Close to discontinuities any stable scheme must degenerate to an at most first-order accurate one.

3.1.1 MOOD key idea : "*a posteriori*" limitation

Classical high order polynomial reconstruction schemes such as the Monotonic Upstream-centered Schemes for Conservation Laws (MUSCL) based on Godunov approach and introduced by van Leer [213], or various Essentially Non-Oscillatory schemes (ENO) proposed by Harten, Osher and Shu [214, 215, 216], are based on an *a priori* limiting procedure to achieve some stability property. In MUSCL like methods unlimited slopes are reduced through the use of a slope limiter whereas the least oscillating polynomial is chosen for ENO/WENO like methods. There is a vast literature about slope/flux limiters, some of them are now known after their discoverers (van Leer [217, 218, 213], van Albada [219], Sweby [220], Barth-Jespersen [221], Venkatakrishnan [222], Koren [223], etc.) or their particularity (minmod [224], superbee [224], monotonized central [218], etc.). Finding the Essentially Non-Oscillating polynomial (hence the name of the method (W)ENO), improving the choice of reconstruction stencils and reducing the possible huge number of stencils have also led to a considerable sum of articles see as instance [214, 215, 216], more specifically [225] and the bibliography herein.
In any case these types of limitation are performed *a priori* by a clever analyze of the available data. This implies that the "worst case scenario" must always be considered as plausible, and, as a consequence, the "precautionary principle" applies. In other words because scientific investigation has found a plausible risk of instability development, *a priori* limitations strike more often and harder than necessary. We believe that these limitations can be relaxed only if further information emerges that provides evidence that no harm will result when using unlimited reconstruction. One way to attain this goal is to check *a posteriori* if a solution has failed to fulfill some stability criteria [1]
The principles of the MOOD method are as simple as : first compute a candidate solution without any limitation, then detect if this solution locally fails to fulfill some stability criteria (problematic regions) and further uses limiter only on problematic regions to recompute the new candidate solution. The new candidate solution is then checked again for eligibility. The MOOD method follows this fundamentally different way. A maximal polynomial degree is set. Then a polynomial degree reduction plays the role of *a posteriori* limitation. An iterative procedure which decrements polynomial degree in problematic regions provides the optimal local polynomial reconstruction which satisfies given stability criteria.
Doing so we can ensure the positivity of the scheme by construction if the lowest order scheme is. Moreover we ensure that the numerical solution obtained is one of the most accurate solution achievable because every higher order polynomial reconstructions have been tested but the first

1. Treating *a posteriori* if a solution is valid is not new and can be found in the context of remapping methods (decreasing of polynomial order in [226], repair methods [141, 142, 8, 9]) and presumably in many other areas.

reconstruction leading to an acceptable solution [2].
In [29] we introduced this MOOD concept which provides up to third-order approximations to hyperbolic scalar or vectorial solutions for two-dimensional geometry. Then in [32] we have extended to general unstructured 2D meshes and to sixth-order convergence in space. Finally in [33] the 3D version has been deployed. We refer the reader to these papers to an exhaustive description of the MOOD method.

3.1.2 MOOD performances in 1D, 2D and 3D

The MOOD method has been entirely developed and extensively tested by S. Diot with his 2D and 3D codes. It has led to three publications [29, 32] and [33]. For the first paper [29] we have presented the method and the associated concepts of cell and edge polynomial degree. An effective third order of accuracy of the MOOD method on advection equation on irregular structured grid has been achieved. Then on Euler equations we have shown that the MOOD method with piecewise parabolic or linear reconstructions is nicely performing on classical test cases (Sod shock tube, four state Riemann problem, Mach 3 step problem, double Mach reflection). The MOOD method is also favorably compared to classical Finite Volume and MUSCL like methods.
The second publication [32] introduces the extension of MOOD to 2D unstructured meshes with higher order polynomials (up to six). In this paper we have shown that the expected high order of convergence is reached both for advection and on smooth solutions of the Euler equations. To reach the sixth-order of accuracy for a $\mathbb{P}_5$ polynomial reconstruction we have relaxed the strict discrete maximum principle which is a cause of limitation to second-order of accuracy along with the use of non-conservative variable reconstructions, see also [227] on this point. A detector of smooth solution has been designed, it avoids polynomial degree decrementing and ensures a high order of accuracy on smooth profiles. The method has been tested on unstructured and non-conformal meshes. For the advection section we have tested the MOOD method on a smooth solution (double sine translation) and on discontinuous profiles (solid body rotation). For the Euler equations we have considered and isentropic vortex problem which admits a smooth exact solution. The MOOD method can effectively produce the optimal order of accuracy, up to sixth-order for a $\mathbb{P}_5$ polynomial reconstruction. The 1D Lax shock tube has further been run to compare MOOD with classical WENO method. The double Mach problem has been used to assess the low storage and speed-up of the MOOD method on three different single core machines. We finally have tested the ability of MOOD method to capture physics in realistic conditions by simulating the experiment proposed in where a planar shock impacts a cylindrical cavity, see Figs. 3.1.
The last paper deals with the 3D version of the method on unstructured grid. This paper is meant to prove that the MOOD method can be developed in 3D on single core machines. The method has been extensively tested on different machines to estimate its actual cost. On advection equation we have shown that the method can reach effective high order of accuracy. For Euler equations we have designed a 3D extension of the impact of a shock wave on a cylindrical cavity. A spherical version of the Sod shock tube has also been simulated. Last the interaction of a shock wave with a quarter of cone has been run with the fourth order accurate MOOD method. This is a 3D extension of the so-called "interaction of a shock wave with a wedge" problem. We reproduce in Fig. 3.2 the figure from the paper where the mesh (colored by the cell volume) and the principal waves are shown.

2. defined as a solution fulfilling user-given stability criteria such as physical admissibility of the solution, positivity of some variables, non-oscillatory solution, etc.

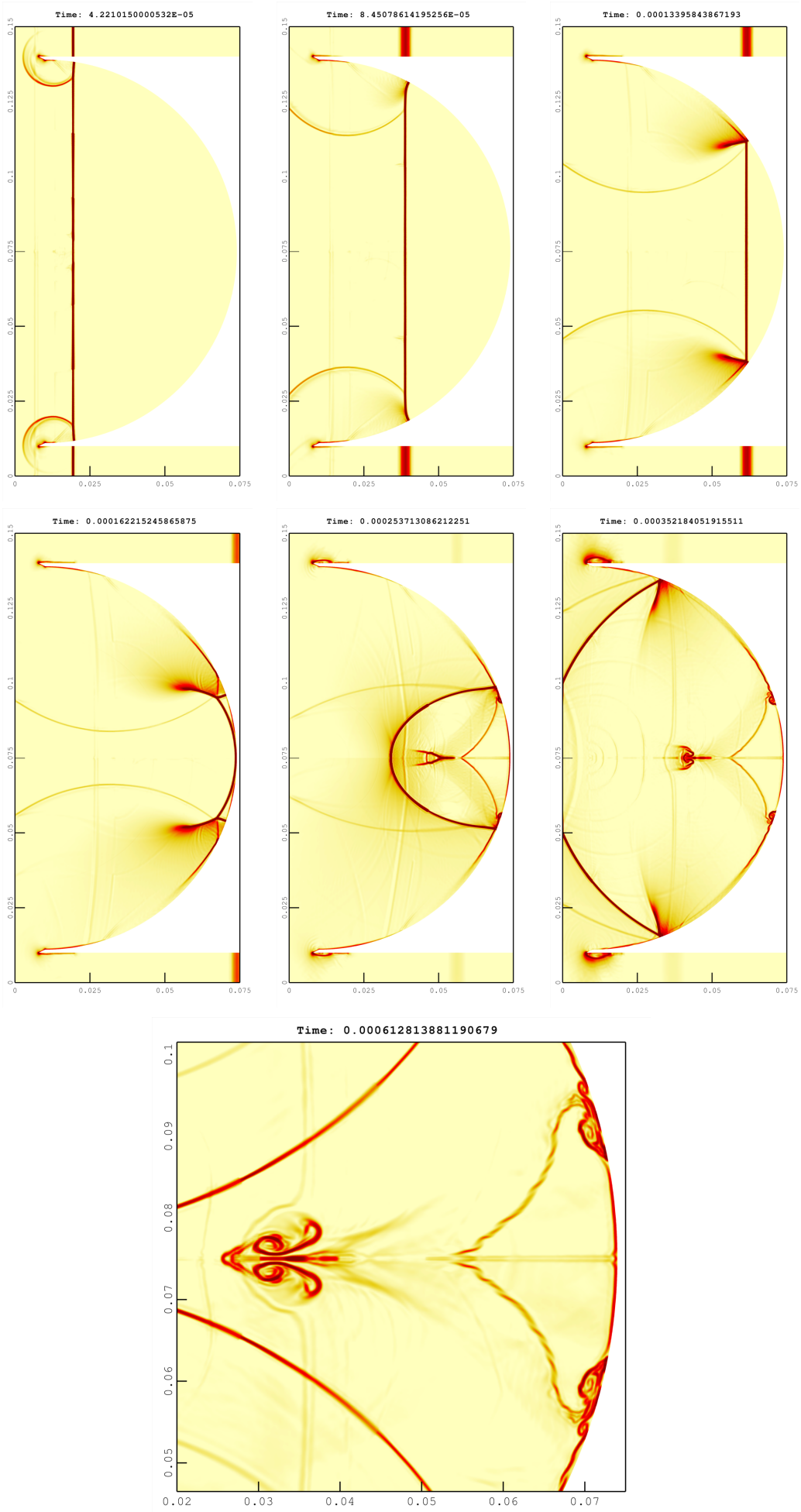


FIGURE 3.1 – *Results of the 2D MOOD-$\mathbb{P}_2$ method on the impact of a shock wave on a cylindrical cavity. Gradient*

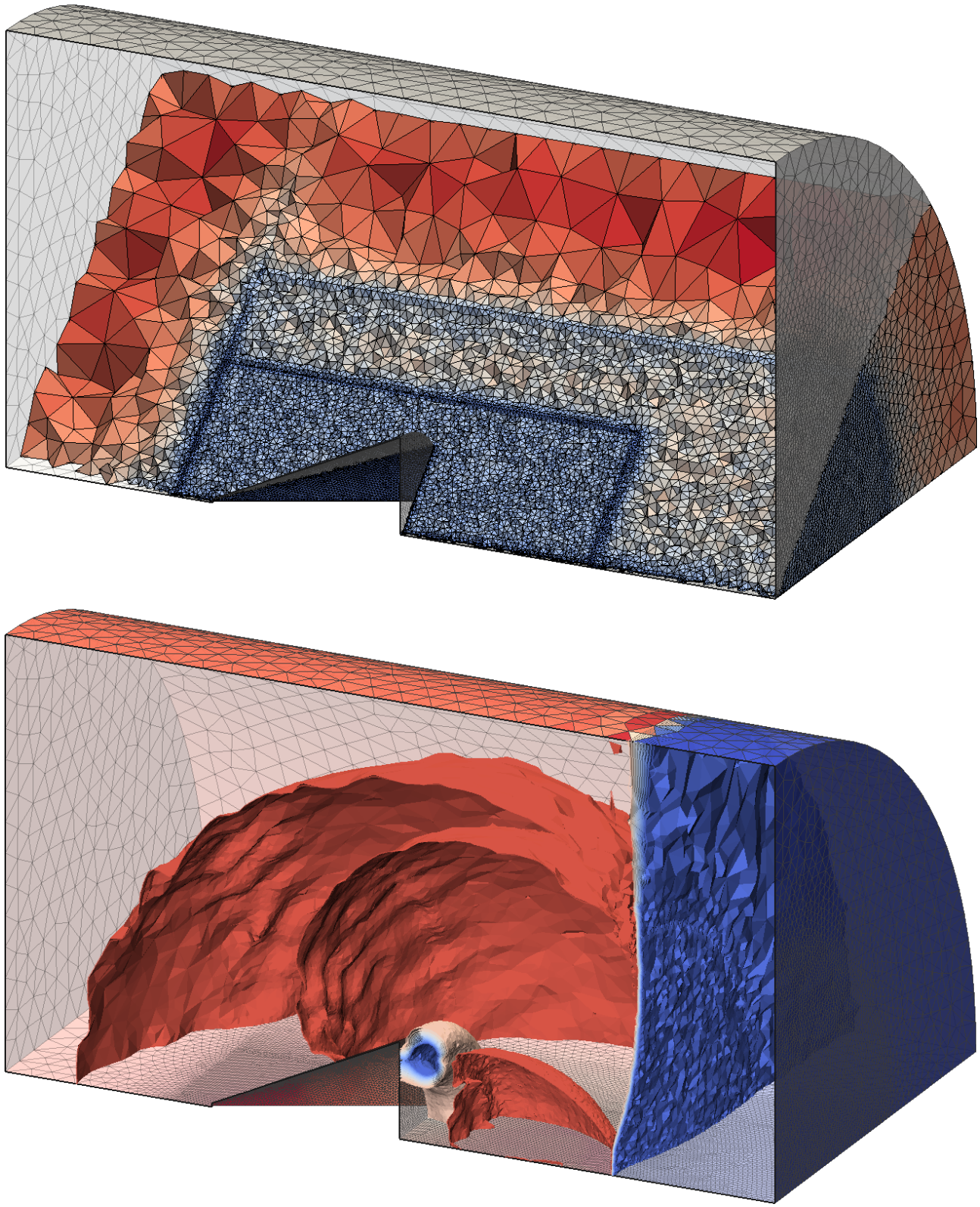


Figure 3.2 – *Results of the 3D MOOD-$\mathbb{P}_3$ method on interaction of a shock wave with a half cone. Top panel : view of the interior of the tetrahedral mesh with the different zones of refinement. Bottom panel : isosurfaces corresponding to the principal waves.*

Journal of Computational Physics 230 (2011) 4028–4050

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A high-order finite volume method for systems of conservation laws—Multi-dimensional Optimal Order Detection (MOOD)

S. Clain [a,b], S. Diot [a,*], R. Loubère [a]

[a] Institut de Mathématiques de Toulouse, Université de Toulouse, France
[b] Departamento de Matemática e Aplicações Campus de Gualtar, 4710-057 Braga Campus de Azurém, 4800-058 Guimarães, Portugal

ARTICLE INFO

Article history:
Received 31 August 2010
Received in revised form 15 February 2011
Accepted 17 February 2011
Available online 24 February 2011

Keywords:
Finite volume
High-order
Conservation law
MUSCL
Polynomial reconstruction
Limitation
MOOD

ABSTRACT

In this paper, we investigate an original way to deal with the problems generated by the limitation process of high-order finite volume methods based on polynomial reconstructions. Multi-dimensional Optimal Order Detection (MOOD) breaks away from classical limitations employed in high-order methods. The proposed method consists of detecting problematic situations after each time update of the solution and of reducing the local polynomial degree before recomputing the solution. As multi-dimensional MUSCL methods, the concept is simple and independent of mesh structure. Moreover MOOD is able to take physical constraints such as density and pressure positivity into account through an "a posteriori" detection. Numerical results on classical and demanding test cases for advection and Euler system are presented on quadrangular meshes to support the promising potential of this approach.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

High-order methods for systems of nonlinear conservation laws are an important challenging question with a wide range of applications. Furthermore in an engineering context such methods may deal with complex multi-dimensional domains requiring unstructured, heterogeneous or even non-conformal meshes. To handle highly stretched unstructured meshes made with different cell shapes, one has to design genuinely multi-dimensional numerical methods which exclude dimensional splitting techniques.

Due to its simplicity (one unknown mean value per cell) and built-in conservativity property, first-order finite volume method is very popular in today's engineering applications or commercial codes. However, it suffers from a major drawback, namely the presence of a large amount of numerical diffusion leading to a poor accuracy and over smoothed discontinuities. High-order space and time finite volume methods based on local polynomial reconstructions and Runge–Kunta algorithm have been developed to improve the approximation accuracy. MUSCL methods are probably the most popular second-order finite volume schemes. First developed in the one-dimensional situation with linear reconstructions [14,27,15,16], the technique has been extended to genuinely multi-dimensional case using structured or unstructured meshes [3,2,12,20,7,21,4]. Stability is achieved using a limiting procedure based on the Maximum Principle. In the present study, the Multi-dimensional Limiting Process (MLP) of [20,21] is employed since it is one of the most up-to-date MUSCL methods. Besides, (Weighted) Essentially Non Oscillatory polynomial reconstruction procedures (ENO/WENO) were designed to reach higher-order of accuracy [10,11,1,24,23] using less restrictive conditions for the limitation which do not guarantee a strict Maximum Principle for scalar problems. Moreover, although ENO/WENO schemes can retain high-order spatial accuracy even at points of extrema, extra difficulties and complexities have to be faced for the implementation on multi-dimensional unstructured grids (see [1,28]) as a large number of stencils for the polynomial reconstructions must be proceeded. Such drawbacks lead us to put ENO/WENO methods aside from the present study.

In this work we propose a genuinely multi-dimensional high-order method within a finite volume Eulerian framework on non-uniform meshes, the Multi-dimensional Optimal Order Detection (MOOD) method. In contrast to the traditional methods which use an *a priori* limitation procedure, the MOOD technique is based on an *a posteriori* detection of problematic cells. In each cell optimal polynomial degrees are determined to build approximated states leading to a discrete maximum principle preserving solution. In an hydrodynamics context, physical properties such as the density and the pressure positivity are considered. Roughly speaking, the polynomial degree may drop to zero in the vicinity of discontinuities leading to a local stable first-order finite volume scheme whereas high-order scheme is achieved in smooth regions. As for other methods, the MOOD method is embedded into the sub-steps of a high-order Runge–Kutta time discretization.

The paper is organized as follows, Section 2 is dedicated to the generic framework used to describe the MOOD method. Section 3 is devoted to the linear reconstruction and to a short presentation of the MLP method [20,21]. The MOOD method for scalar problems is detailed in the fourth section while Section 5 is dedicated to an extension of MOOD method to the Euler equations. At last, the numerical results for the advection and the Euler equations problems are respectively gathered in Sections 6 and 7. Classical tests are carried out and comparisons to the results of MLP method are provided. Several numerical examples prove the efficiency of the MOOD method in its second- and third-order version. The last section finally gathers conclusion and perspectives.

2. General framework

We consider the generic scalar hyperbolic equation defined on a domain $\Omega \subset \mathbb{R}^2$, $t > 0$ cast in the conservative form

$$\partial_t u + \nabla \cdot F(u) = 0, \tag{1a}$$
$$u(\cdot,0) = u_0, \tag{1b}$$

where $u = u(\mathbf{x},t)$ is the unknown function, $\mathbf{x} = (x_1,x_2)$ denotes a point of Ω and t the time. F is the physical flux and u_0 is the initial condition. Boundary conditions shall be prescribed in the following.

To elaborate the discretization in space and time, we introduce the following ingredients. We assume that the computation domain Ω is a polygonal bounded set of $\mathbb{R}^2$ divided into quadrangles K_i, $i \in \mathcal{E}_{el}$ where $\mathcal{E}_{el}$ is the cell index set with $\mathbf{c_i}$ being the cell centroid. For each cell K_i, $\lambda(i)$ is the set of all the nodes P_m, $m \in \lambda(i)$ while e_{ij} denotes the common edge between K_i and K_j with $j \in \underline{\nu}(i)$, $\underline{\nu}(i)$ being the index set of all the elements which share a common side with K_i. Moreover, $\overline{\nu}(i)$ represents the index set of all K_j such that $K_i \cap K_j \neq \emptyset$ (see Fig. 1). At last, $|K_i|$ and $|e_{ij}|$ measure the surface of K_i and the length of e_{ij}, respectively, and $\mathbf{n}_{ij}$ is the unit outward normal vector of K_i.

To compute an approximation of the solution of Eq. (1), we recall the generic first-order explicit finite volume scheme

$$u_i^{n+1} = u_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \mathbb{F}\left(u_i^n, u_j^n, \mathbf{n}_{ij}\right), \tag{2}$$

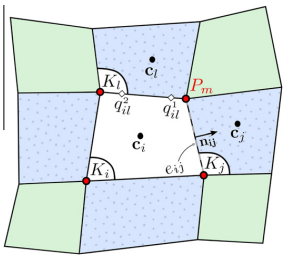


Fig. 1. Mesh notation. K_i is a generic element with the centroid $\mathbf{c_i}$. Index set $\underline{\nu}(i)$ corresponds to blue cells with dots, $\overline{\nu}(i)$ corresponds to every non-white cells and $\lambda(i)$ is the set of red P_m node indexes. Edges are denoted by e_{ij} with $\mathbf{n}_{ij}$ the unit outward normal vector of element K_i. Numerical integration on edge e_{ij} is performed with the two Gauss points q_{ij}^1, q_{ij}^2. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

* Corresponding author.
E-mail addresses: stephane.clain@math.univ-toulouse.fr (S. Clain), steven.diot@math.univ-toulouse.fr (S. Diot), raphael.loubere@math.univ-toulouse.fr (R. Loubère).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.02.026

where $\mathbb{F}(u_i^n, u_j^n, \mathbf{n}_{ij})$ is a numerical flux which satisfies the classical properties of consistency and monotonicity.

Unfortunately, such a scheme only provides first-order accuracy in space and higher-order reconstruction techniques are used to improve the solution approximation. To this end, we substitute in Eq. (2) the first-order approximation u_i^n and u_j^n with better approximations of u on the e_{ij} edge and consider the generic spatial high-order finite volume scheme

$$u_i^{n+1} = u_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}(u_{ij,r}^n, u_{ji,r}^n, \mathbf{n}_{ij}), \tag{3}$$

where $u_{ij,r}^n$ and $u_{ji,r}^n$, $r = 1,\ldots,R$ are high-order representations of u on both sides of edge e_{ij} and ξ_r denote the quadrature weights for the numerical integration. In practice, $u_{ij,r}^n$ and $u_{ji,r}^n$ are two approximations of $u(q_{ij}^r, t^n)$ at quadrature points $q_{ij}^r \in e_{ij}$, $r = 1,\ldots,R$ (see Fig. 1).

For the sake of simplicity, let us write the scheme under the compact form

$$u_h^{n+1} = u_h^n + \Delta t \mathcal{H}^R(u_h^n) \tag{4}$$

with $u_h^n = \sum_{k \in \mathcal{E}_{el}} u_k^n \mathbb{1}_{K_k}$ the constant piecewise approximation of function u and operator $\mathcal{H}^R$ being defined as

$$\mathcal{H}^R(u_h^n) := \sum_{i \in \mathcal{E}_{el}} \left(- \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}\left(u_{ij,r}^n, u_{ji,r}^n, \mathbf{n}_{ij}\right) \right) \mathbb{1}_{K_i}. \tag{5}$$

To provide a high-order method in time, we use the third-order TVD Runge–Kutta method (see [24]) which corresponds to a convex combination of three explicit steps

$$u_h^{(1)} = u_h^n + \Delta t \mathcal{H}^R(u_h^n), \tag{6a}$$
$$u_h^{(2)} = u_h^{(1)} + \Delta t \mathcal{H}^R(u_h^{(1)}), \tag{6b}$$
$$u_h^{(3)} = \left(\frac{3u_h^n + u_h^{(2)}}{4}\right) + \Delta t \mathcal{H}^R\left(\frac{3u_h^n + u_h^{(2)}}{4}\right), \tag{6c}$$
$$u_h^{n+1} = \frac{u_h^n + 2u_h^{(3)}}{3}. \tag{6d}$$

Remark 1. Note that a high-order scheme in space and time can be rewritten as convex combinations of the first-order scheme. From a practical point of view, implementation of the high-order scheme from an initial first-order scheme is then straightforward.

The main challenge is to build the approximations $u_{ij,r}^n$ and $u_{ji,r}^n$ on both sides of edge e_{ij} with $r = 1,\ldots,R$ to be plugged into relations (5) and (6). Polynomial reconstructions provide high-order approximations but unphysical oscillations arise in the vicinity of discontinuities. Indeed, the exact solution of an autonomous scalar conservation law (1) satisfies a local Maximum Principle and we intend to build the reconstructions such that this stability property is fulfilled at the numerical level (see [5,6] and references herein). To this end, we state the following definition.

Definition 2. A numerical scheme (4) satisfies the Discrete Maximum Principle (DMP) if for any cell index $i \in \mathcal{E}_{el}$ one has

$$\min_{j \in \overline{\nu}(i)}(u_i^n, u_j^n) \leqslant u_i^{n+1} \leqslant \max_{j \in \overline{\nu}(i)}(u_i^n, u_j^n). \tag{7}$$

3. A short review on a multi-dimensional MUSCL method

All L^∞ stable second-order schemes are based on piecewise linear reconstructions equipped with a limiting procedure. The polynomial reconstruction provides the accuracy while the limitation algorithm ensures the physical relevancy of the numerical approximation. We briefly present the piecewise linear reconstruction step and recall the MLP method proposed in [21] which is used in the numerical part of this paper.

3.1. Linear reconstruction

Let $(u_i)_{i \in \mathcal{E}_{el}}$ be a set of cell centered mean values given on cells K_i. In order to simplify notations, let K be a generic cell with centroid $\mathbf{c} = (c_1,c_2)$. Considering mean values on a chosen neighborhood made of cells K_j, $j \in \nu$, we seek a polynomial function $\tilde{u}(\mathbf{x})$ of degree $d = 1$. Let us define the notation for the mean value as

$$\langle \tilde{u}(\mathbf{x}) \rangle_K \overset{\text{def}}{=} \frac{1}{|K|} \int_K \tilde{u}(\mathbf{x}) d\mathbf{x}.$$

Usually we ask for the following criteria

Criterion 3. The polynomial reconstruction $\tilde{u}$ must fulfill

(1) $\langle \tilde{u}(\mathbf{x}) \rangle_K = \bar{u}$ where $\bar{u}$ is the mean value approximation of u on K.
(2) The polynomial coefficients are the ones minimizing the functional

$$E(\tilde{u}) = \sum_{j \in \nu} \left(u_j - \langle \tilde{u}(\mathbf{x}) \rangle_{K_j}\right)^2, \tag{8}$$

A classic way to write $\tilde{u}$ is

$$\tilde{u}(\mathbf{x}) = \bar{u} + G \cdot (\mathbf{x} - \mathbf{c}), \tag{9}$$

where $G = (G_1, G_2)$ is a constant approximation of ∇u on K. The first condition of Criterion 3 is directly satisfied and classical techniques like least squares methods are used to determine vector G that minimizes the functional E in Eq. (8).

3.2. Gradient limitation

As we mentioned above, a finite volume scheme only based on a local polynomial reconstruction without limiting procedure produces spurious oscillations. Initiated by the pioneer works of Kolgan and Van Leer [14,15,27,16], the MUSCL technique deals with a local linear reconstruction like (9) on each cell K where the gradient G is reduced by a limiter coefficient $\phi \in [0,1]$

$$\tilde{u}(\mathbf{x}) = \bar{u} + \phi(G \cdot (\mathbf{x} - \mathbf{c})), \tag{10}$$

such that any reconstructed values satisfy the Discrete Maximum Principle (see [2,3,12]). We choose to detail and use the MLP limiter instead of the classical Barth–Jespersen limiter because it provides more accurate results (see [21]). The MLP limiter applies the following procedure:

- Construction of an unlimited slope G using the neighbor cells $K_j, j \in \overline{\nu}$.
- Evaluation of the unlimited reconstruction (9) at the vertices P_m of $K : u_m = \tilde{u}(P_m), m \in \lambda$ the nodes index set of K.
- Evaluation of the bounds for each node P_m

$$\delta u_m^{\max} = \max_{j, P_m \in \lambda(j)} (u_j - \bar{u}), \quad \delta u_m^{\min} = \min_{j, P_m \in \lambda(j)} (u_j - \bar{u}).$$

- Evaluation of the vertex based limiter ϕ_m

$$\phi_m = \begin{cases} \min\left(1, \frac{\delta u_m^{\max}}{u_m - \bar{u}}\right) & \text{if } u_m - \bar{u} > 0, \\ \min\left(1, \frac{\delta u_m^{\min}}{u_m - \bar{u}}\right) & \text{if } u_m - \bar{u} < 0, \\ 1 & \text{if } u_m - \bar{u} = 0. \end{cases}$$

- Cell-centered limiter $\phi = \min_{m \in \lambda} \phi_m$.

The MLP technique provides a second-order finite volume scheme which satisfies the Discrete Maximum Principle under a more restrictive CFL condition than the CFL condition of the first-order scheme.

Remark 4. Although there exists a large literature about piecewise linear limitation, the extension of MUSCL type methods to piecewise quadratic or even higher degree polynomials in a multi-dimensional context is not yet achieved. An efficient limitation process is still an under-investigation field of research.

4. The Multi-dimensional Optimal Order Detection method (MOOD)

Classical high-order methods are based on an *a priori* limitation of the reconstructed values which are plugged into a one time step generic finite volume scheme to update the mean values (see Fig. 2 top).

Unlike existing methods, the MOOD technique proceeds with an *a posteriori* limitation. Over each cell, an unlimited polynomial reconstruction is carried out to build a prediction $u_h^\star$ of the updated solution. Then the *a posteriori* limitation consists of reducing the polynomial degree and recomputing the predicted solution $u_h^\star$ until the DMP property (7) is achieved. To this end, a prescribed maximum degree $d_{\max}$ is introduced and used to perform an initial polynomial reconstruction on each cell. Through an iterative decremental procedure, we determine the *optimal degree* $d_i \leqslant d_{\max}$ on each cell K_i such that each updated mean value $u_i^\star$ fulfills the DMP property (see Fig. 2 bottom).

In the following we focus on the quadratic polynomial case $d_{\max} = 2$ and first present the local quadratic reconstruction of [19]. Then the MOOD method is detailed and we prove that the numerical approximations satisfy the DMP property.

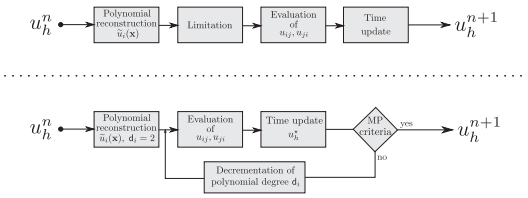


Fig. 2. Classical high-order methods idea (top) and MOOD idea (bottom).

4.1. Quadratic reconstruction

Using the same framework as in Section 3.1, the quadratic polynomial reconstruction is written

$$\tilde{u}(\mathbf{x}) = \bar{u} + G\cdot(\mathbf{x}-\mathbf{c}) + \frac{1}{2}\left((\mathbf{x}-\mathbf{c})^t H(\mathbf{x}-\mathbf{c}) - \overline{H}\right) \tag{11}$$

with

$$\overline{H} = \langle (\mathbf{x}-\mathbf{c})^t H(\mathbf{x}-\mathbf{c})\rangle_K, \quad H = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22}\end{bmatrix},$$

where matrix H is an approximation of the Hessian matrix $\nabla^2 u$ on K. Note that by construction, the mean value of $\tilde{u}$ on K is still equal to $\bar{u}$. A minimization technique is used to compute G and H. To this end, for a cell K_j, let us define the integrals

$$\mathbf{x}_{K_j}^{[\alpha,\beta]} = \left\langle (x-c_1)^\alpha (y-c_2)^\beta \right\rangle_{K_j} - \left\langle (x-c_1)^\alpha (y-c_2)^\beta \right\rangle_{K}.$$

Algebraic manipulations yield the following expression for $\langle\tilde{u}(\mathbf{x})\rangle_{K_j}$

$$\langle\tilde{u}(\mathbf{x})\rangle_{K_j} = \bar{u} + \left(G_1\mathbf{x}_{K_j}^{(1,0)} + G_2\mathbf{x}_{K_j}^{(0,1)}\right) + \frac{1}{2}\left(H_{11}\mathbf{x}_{K_j}^{(2,0)} + 2H_{12}\mathbf{x}_{K_j}^{(1,1)} + H_{22}\mathbf{x}_{K_j}^{(0,2)}\right). \tag{12}$$

This expression is further derived for any cell K_j with $j\in v$ to form an over-determined linear system of the form $A\Lambda = B$ with

$$A = \begin{pmatrix} \mathbf{x}_{K_1}^{(1,0)} & \mathbf{x}_{K_1}^{(0,1)} & \mathbf{x}_{K_1}^{(2,0)} & \mathbf{x}_{K_1}^{(1,1)} & \mathbf{x}_{K_1}^{(0,2)} \\ \mathbf{x}_{K_2}^{(1,0)} & \mathbf{x}_{K_2}^{(0,1)} & \mathbf{x}_{K_2}^{(2,0)} & \mathbf{x}_{K_2}^{(1,1)} & \mathbf{x}_{K_2}^{(0,2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}_{K_N}^{(1,0)} & \mathbf{x}_{K_N}^{(0,1)} & \mathbf{x}_{K_N}^{(2,0)} & \mathbf{x}_{K_N}^{(1,1)} & \mathbf{x}_{K_N}^{(0,2)} \end{pmatrix}, \quad \Lambda = \begin{pmatrix} G_1 \\ G_2 \\ \frac{1}{2}H_{11} \\ H_{12} \\ \frac{1}{2}H_{22}\end{pmatrix}, \quad B = \begin{pmatrix} u_1-\bar{u} \\ u_2-\bar{u} \\ \vdots \\ u_N-\bar{u}\end{pmatrix} \tag{13}$$

with $N = \#v$. This system is solved with a QR decomposition of A using Householder transformations, such that $Q\in\mathcal{M}_{N\times N}(\mathbb{R})$ is an orthogonal matrix and $R\in\mathcal{M}_{N\times 5}(\mathbb{R})$ an upper-triangular one. Finally back-substitution of $R\Lambda = Q^tB$ defines $\tilde{u}$ (see [19]).

Remark 5. A left preconditioner matrix can be applied to reduce the system sensitivity and improve the reconstruction quality. For example, in [19], the authors use a diagonal matrix whose coefficients $\omega_{jj} = \|\mathbf{c}_j - \mathbf{c}\|^{-2}$ $(j = 1,\dots,N)$ correspond to geometrical weights in order to promote closest informations.

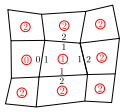
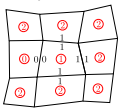
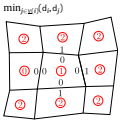
4.2. Description of the MOOD method

We now detail the MOOD technique considering the simple case where an explicit time discretization is employed. Moreover, without loss of generality, we present the method using only one quadrature point ($R = 1$) and skip the subscript r denoting u_{ij} in place of $u_{ij,r}$. Extension to several quadrature points ($R > 1$) is straightforward.

Assume that we have $u_h^n = \sum_{i\in\mathcal{E}_{el}} u_i^n \mathbb{1}_{K_i}$ an approximation of u at time t^n, the goal is to build a relevant $u_h^{n+1} = \sum_{i\in\mathcal{E}_{el}} u_i^{n+1}\mathbb{1}_{K_i}$ at time $t^{n+1} = t^n + \Delta t$. To this end, we define the following fundamental notions:

- d_i is the Cell Polynomial Degree (CellPD) which represents the degree of the polynomial reconstruction on cell K_i.
- d_{ij} and d_{ji} are the Edge Polynomial Degrees (EdgePD) which correspond to the effective degrees used to respectively build u_{ij} and u_{ji} on both sides of edge e_{ij}.

Table 1
Evaluation of the EdgePD d_{ij} using the CellPD of the two neighbor elements. Analytic formula on first line. Examples on the second line where CellPD are surrounded in red and EdgePD for internal edges are in black. Missing cells are assumed to have CellPD equal to 2.

	EPD_0 strategy	EPD_1 strategy	EPD_2 strategy
EdgePD d_{ij}	d_i	$\min(\mathrm{d}_i,\mathrm{d}_j)$	$\min_{j\in\overline{v}(i)}(\mathrm{d}_i,\mathrm{d}_j)$
Example			

The MOOD method consists of the following iterative procedure:

1. **CellPD** initialization. Each CellPD is initialized to d_{max}.
2. **EdgePD** evaluation. Each EdgePD is set up as a function of the neighboring CellPD (see Table 1).
3. Quadrature points evaluation. Each u_{ij} is evaluated with the polynomial reconstruction of degree d_{ij}.
4. Mean values update. The updated values $u_h^\star$ are computed using the finite volume scheme (3).
5. DMP test. The DMP criterion is checked on each cell K_i

$$\min_{j\in\overline{v}(i)}(u_j^n, u_i^n) \leqslant u_i^\star \leqslant \max_{j\in\overline{v}(i)}(u_j^n, u_i^n). \tag{14}$$

   If $u_i^\star$ does not satisfy (14) the CellPD is decremented, $\mathrm{d}_i := \max(0, \mathrm{d}_i - 1)$.
6. Stopping criterion. If all cells satisfy the DMP property, the iterative procedure stops with $(\rho, u_1, u_2, E)_h^{n+1} = (\rho, \rho u_1, \rho u_2, E)_h^\star$ else go to step 2.

We give in Table 1 three possible strategies of EdgePD calculation. The simplest one named EPD_0 consists of setting $\mathrm{d}_{ij} = \mathrm{d}_i$ and $\mathrm{d}_{ji} = \mathrm{d}_j$ whereas EPD_1 chooses the minimal value between d_i and d_j for both d_{ij} and d_{ji}. At last, the smallest CellPD of all the direct neighbor cells is taken in the EPD_2 strategy.

To conclude the section, there are two important remarks which dramatically reduce the computational cost.

Remark 6. If $\mathrm{d}_{ij} < d_{\max}$, there is no need to recompute a polynomial of degree d_{ij}, a simple truncation of the initial polynomial of degree $d_{\max}$ should be performed.

Remark 7. Only cells K_i where CellPD has been decremented and their neighbors in a compact stencil have to be updated. Consequently only these cells have to be checked during next iterations of the MOOD procedure in the current time step. For instance the compact stencil for EPD_0 and EPD_1 is $\underline{v}(i)$ while for EPD_2 it is $\{\underline{v}(i)\cup\{\underline{v}(j), j\in\underline{v}(i)\}\}$.

4.3. Convergence of the MOOD method

We first recall the classical stability result (see [6] and references herein).

Proposition 8. *Let us consider the generic first-order finite volume scheme (2) with reflective boundary conditions. If the numerical flux is consistent and monotone, then the DMP property given by Definition 2 is satisfied.*

It implies that if $u_{ij} = u_i$ and $u_{ji} = u_j$ for all $j\in\underline{v}(i)$ then relation (7) holds. To prove that the iterative MOOD method provides a solution which satisfies the DMP, we introduce the following definition.

Definition 9. An EPD strategy is said upper-limiting (with respect to the CellPD) if for any K_i

$$\mathrm{d}_i = \bar{\mathrm{d}} \;\Rightarrow\; \mathrm{d}_{ij} \leqslant \bar{\mathrm{d}} \quad \text{and} \quad \mathrm{d}_{ji} \leqslant \bar{\mathrm{d}}, \quad \forall j\in\underline{v}(i). \tag{15}$$

We then have the following theorem.

Theorem 10. *Let us consider the generic high-order finite volume scheme with reflective boundary conditions and assume that the numerical flux is consistent and monotone. If the* EPD *strategy is upper-limiting then the MOOD method provides an updated solution u_h^{n+1} which satisfies the DMP property after a finite number of iterations.*

Proof. Let d_i be the CellPD of cell K_i. If $\mathrm{d}_i = 0$, then Eq. (15) implies that $\mathrm{d}_{ij} = \mathrm{d}_{ji} = 0$, hence $u_{ij}^n = u_i^n$ and $u_{ji}^n = u_j^n$, for all $j\in\underline{v}(i)$. We recover the first-order scheme (2) and Proposition 8 yields that u_i^{n+1} satisfies the DMP property (7). Otherwise, if $\mathrm{d}_i > 0$ then two situations arise. Either the Maximum Principle is satisfied and we do not modify d_i or we decrement d_i. Consequently if the maximum principle is not satisfied for all cells, then there is at least one cell having its CellPD positive which has to be decremented. Since we cannot decrement more than $d_{\max}\times\#(\mathcal{E}_{el})$ times, the iterative procedure stops after a finite number of iterations and the solution satisfies the DMP property. □

Remark 11. Note that EPD_1 and EPD_2 are upper-limiting strategies whereas EPD_0 strategy does not satisfy condition (15). Thus EPD_0 cannot be used since MOOD iterative procedure may loop endlessly.

Remark 12. To carry out a third-order Runge–Kutta time discretization (6) which provides a solution satisfying the DMP property, one has to perform the MOOD technique for each explicit sub-step since (6d) can be written as a convex combination.

5. Extension to the Euler equations

In this section, we propose an extension of the MOOD method to the Euler equations

$$\partial_t\begin{pmatrix}\rho\\ \rho u_1\\ \rho u_2\\ E\end{pmatrix} + \partial_{x_1}\begin{pmatrix}\rho u_1\\ \rho u_1^2 + p\\ \rho u_1 u_2\\ u_1(E+p)\end{pmatrix} + \partial_{x_2}\begin{pmatrix}\rho u_2\\ \rho u_1 u_2\\ \rho u_2^2 + p\\ u_2(E+p)\end{pmatrix} = 0, \tag{16}$$

where ρ, $\mathbf{V} = (u_1, u_2)$ and p are the density, velocity and pressure, respectively, while the total energy per unit volume E is given by

$$E = \rho\left(\frac{1}{2}\mathbf{V}^2 + e\right), \quad \mathbf{V}^2 = u_1^2 + u_2^2,$$

where e is the specific internal energy. For an ideal gas, this system is closed by the equation of state

$$e = \frac{p}{\rho(\gamma-1)}$$

with γ the ratio of specific heats.

Despite that the physical variables do not have to respect the maximum principle, classical methods such as the MUSCL technique use a limiting procedure derived from the scalar case to keep the numerical solution from producing spurious oscillations. A popular choice consists of reconstructing and limiting the density, the velocity components and the pressure variables but other limitations can be carried out: the internal energy, the specific volume or the characteristic variables for instance.

Although applying the MOOD technique to each variable independently gives physically admissible solutions, an excessive diffusion is noticed. We thus propose a strategy to both have an accurate approximation where the solution is smooth and prevent the oscillations from appearing close to the discontinuities. In the following we consider ρ, u_1, u_2 and p as the variables to be reconstructed.

First we have to provide physically relevant reconstructed values at quadrature points, and since no limitation is used in the MOOD method, negative reconstructed values for pressure or density must be avoided (it would be the same for energy or specific volume). In that case, first-order values are substituted to the unphysical reconstructed values, for instance if the reconstructed value ρ_{ij} is negative on cell K_i, we replace it with the mean value ρ_i.

We now describe how we choose to use the two fundamental notions of the MOOD method (CellPD and EdgePD) in the Euler equations framework. Instead of using one CellPD per cell and per variable, we choose to define only one CellPD per cell and to use it for all variables. Consequently only one EdgePD is defined per side of an edge and used for all variables.

As in the scalar case, we first build the local polynomial reconstruction of maximal degree $d_{\max}$ for each variable. Then we apply the MOOD algorithm of Section 4.2 where we substitute steps 5 and 6 with the following stages:

5. Density DMP test. The DMP criterion is checked on the density

$$\min_{j\in\overline{v}(i)}(\rho_j^n, \rho_i^n) \leqslant \rho_i^\star \leqslant \max_{j\in\overline{v}(i)}(\rho_j^n, \rho_i^n). \tag{17}$$

   If $\rho_i^\star$ does not satisfy (17) the CellPD is decremented, $\mathrm{d}_i := \max(0, \mathrm{d}_i - 1)$.
6. Pressure positivity test. The pressure positivity is checked and if $p_i^\star \leqslant 0$ and d_i has not been altered by step 5 then the CellPD is decremented, $\mathrm{d}_i := \max(0, \mathrm{d}_i - 1)$.
7. Stopping criterion. If, for all $i\in\mathcal{E}_{el}$, d_i has not been altered by steps 5 and 6 then the iterative procedure stops and returns $u_h^{n+1} = u_h^\star$ else go to step 2.

Next section is dedicated to numerical experiments to assess the computational efficiency of the MOOD method.

6. Numerical results — the scalar case

Let Ω be the unit square $[0,1]\times[0,1]$. We first consider the linear advection problem of a scalar quantity u with velocity $V(\mathbf{x})$:

$$\partial_t u + \nabla.(Vu) = 0, \tag{18a}$$
$$u(.,0) = u^0, \tag{18b}$$

where $V(\mathbf{x})$ is a given continuous function on Ω and u^0 is the initial function we shall characterize in the following. In this section periodic boundary conditions are prescribed on $\partial\Omega$.

Comparisons are drawn between the simple first-order finite volume method (denoted FV with an abuse of terminology), the MUSCL method proposed in [21] (MLP) and the MOOD method with $d_{\max} = 1$ (MOOD-P1) and $d_{\max} = 2$ (MOOD-P2).

We use the following monotone upwind numerical flux (see Eq. (2))

$$\mathbb{F}(u_i^n, u_j^n, \mathbf{n}_{ij}) = [V(\mathbf{x})\cdot\mathbf{n}_{ij}]^+ u_i^n + [V(\mathbf{x})\cdot\mathbf{n}_{ij}]^- u_j^n,$$

where the velocity is evaluated at the quadrature point $\mathbf{x}$ and the positive and negative parts are respectively defined by

$$[\alpha]^+ = \max(0,\alpha) \quad \text{and} \quad [\alpha]^- = \min(0,\alpha).$$

Notice that we use $\overline{v}(i)$ as the reconstruction stencil. Lastly two Gauss points are used on each edge to provide a third-order accurate spatial integration while time integration is performed with a forward Euler scheme for the FV method and with the RK3-TVD method given by system (6) for the MLP and MOOD methods.

Following Remark 12, we simply apply the MOOD procedure detailed in Section 4.2 to each sub-step of the RK3-TVD. The CellPD are thus reinitialized to $d_{\max}$ at the beginning of each time sub-step.

6.1. Test descriptions

The method accuracy is measured using L^1 and L^∞ errors which are computed with

$$err_1 = \sum_{i\in\mathcal{E}_{el}} |u_i^N - u_i^0||K_i| \quad \text{and} \quad err_\infty = \max_{i\in\mathcal{E}_{el}} |u_i^N - u_i^0|,$$

where $(u_i^0)_i$ and $(u_i^N)_i$ are respectively the cell mean values at initial time $t = 0$ and final time $t = t_f = N\Delta t$.

Two classical numerical experiments are carried out to demonstrate the ability of the method to provide effective third-order accuracy and to handle discontinuities with a very low numerical diffusion.

Double Sine Translation (DST): We consider a constant velocity $V = (2,1)$ and the initial condition is the C^∞ function

$$u^0(x_1,x_2) = \sin(2\pi x_1)\sin(2\pi x_2).$$

The final time is $t_f = 2.0$. Since we use periodic boundary conditions, the final time corresponds to a full revolution such that the exact solution coincides with the initial one.

Solid Body Rotation (SBR): First introduced by Leveque [17], this solid body rotation test uses three shapes which are a hump, a cone and a slotted cylinder. Each shape is located within a circle of radius $r^0 = 0.15$ and centered at (x_1^0, y_2^0).

Hump centered at $(x_1^0, x_2^0) = (0.25, 0.5)$

$$u^0(x_1,x_2) = \frac{1}{4}(1+\cos(\pi\min(r(x_1,x_2),1))).$$

Cone centered at $(x_1^0, x_2^0) = (0.5, 0.25)$

$$u^0(x_1,x_2) = 1 - r(x_1,x_2).$$

Slotted cylinder centered at $(x_1^0, x_2^0) = (0.5, 0.75)$

$$u^0(x_1,x_2) = \begin{cases} 1 & \text{if } |x_1 - 0.5| < 0.25, \text{ or } x_2 > 0.85, \\ 0 & \text{elsewhere,} \end{cases}$$

where $r(x_1,x_2) = \frac{1}{\pi}\sqrt{(x_1-x_1^0)^2 + (x_2-x_2^0)^2}$. To perform the rotation, we use the velocity $V(\mathbf{x}) = (-x_2 + 0.5, x_1 - 0.5)$ and the final time $t_f = 2\pi$ corresponds to one full rotation.

6.2. Numerical results

6.2.1. Comparison between EPD_1 and EPD_2 strategies

We consider the DST test case on uniform meshes from 20×20 to 160×160 cells and compare the L^1 and L^∞ errors and convergence rates displayed in Table 2 using EPD_1 and EPD_2 strategies with the MOOD-P2 method. We obtain an almost effective third-order convergence in L^1 norm and a 1.6 convergence rate in L^∞ norm for the two strategies. We observe in this case that the L^1 and L^∞ errors for EPD_1 are slightly less important than for EPD_2 and the convergence orders seem to indicate that the EPD_1 strategy should be privileged. Moreover, from a practical point of view, the EPD_1 implementation is performed with a more compact stencil than the EPD_2 (see Remark 7). In the sequel, only EPD_1 strategy is used.

6.2.2. Comparison between FV, MLP, MOOD-P1 and MOOD-P2 with EPD_1 strategy on uniform meshes

Double Sine Translation. We report in Tables 3–5 the L^1 and L^∞ errors and convergence rates for FV, MLP, MOOD-P1, MOOD-P2, unlimited P1 and P2 reconstruction methods, respectively. At last, we plot in Fig. 3 the convergence curves for the four methods as well as the convergence curves for the unlimited versions.

The high-order finite volume methods with the two Gauss points and the RK3 time scheme reach the optimal convergence rate for the unlimited P1 and P2 reconstructions hence the limiting procedure has to be blamed for the accuracy discrepancy.

Fig. 3 shows that the optimal convergence rate in L^1 error for P1, MOOD-P1 and MLP methods is achieved since the curves fit very well. On the other hand, the P2 and MOOD-P2 curves are very close and parallel which confirms that MOOD-P2 is an effective third-order method for the L^1 norm. For the L^∞ norm, none of the limited methods is over the effective second-order while the unlimited P1 and P2 provide an effective second- and third-order, respectively. Indeed the strict maximum

Table 2
L^1 and L^∞ errors and convergence rates for DST problem with the MOOD-P2 method: EPD_1 strategy (left) and EPD_2 strategy (right).

Nb of cells	EPD_1				EPD_2			
	err_1		err_∞		err_1		err_∞	
20×20	9.469E−02	–	3.960E−01	–	1.104E−01	–	4.506E−01	–
40×40	1.113E−02	3.09	1.333E−01	1.57	1.382E−02	3.00	1.566E−01	1.52
80×80	1.768E−03	2.65	4.164E−02	1.68	2.309E−03	2.58	5.196E−02	1.59
160×160	2.481E−04	2.83	1.304E−02	1.68	3.262E−04	2.82	1.698E−02	1.61

Table 3
L^1 and L^∞ errors and convergence rates for the DST on uniform meshes with FV and MLP methods.

Nb of cells	FV				MLP			
	err_1		err_∞		err_1		err_∞	
20×20	3.924E−01	–	9.371E−01	–	1.417E−01	–	3.765E−01	–
40×40	3.480E−01	0.17	8.375E−01	0.16	3.038E−02	2.22	1.121E−01	1.75
80×80	2.663E−01	0.39	6.241E−01	0.42	6.904E−03	2.14	3.534E−02	1.67
160×160	1.734E−01	0.62	3.964E−01	0.65	1.693E−03	2.03	1.167E−02	1.60

Table 4
L^1 and L^∞ errors and convergence rates for the DST on uniform meshes with MOOD-P1 and MOOD-P2 methods.

Nb of cells	MOOD-P1				MOOD-P2			
	err_1		err_∞		err_1		err_∞	
20×20	1.502E−01	–	4.876E−01	–	9.469E−02	–	3.960E−01	–
40×40	3.141E−02	2.26	1.629E−01	1.58	1.113E−02	3.09	1.333E−01	1.57
80×80	7.438E−03	2.08	5.188E−02	1.65	1.768E−03	2.65	4.164E−02	1.68
160×160	1.787E−03	2.06	1.675E−02	1.63	2.481E−04	2.83	1.304E−02	1.68

Table 5
L^1 and L^∞ errors and convergence orders for the DST on uniform meshes with P1 and P2 methods.

Nb of cells	P1				P2			
	err_1		err_∞		err_1		err_∞	
20×20	1.334E−01	–	3.227E−01	–	7.130E−02	–	1.729E−01	–
40×40	2.896E−02	2.20	6.593E−02	2.29	9.877E−03	2.85	2.427E−02	2.83
80×80	6.604E−03	2.13	1.408E−02	2.23	1.255E−03	2.98	3.091E−03	2.97
160×160	1.603E−03	2.04	3.310E−03	2.09	1.573E−04	3.00	3.876E−04	3.00

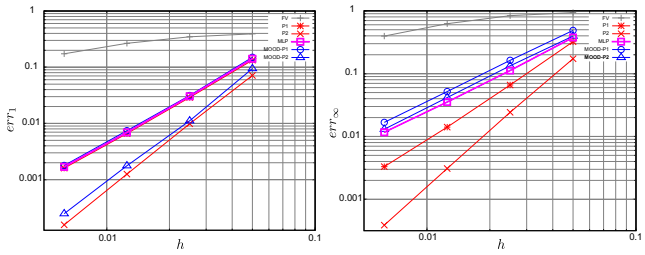


Fig. 3. Convergence curves of err_1(left) and err_∞(right) for the DST on uniform meshes.

principle application at extrema is responsible for the L^∞ error discrepancy and we can expect nothing more than a second-order scheme in L^∞ norm, whatever the polynomial degree is when the DMP condition is enforced.

Solid Body Rotation. We employ a 140×140 uniform mesh of square elements in order to compare our results with $100 \times 100 \times 2$ triangular mesh in Ref. [21]. We display in the left panels of Fig. 4 three-dimensional elevations while top views of ten uniformly distributed isolines from 0 to 1 are printed in the right panels. We can measure the scheme accuracy by counting the number of isolines outside of the slot since the exact solution isolines would fit the slot shape. The smaller number of isolines outside of the slot is, the more accurate the scheme is. With the MLP reconstruction, we observe three isolines outside while we have only two with the MOOD-P1. At last, the outstanding result is that we have just one isoline outside of the slot with the MOOD-P2 method which proves the great ability of the technique to handle and preserve discontinuities.

6.2.3. Comparison between FV, MLP, MOOD-P1 and MOOD-P2 with EPD_1 strategy on non-uniform meshes

Approximation accuracy is reduced when one employs meshes with large deformations, *i.e.* the elements are no longer rectangular but quadrilateral with large aspect ratios. The present subsection investigates the MOOD method sensitivity to mesh distortion.

To obtain the distorted mesh for the DST, we proceed in two stages. First the following transformation is applied to an uniform mesh

$$x_1 \to \begin{cases} x_1(10x_1^2 + 5x_1 + 1), & \text{if } x_1 \leqslant 0.5, \\ (x_1 - 1)(10(x_1 - 1)^2 + 5(x_1 - 1)) + 1, & \text{elsewhere} \end{cases}$$

and we operate in the same way with variable x_2.

Then we apply a second transformation

$$x_1 \to x_1 + 0.1|x_1 - 0.5|\cos(6\pi(x_2 - 0.5))\sin(4\pi(x_1 - 0.5)),$$
$$x_2 \to x_2 + 0.1|x_2 - 0.5|\cos(4\pi(x_1 - 0.5))\sin(6\pi(x_2 - 0.5)).$$

As an example two non-uniform meshes are given in Fig. 5. Notice that the shape of domain Ω is preserved by the transformation.

Double Sine Translation. We report in Tables 6–8 the L^1 and L^∞ errors and convergence rates for FV, MLP, MOOD-P1, MOOD-P2, unlimited P1 and P2 reconstruction methods, respectively. At last, we plot in Fig. 6 the convergence curves for the four methods as well as the convergence curves for the unlimited versions.

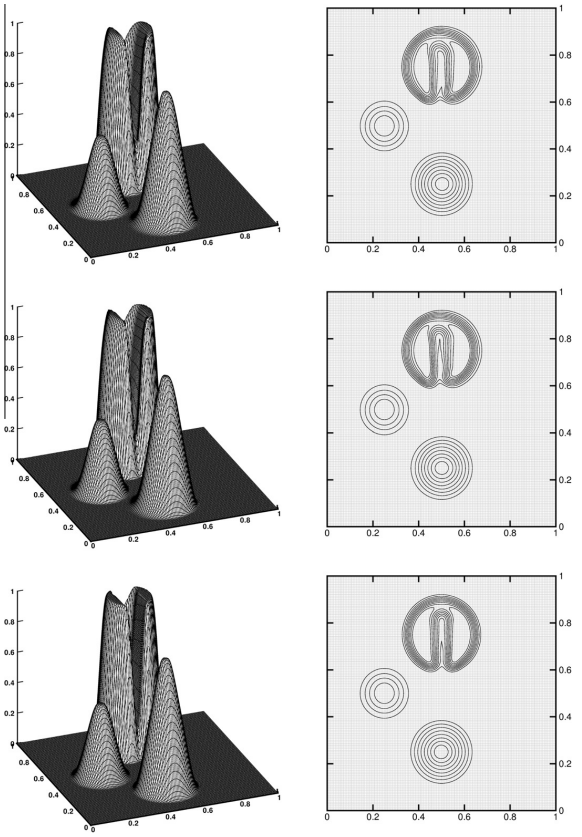


Fig. 4. Results of SBR on a 140×140 uniform mesh. Isolines are from 0 to 1 by 0.1. Top: MLP method — Middle: MOOD-P1 method — Bottom: MOOD-P2 method.

We first observe in Table 8 an accuracy discrepancy with the unlimited reconstructions since the L^∞ errors are roughly 10 times larger for the distorted mesh than for the uniform one given in Table 5. Nevertheless, we obtain good effective rates of

convergence both in L^1 and L^∞ norm for the P1 and P2 reconstructions. Optimal second-order scheme is achieved for the P1 method and convergence rate is around 2.9 for the P2 reconstruction.

For the L^1 norm, P1, MOOD-P1 and MLP convergence curves fit well hence we get the optimal accuracy with the three methods. In the same way, the P2 and MOOD-P2 are also superimposed which means that MOOD-P2 is optimal with respect to the unlimited case. For the L^∞ norm, MLP method convergence rate is around 1.6 whereas the MOOD-P1, MOOD-P2 and P1 provide a 1.9 convergence rate. Notice that the MOOD-P2 produces more accurate results but does not reach the third-order convergence since it has to respect a strict DMP property. Finally, Table 9 shows that the extrema are better approximated with respect to the exact solution with the MOOD methods than the MLP method, in particular when coarse meshes are employed.

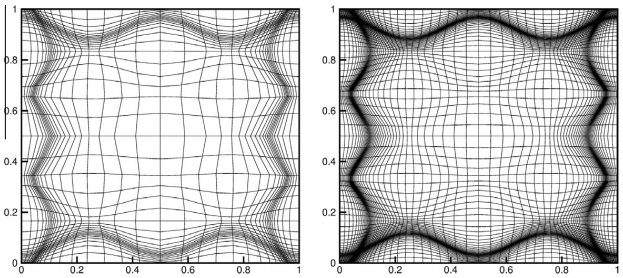


Fig. 5. The 40×40 and 80×80 non-uniform meshes for the DST.

Table 6
L^1 and L^∞ errors and convergence rates for the DST on non-uniform meshes with FV and MLP methods.

Nb of cells	FV				MLP			
	err_1		err_∞		err_1		err_∞	
20×20	4.053E−01	–	9.032E−01	–	3.907E−01	–	8.752E−01	–
40×40	4.038E−01	0.01	9.822E−01	−0.12	1.893E−01	1.05	5.306E−01	0.72
80×80	3.834E−01	0.07	9.486E−01	0.05	4.370E−02	2.11	1.806E−01	1.55
160×160	3.144E−01	0.29	7.825E−01	0.28	9.846E−03	2.15	5.889E−02	1.62

Table 7
L^1 and L^∞ errors and convergence rates for the DST on non-uniform meshes with MOOD-P1 and MOOD-P2 methods.

Nb of cells	MOOD-P1				MOOD-P2			
	err_1		err_∞		err_1		err_∞	
20×20	3.770E−01	–	8.557E−01	–	3.408E−01	–	7.897E−01	–
40×40	1.599E−01	1.24	4.541E−01	0.91	8.992E−02	1.92	3.222E−01	1.29
80×80	3.892E−02	2.04	1.314E−01	1.79	1.375E−02	2.71	9.199E−02	1.81
160×160	9.170E−03	2.09	3.374E−02	1.96	1.922E−03	2.84	2.483E−02	1.89

Table 8
L^1 and L^∞ errors and convergence rates for the DST on non-uniform meshes with P1 and P2 methods.

Nb of cells	P1				P2			
	err_1		err_∞		err_1		err_∞	
20×20	3.658E−01	–	8.312E−01	–	FAIL	–	FAIL	–
40×40	1.534E−01	1.25	3.793E−01	1.13	8.328E−02	–	2.135E−01	–
80×80	3.856E−02	1.99	9.760E−02	1.96	1.403E−02	2.57	3.582E−02	2.58
160×160	9.052E−03	2.09	2.643E−02	1.88	1.920E−03	2.87	4.917E−03	2.86

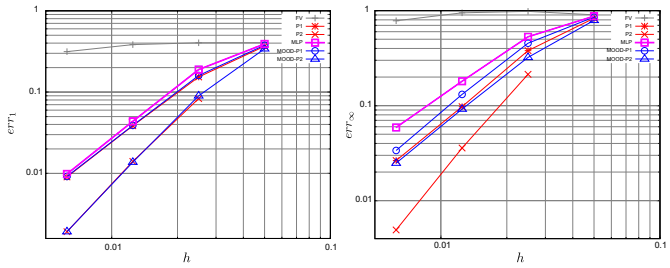


Fig. 6. Convergence curves of err_1(left) and err_∞(right) for the DST on non-uniform meshes.

Table 9
Min and Max for DST on non-uniform meshes with MLP, MOOD-P1 and MOOD-P2.

Nb of cells	MLP		MOOD-P1		MOOD-P2	
	Min	Max	Min	Max	Min	Max
20×20	−3.740E−02	3.479E−02	−7.168E−02	7.566E−02	−1.376E−01	1.516E−01
40×40	−4.634E−01	4.645E−01	−5.445E−01	5.458E−01	−6.738E−01	6.792E−01
80×80	−8.179E−01	8.204E−01	−8.747E−01	8.743E−01	−9.098E−01	9.079E−01
160×160	−9.433E−01	9.431E−01	−9.655E−01	9.668E−01	−9.752E−01	9.748E−01

Solid Body Rotation. The mesh deformation presented above is not as relevant for the SBR as for the DST since the solid bodies rotate and do not go through the boundaries. A slight modification of the first step has been done

$$x_1 \to \begin{cases} x_1(5x_1^2 + 2.5x_1 + 1), & \text{if } x_1 \leqslant 0.5, \\ (x_1 - 1)(5(x_1 - 1)^2 + 2.5(x_1 - 1)) + 1, & \text{elsewhere} \end{cases}$$

and we operate in the same way with variable x_2.

The 140×140 non-uniform mesh is visible on the isolines top views. We display in the left panels of Fig. 7 three-dimensional elevations while top views of 10 uniformly distributed isolines from 0 to 1 are in the right panels.

As in the smooth case, MOOD methods perform better than MLP on the distorted mesh. Although they are both second-order methods, we notice that MOOD-P1 gives a clearly better solution than the one computed with MLP, even on the smooth profiles. Moreover the MOOD-P2 result supports the usefulness of using a third-order method since an important gain in symmetry of the solution is obtained.

7. Numerical results — the Euler case

We now turn to the Euler Eq. (16) to test the MOOD method. Efficiency, accuracy and stability of the method are investigated on classical tests. In the present article, we use the HLL numerical flux detailed in [26]. Once again comparisons are drawn with the MLP technique proposed in [21]. We apply the MOOD method using the detection strategy presented in Section 5 to each sub-step of the RK3-TVD time discretization.

First the classical 1D Sod shock tube is used to test the ability of MOOD in reproducing simple waves. This test is first run on an uniform mesh and then on a non-uniform one to estimate the gain obtained when using MOOD method. Then we proceed with a 2D Riemann problem proposed by [25] (see also [18]). We conclude the series of tests with two classical references, the Mach 3 wind tunnel with a step problem [21,29] and the double Mach problem [21,29]. These two tests are run with MLP, MOOD-P1 and MOOD-P2 on uniform meshes for comparison purposes with classical results from literature.

7.1. Sod shock tube

The one dimensional Sod problem is used as a sanity check for the MOOD method. The computational domain is the rectangular domain $\Omega = [0,1] \times [0,0.2]$. The exact solution is invariant in x_2-direction. The interface between the left state

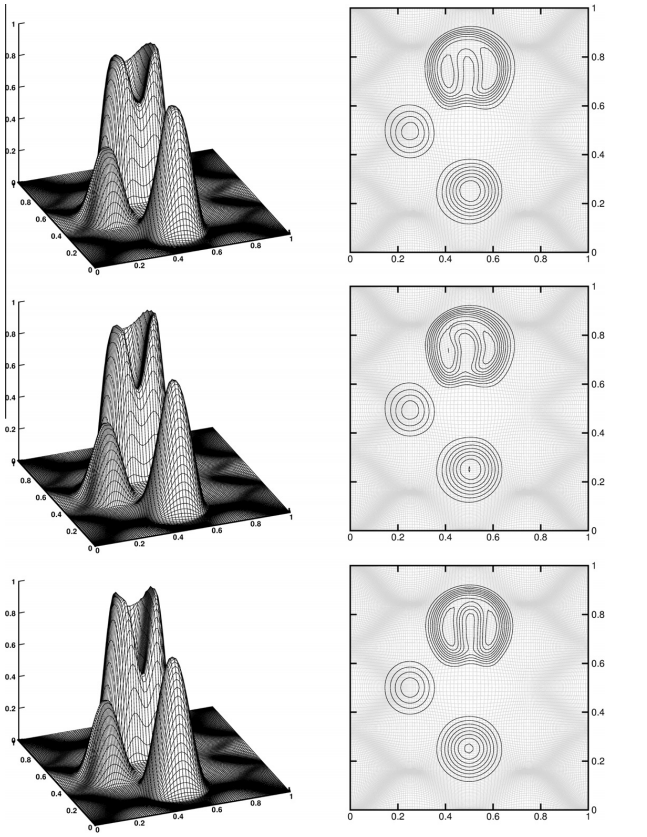


Fig. 7. Results of SBR on a 140×140 non-uniform mesh. Isolines are from 0 to 1 by 0.1. Top: MLP method — Middle: MOOD-P1 method — Bottom: MOOD-P2 method.

$(\rho, u_1, u_2, p) = (1,0,0,1)$ and the right one $(0.125,0,0,0.1)$ is located at $x_1 = 0.5$. Reflective boundary conditions are prescribed. The final time is $t_f = 0.2$.

Uniform mesh. The computational domain is uniformly meshed by 100 cells in the x_1 direction and 10 cells in the x_2 direction. We plot the density and the x_1-velocity at the final time with the exact solution using the MLP, MOOD-P1 and MOOD-P2 methods in Fig. 8. The curves show a very good agreement between the three methods. The plateau between the contact and the shock is wavy with the MLP method while MOOD produces better constant states. However we observe an undershoot (resp. overshoot) at the tail of the rarefaction with MOOD-P2 for the density (resp. velocity).

Non-uniform mesh. The same simulation is performed on the non-uniform mesh plotted in Fig. 9. The density and the x_1-velocity solutions at the final time using the MLP, MOOD-P1 and MOOD-P2 methods are also printed in Fig. 9. All cell values are represented so that the preservation of the 1D symmetry in the x_2 direction can be evaluated by the thickness of the points cloud. Clearly the MLP method provides the largest dispersion whereas the MOOD-P2 method manages to better preserve the x_2 invariance. Such a test case suggests that the MOOD method is less sensitive to mesh deformation. As in the uniform case an undershoot at the tail of the rarefaction wave appears for MOOD-P2 method but the solution is genuinely

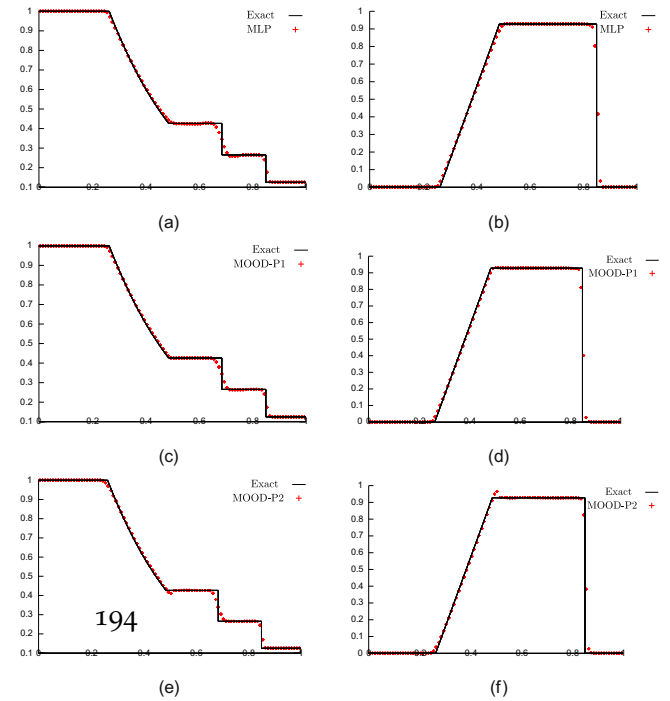


Fig. 8. Sod shock tube problem: density and x_1-velocity solutions on 100×10 uniform mesh for (a and b): MLP — (c and d): MOOD-P1 — (e and f): MOOD-P2.

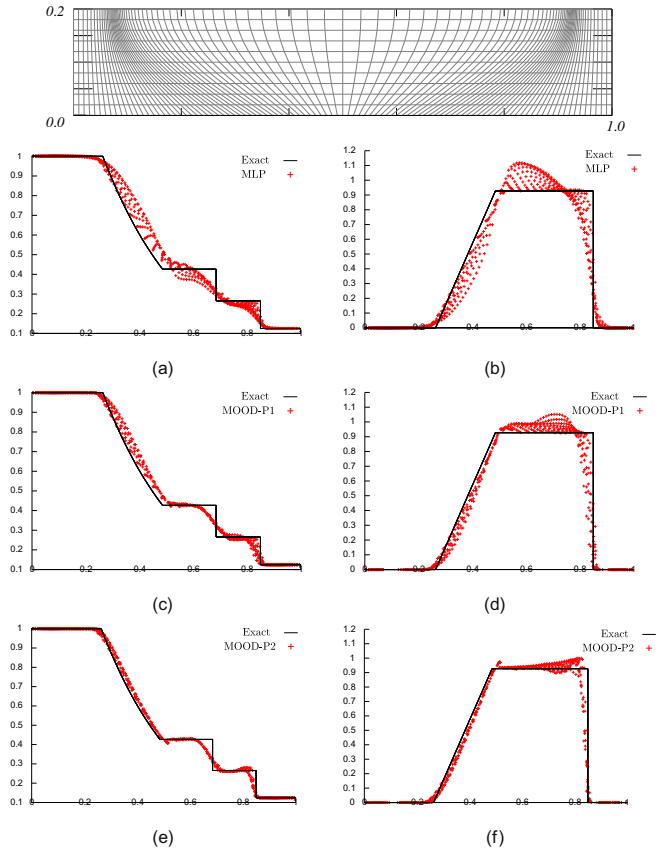


Fig. 9. Sod shock tube problem: non-uniform 100×10 mesh (Top) — density and x_1-velocity solutions on the above mesh for (a and b): MLP — (c and d): MOOD-P1 — (e and f): MOOD-P2.

improved by comparison with MLP. The MOOD-P1 is an intermediate case where the dispersion is reduced in comparison with the MLP method but where the MOOD-P2 accuracy is not reached.

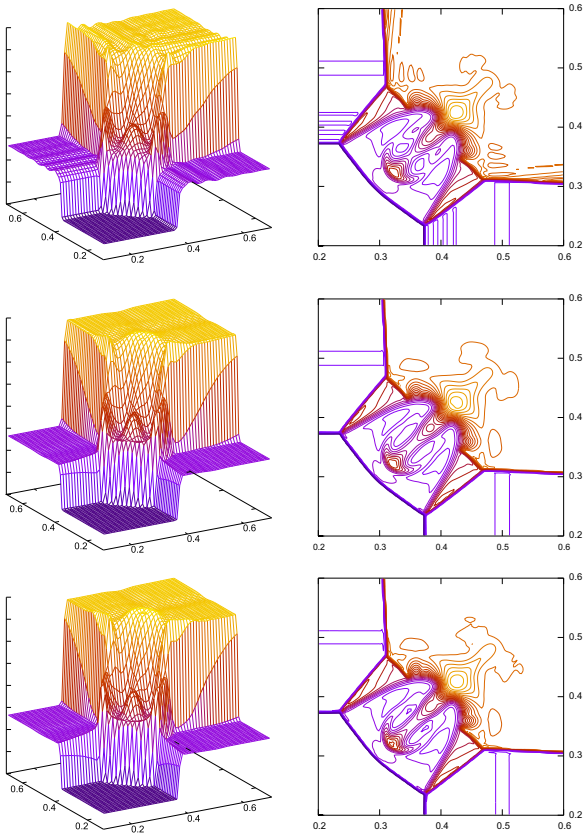


Fig. 10. Density solution to the four states Riemann problem. On the left 3D views on the 100×100 mesh. On the right top views with 30 isolines between ρ_m and ρ_M on the 400×400 mesh. Top: MLP method $\rho_m = 0.138$, $\rho_M = 1.821$ — Middle: MOOD-P1 method $\rho_m = 0.1377$, $\rho_M = 1.805$ — Bottom: MOOD-P2 method $\rho_m = 0.1379$, $\rho_M = 1.805$.

7.2. Four states Riemann problem

We now deal with one of the four states Riemann problem which corresponds to a truly 2D Riemann problem. The computational domain $\Omega = [0,1] \times [0,1]$ is first uniformly meshed by a 100×100 and then by a 400×400 quadrangles grid. The four sub-domains correspond to four identical squares separated by the lines $x_1 = 0.5$ and $x_2 = 0.5$. Initial conditions on each sub-domains are

- for the lower-left domain Ω_{ll}, $(\rho, u_1, u_2, p) = (0.029, 0.138, 1.206, 1.206)$,
- for the lower-right domain Ω_{lr}, $(0.3, 0.5323, 0, 1.206)$,
- for the upper-right domain Ω_{ur}, $(1.5, 1.5, 0, 0)$,
- for the upper-left domain Ω_{ul}, $(0.3, 0.5323, 1.206, 0)$.

Each sub-domain is filled with a perfect gas of constant $\gamma = 1.4$. Outflow boundary conditions are prescribed and the computation is carried out till the final time $t_f = 0.3$. Density at the final time is presented for the three methods in Fig. 10. For each method on the left side one displays a three-dimensional elevation on the 100×100 mesh while in the right panels 30 isolines are plotted between the minimal density, ρ_m, and maximal one, ρ_M of each method on the 400×400 mesh. The 3D views clearly show that some artificial oscillations on the plateau are generated by the MLP method whereas the MOOD method better preserves the constant states. On the isoline view, we observe that the MOOD-P2 method gives thinner shocks and a finer resolved central peak at $x_1 = x_2 = 0.35$. As expected, this suggests that the MOOD-P2 method is more accurate.

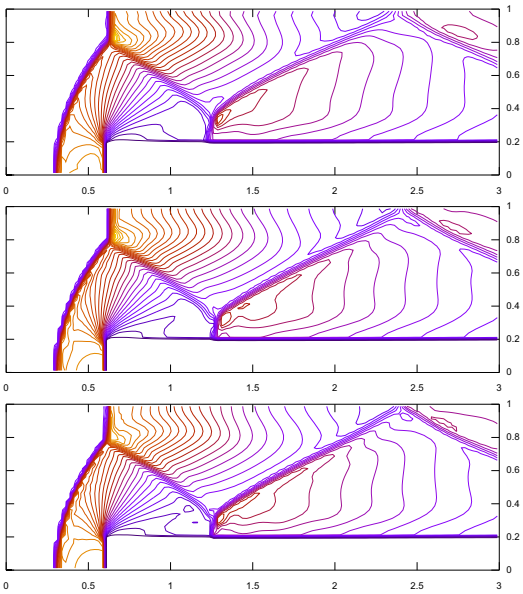


Fig. 11. Mach 3 problem — density solutions with 30 isolines between ρ_m and ρ_M on a 120×40 uniform mesh. Top: MLP method $\rho_m = 0.5437$, $\rho_M = 6.75$ — Middle: MOOD-P1 method $\rho_m = 0.5589$, $\rho_M = 6.58$ — Bottom: MOOD-P2 method $\rho_m = 0.5358$, $\rho_M = 6.047$.

7.3. Mach 3 wind tunnel with a step

The test was initially proposed in [29]. A uniform Mach 3 flow enters in a tunnel which contains a 0.2 unit length step leading to a flow with complex structures of interacting shocks. The wind tunnel is 1 length unit wide and 3 length units long and the step is located at 0.6 length unit from the left-hand side of the domain. At the initial time we consider a perfect gas ($\gamma = 1.4$) with constant density $\rho^0 = 1.4$, uniform pressure $p^0 = 1.0$ and constant velocity $\mathbf{V}^0 = (3,0)$. Reflective boundary conditions are prescribed for the upper and lower sides as well as in front of the step. An inflow condition is set on the left boundary and an outflow condition on the right one. Numerical simulations are carried out till the final time $t_f = 4$.

We plot a series of figures presenting 30 density isolines for two different uniform meshes on which the three methods are tested. We first consider the situation with coarse mesh using 120×40 cells. Fig. 11 represents the density computed with the MLP, the MOOD-P1 and MOOD-P2 methods, respectively on top, middle and bottom panels. It is noticeable that the MOOD method results are the most accurate. The shocks are less diffused and we can already observe the contact discontinuity formation of the upper slip line. With the MLP method, we remark that the formation of a triple point at $x_1 = 1.25$ above the step (at a distance of about 0.1) while the junction point should be exactly on the step interface. With the MOOD-P2 method, the triple point is closer to the interface (half the distance with respect to the MLP case).

We plot the density obtained with a finer uniform mesh of 480×160 cells in Fig. 12. The mesh refinement implies more accurate solutions for any method. Nevertheless MOOD methods still provide the best numerical approximations. However the method does not reveal the Kelvin–Helmholtz instabilities as in [8] as the strict DMP on the density reduces the scheme accuracy along the slip line and consequently increases the numerical dissipation.

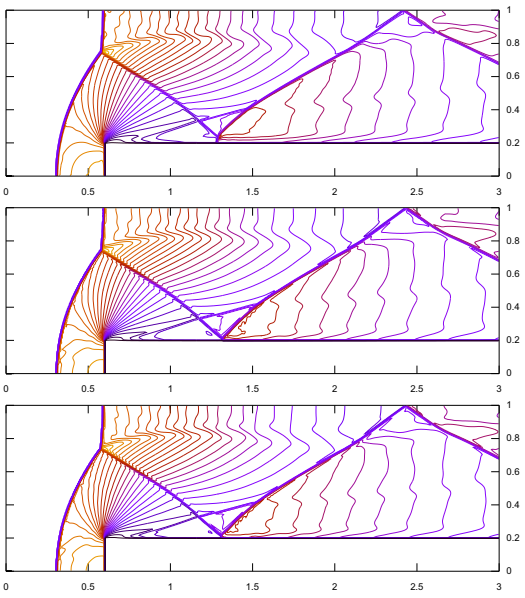


Fig. 12. Mach 3 problem — density solutions with 30 isolines between ρ_m and ρ_M on 480×160 mesh. Top: MLP method $\rho_m = 0.176$, $\rho_M = 6.802$ — Middle: MOOD-P1 method $\rho_m = 0.150$, $\rho_M = 6.483$ — Bottom: MOOD-P2 method $\rho_m = 0.123$, $\rho_M = 6.257$.

7.4. Double Mach reflection of a strong shock

The last problem is the double mach reflection of a strong shock proposed in [29]. This test problem involves a Mach 10 shock which initially makes a 60° angle with a reflecting wall. The air ahead of the shock is at rest and has uniform initial density $\rho^0 = 1.4$ and pressure $p^0 = 1$. A perfect gas with $\gamma = 1.4$ is considered. The reflecting wall lies along the bottom of the domain, beginning at $x_1 = 1/6$. The shock makes a 60° angle with the x_1 axis and extends to the top of the domain at $x_2 = 1$. The short region from $x_1 = 0$ to $x_1 = 1/6$ along the bottom boundary at $x_2 = 0$ is always assigned values for the initial post-shock flow. We prescribe a reflective condition on the bottom part for $x_1 > 1/6$, inflow boundary condition on the left side and outflow condition on the right side. At the top boundary, the boundary conditions are set to describe the exact motion of the Mach 10 flow (see [8]).

First for the three methods, a 30 density isolines top view on the 480×120 uniform mesh using Lax–Friedrich's flux are plotted in Fig. 13. These results have to be compared to results of Fig. 12 in [9] and Fig. 13 in [13]. Then zoomed top views of 50 isolines — between minimal and maximal values, ρ_m and ρ_M, respectively, taken over the results of the three methods on a same mesh — of the results obtained with the HLL flux are plotted in Fig. 14 for the 960×240 uniform mesh on left and for the 1920×480 one on right.

The first Mach stem M1 is connected to the main triple junction point with the incident shock wave and the reflected wave. A slip line is generated from the triple junction point behind the incident shock. A secondary Mach stem M2 also appear and interact with the slip line. As expected, the MOOD-P2 manages to better capture the Mach stem M1 (and M2 when

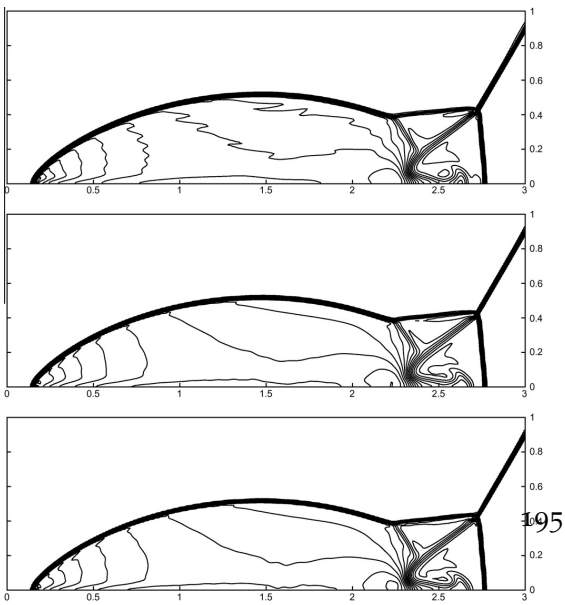


Fig. 13. Double Mach problem on 480×120 — Top: MLP method $\rho_m = 1.40$, $\rho_M = 22.21$ — Middle: MOOD-P1 method $\rho_m = 1.40$, $\rho_M = 20.05$ — Bottom: MOOD-P2 method $\rho_m = 1.40$, $\rho_M = 20.10$.

we employ finer meshes) with respect to the two other methods. The slip line corresponds to a contact discontinuity where the jump of tangential velocity may generate Kelvin Helmholtz instabilities. Usually, the amount of instabilities measures the numerical diffusion influence [22]: large instabilities derive from small numerical diffusion and the number of plane vortexes in the slip line is a qualitative measure of the scheme diffusivity. In our test, even with the finest mesh, no instability is

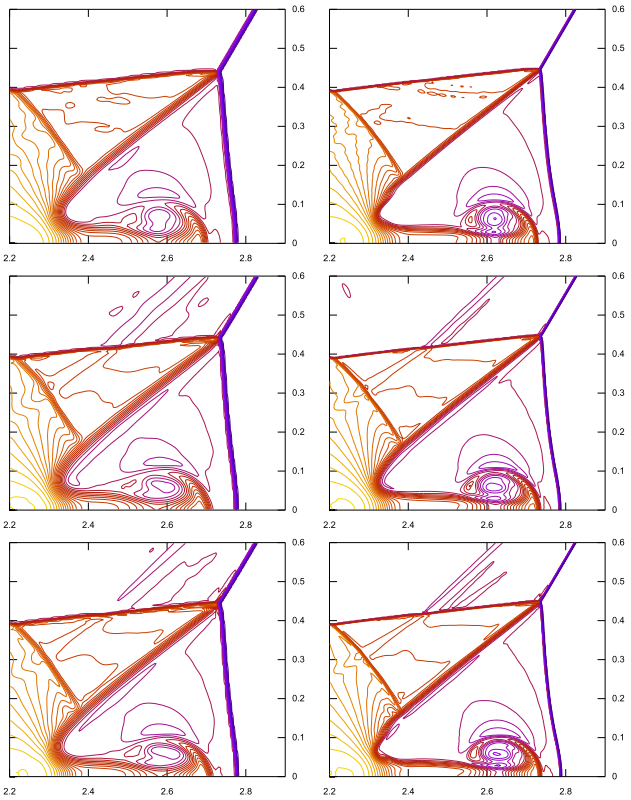


Fig. 14. Double Mach problem on 960×240 (left) and on 1920×480 (right) – zoom on the wave interaction zone – Top: MLP method $\rho_m = 1.400$, $\rho_M = 22.400$ on left and $\rho_m = 1.400$, $\rho_M = 22.68$ on right– Middle: MOOD-P1 method $\rho_m = 1.236$, $\rho_M = 22.550$ on left and $\rho_m = 1.216$, $\rho_M = 22.0$ on right – Bottom: MOOD-P2 method $\rho_m = 1.162$ $\rho_M = 22.800$ on left and $\rho_m = 1.146$, $\rho_M = 21.99$ on right.

Table 10
Computational time ratios between MOOD methods and MLP for different problems.

	Method		
Problem	MLP	MOOD-P1	MOOD-P2
DST	1	1.1	1.73
SBR	1	1.4	2.65
Sod shock tube	1	0.84	1.3
Mach 3 wind	1	1.08	1.6
Double Mach	1	0.99	1.06
Average	1	1.08	1.67

reported. Indeed, the application of a strict DMP reduces the accuracy of the scheme in the vicinity of the slip line maintaining a too large amount of diffusion. Nevertheless, other choices of detection variables could be investigated to reduce the numerical diffusion of contact discontinuities.

7.4.1. Computational cost comparison between MLP, MOOD-P1 and MOOD-P2

In this last section, we give in Table 10 the ratios between MOOD methods computational times and MLP ones. For each test case, computational times are calculated on a given mesh. Numerical experiments show that the ratios are equivalent for finer or coarser meshes.

We recall that these ratios should only be taken as examples because computational times are strongly dependent of implementation and compilation and all runs are carried out on a single core. Table 10 shows that the MOOD-P1 method is slightly more expensive than MLP but gives better results on general meshes. In the scalar case, the difference between ratios of DST and SBR problem are explained by the fact that more iterations during the MOOD procedure, due to more DMP violations, are implied by non-smooth profiles. The MOOD-P2 computational cost is competitive (at most around 2.7 times more expensive than MLP on our numerical experiments) in regard to the observed accuracy improvement, see for instance Figs. 7 or 9.

8. Conclusion and perspectives

This paper presents a high-order polynomial finite volume method named Multi-dimensional Optimal Order Detection (MOOD) for conservation laws. Contrarily to classical high-order methods MOOD procedure is based on a test of the Discrete Maximum Principle (DMP) after an evaluation of the solution with unlimited polynomials. If the DMP property is not fulfilled then the polynomial degree is reduced and the solution is locally re-evaluated. This procedure is repeated up to satisfaction of the DMP which is always achieved after a finite number of iterations.

There are several important features of MOOD method which have to be compared with classical high-order methods, namely

- The MOOD method is an *a posteriori* limiting process, whereas classical limiting strategies perform an *a priori* limitation.
- The MOOD method computes one and only one high-order polynomial per cell and employs it without any limitation.
- Within the same cell the polynomial degree can be different on each edge.
- The MOOD method ensures the Discrete Maximum Principle (DMP) under the first-order CFL constraint.
- The MOOD method has no restriction to deal with higher polynomial degrees and polygonal meshes.

Two-dimensional numerical results are provided for advection and the Euler equations problems on regular and highly non-regular quadrangular meshes. They clearly show that MOOD method presents some promising good behaviors. The second-order MOOD method is at least equivalent to a second-order multi-dimensional MUSCL method on uniform grids but produces better results on non-uniform ones. A third-order version of MOOD has been shown to be effective on regular and non-regular solutions for a small extra computational effort.

This paper is the first one presenting the MOOD concept and extensions are currently under investigations, as instance the behavior of the MOOD with polynomials of degree greater than two on polygonal meshes.

References

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.
[2] T.J. Barth, Numerical methods for conservation laws on structured and unstructured meshes, VKI March 2003 Lectures Series.
[3] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Report 89-0366, 1989.
[4] T. Buffard, S. Clain, Monoslope and multislope MUSCL methods for unstructured meshes, J. Comput. Phys. 229 (2010) 3745–3776.
[5] S. Clain, Finite volume L^∞-stability for hyperbolic scalar problems, preprint HAL. Available from: <http://hal.archives-ouvertes.fr/hal-00467650/fr/>.
[6] S. Clain, V. Clauzon, L^∞ stability of the MUSCL methods, Numer. Math. 116 (2010) 31–64.
[7] I. Christov, B. Popov, New non-oscillatory central schemes on unstructured triangulations for hyperbolic systems of conservation laws, J. Comput. Phys. 227 (2008) 5736–5757.

[8] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[9] O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, J. Comput. Phys. 144 (1998) 194–212.
[10] A. Harten, S. Osher, Uniformly highGorder accurate nonoscillatory schemes I, SIAM J. Numer. Anal. 24 (1987) 279–309.
[11] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly highGorder accurate nonoscillatory schemes III, J. Comput. Phys. 71 (1987) 279–309.
[12] M.E. Hubbard, Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids, J. Comput. Phys. 155 (1999) 54–74.
[13] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[14] V.P. Kolgan, Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics, Transactions of the Central Aerohydrodynamics Institute, vol. 3, No. 6, Russian, pp. 68–77, 1972.
[15] V.P. Kolgan, Finite-difference schemes for computation of three dimensional solutions of gas dynamics and calculation of a flow over a body under an angle of attack, 1975, Transactions of the Central Aerohydrodynamics Institute, Russian, vol. 6, No. 2, pp. 1–6.
[16] V.P. Kolgan, Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics, J. Comput. Phys. (2010), doi:10.1016/j.jcp.2010.12.033.
[17] Randall J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (1996) 627–665.
[18] R. Liska, B. Wendroff, Comparison of several difference schemes on 1D and 2D test problems for the Euler equations, SIAM J. Sci. Comput. 25 (2003) 995–1017.
[19] C.F. Ollivier-Gooch, Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction, J. Comput. Phys. 133 (1997) 6–17.
[20] S.-H. Yoon, C. Kim, K.H. Kim, Multi-dimensional limiting process for three-dimensional flow physics analyses, J. Comput. Phys. 227 (2008) 6001–6043.
[21] J.S. Park, S.-H. Yoon, C. Kim, Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids, J. Comput. Phys. 229 (2010) 788–812.
[22] J. Shi, Y-T Zhang, C-W Shu, Resolution of high order WENO schemes for complicated flow structures, J. Comput. Phys. 186 (2003) 690–696.
[23] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, SIAM Rev. 51 (1) (2009) 82–126.
[24] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing scheme, J. Comput. Phys. 77 (1988) 439–471.
[25] C.W. Schulz-Rinne, J.P. Collins, H.M. Glaz, Numerical solution of the Riemann problem for two-dimensional gas dynamics, SIAM J. Sci. Comput. 14 (1993) 1394–1414.
[26] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, second ed., Springer, 1999.
[27] B. Van Leer, Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme, J. Comput. Phys. 14 (1974) 361–370.
[28] W.R. Wolf, J.L.F. Azevedo, High-order ENO and WENO schemes for unstructured grids, Int. J. Numer. Methods Fluids 55 (10) (2007) 917–943.
[29] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.

Computers & Fluids 64 (2012) 43–63

Contents lists available at SciVerse ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Improved detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials

Steven Diot [a,*], Stéphane Clain [a,b,c], Raphaël Loubère [a]

[a] Institut de Mathématiques de Toulouse, Université de Toulouse, France
[b] Departamento de Matemática e Aplicações, Campus de Gualtar, 4710-057 Braga, Portugal
[c] Centro de Matemática, Campus de Azurém, 4800-058 Guimarães, Portugal

ARTICLE INFO

Article history:
Received 1 November 2011
Received in revised form 7 May 2012
Accepted 8 May 2012
Available online 16 May 2012

Keywords:
Finite volume
High-order
Conservation law
Polynomial reconstruction
Limitation
Polygonal
Non-conformal
Unstructured
Euler
MOOD
Positivity-preserving

ABSTRACT

This paper extends the MOOD method proposed by the authors in [A high-order finite volume method for hyperbolic systems: Multi-Dimensional Optimal Order Detection (MOOD), J Comput Phys 2011;230: 4028–50], along two complementary axes: extension to very high-order polynomial reconstruction on non-conformal unstructured meshes and new detection criteria. The former is a natural extension of the previous cited work which confirms the good behavior of the MOOD method. The latter is a necessary brick to overcome limitations of the discrete maximum principle used in the previous work. Numerical results on advection problems and hydrodynamics Euler equations are presented to show that the MOOD method is effectively high-order (up to sixth-order), intrinsically positivity-preserving on hydrodynamics test cases and computationally efficient.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

In a recent paper [7], an original high-order method, namely the Multi-dimensional Optimal Order Detection (MOOD) method, has been introduced to provide up to third-order approximations to hyperbolic scalar or vectorial solutions for two-dimensional geometry. The present article deals with new extensions of the method to general unstructured 2D meshes and to sixth-order convergence in space. Classical high-order reconstructions such as MUSCL or ENO/WENO methods are based on an *a priori* limiting procedure to achieve stability property. The MOOD method follows a fundamentally different way since the limiting procedure (polynomial degree reduction for instance) is achieved *a posteriori* and provides the optimal local polynomial reconstruction which satisfies given stability criteria.

The quest [41] of the (very) high-order schemes starts in the early 1970s with the pioneer works of Van-Leer [42] and Kolgan [22–24]. Since this date, a large literature was dedicated to the limited reconstruction methods for structured and unstructured meshes. Several strategies became very popular due to their intrinsic simplicity such that the MUSCL method [3,5,6,19,25,31] or their efficiency to achieve very high-order accuracy such that the ENO/WENO method [1,20,28,29,35,36,45,17,18,34,48,40], the discontinuous Galerkin method [8–11], the ADER method [13,38,39,14,15], the residual distribution scheme [2,12,32] and the spectral method [16,43,44].

While second-order methods do not require particular cautions, dealing with higher-order methods leads to at least three specific difficulties which, up to our knowledge, are not always clearly identified. First point one should not consider the mean value of a function equivalent to the cell centroid value as it is often done in the MUSCL community. The point is straightforward to overcome but important to notice for newcomers in the field of higher-order numerical schemes. Second point, for vectorial problems the reconstruction process must be done on mean values of the conservative variables and not on non-linear combinations of them. This point is often implied in the classical ENO/WENO papers but is rarely clearly stated and this may mislead newcomers in the high-order community because the order of accuracy discrepancy can be missed depending on the numerical tests used. Contrarily one proposes the isentropic vortex in motion test case to numerically prove that if the primitive variables are used for the reconstruction process then very high-order of accuracy cannot be reached. Third point the Discrete Maximum Principle property on mean values should not be used anymore as a guide line for limitation. We propose in this paper to overcome this difficulty by a new limiting criteria (or Detection Criteria in the MOOD jargon) adapted to provide a full high-order method still maintaining robust stability. Simple examples are introduced within the text when some difficulties related to these points are to be expected.

The basic idea of the MOOD method consists of determining the higher polynomial degree of each local cell still satisfying some stability restrictions. To this end, an iterative process is developed. We perform a local polynomial reconstruction of degree d_i for each cell K_i at the current time t^n and compute a candidate solution at time t^{n+1} without any limiting features. Then a detecting procedure is carried out to check the cells which do not respect the stability criteria and we reduce the local polynomial degree to obtain a better stability. We state that the method is *a posteriori* since the limiting procedure (namely the polynomial degree reduction) is performed after the candidate solution computation. Such a situation is very useful to test the admissibility of the solution. Furthermore, one has to carry out the limiting algorithm if, and only if, it is necessary while the traditional *a priori* method performs unnecessary limitation.

In this paper we propose extensions of the MOOD method which take into account the three difficulties mentioned above. More precisely different detection processes both for the advection and hydrodynamics equations are developed. We numerically prove that these detection processes provide the effective higher-order of accuracy on smooth profiles (up to sixth-order). Moreover we show that for the hydrodynamics equations the method is positivity-preserving by construction and we numerically observe this behavior. The test case have been carried out on non-regular, polygonal and non-conformal meshes and the last test case of the paper show the ability of the MOOD method to simulate complex physics from an experimental set-up of the impact of a shock wave on a cylindrical cavity.

The paper is organized as follows. Section 2 is dedicated to the generic framework used to describe the MOOD method where the high-order finite volume scheme is presented. Several obstacles to achieve high-order reconstruction are pointed out and the polynomial reconstruction based on the mean value approximation is detailed. In Section 3, we introduce new criteria to obtain very high-order accurate schemes still preserving local stability. To show the MOOD method efficiency, numerical tests both for the scalar and the vectorial case are carried out in Section 4. We mainly focus on the method accuracy and its robustness. We draw some remarks and future developments in the last section.

2. The MOOD method

2.1. General concept

The MOOD method is a generic procedure that solves multi-dimensional hyperbolic system of equations on an unstructured grid in the Eulerian framework. Given different numerical finite volume schemes the MOOD method provides an optimal choice for each computational cell by mitigating accuracy *vs* robustness. From an abstract point of view the MOOD algorithm involves two main ingredients: An ordered list of numerical schemes and a set of constraints with detection criteria which defines the desirable properties the numerical solution should have.

The over-topping numerical scheme represents the *best* scheme one would like to employ. Usually this scheme is the most accurate but less robust one. At the very end of the list lays the least accurate but more robust scheme which is assumed to be satisfactory in all possible situations due to the stabilization effect generated by its intrinsic numerical dissipation. In this paper the list is composed of a robust first-order scheme (an upwind or a Rusanov, HLL, HLLC scheme as instance) while several second or higher-order schemes using polynomial reconstructions compose an ordered list of desirable schemes (see Fig. 1 for instance). The second ingredient is the detecting procedure of a set of constraints which determine the local eligibility of the solution for each cell.

We recall that discontinuous solutions may not be handled with high-order reconstructions since local spurious and unphysical oscillations may take place. The low-order numerical scheme should be used to prevent the numerical approximations from oscillating and force to respect some constraints or mathematical properties that depend on equations under consideration. The numerical solution is considered as eligible if it fulfills given properties. As instance the positivity of certain variables such as density or pressure in hydrodynamics equations or the Discrete Maximum Principle for advection equation shall be considered.

In this paper the *k*th numerical scheme of the list is a finite volume scheme using unlimited piecewise polynomial reconstruction of degree *k*. Ultimately this scheme has a *k* + 1th-order of accuracy for smooth solutions. Consequently the LO scheme is the generic firs-order finite volume scheme and the HO-1 scheme corresponds to an unlimited MUSCL method.

The core of the MOOD method is a loop over the cells to determine the optimal polynomial degree one can safely use to produce an eligible numerical solution. It amounts to select a numerical method in the ordered list of Fig. 1.

To this end, given a generic cell K_i and its neighbor cells K_j having edge e_{ij} in common, we first recall two definitions introduced in [7] and then give a new one to extend the MOOD concept:

- d_i is the Cell Polynomial Degree (CellPD) which represents the degree of the polynomial reconstruction on K_i.
- $d_{ij} = d_{ji} = \min(d_i, d_j)$ are the Edge Polynomial Degrees (EdgePD) corresponding to the degrees of the polynomial reconstructions used to compute approximations of the solution on edge e_{ij}.

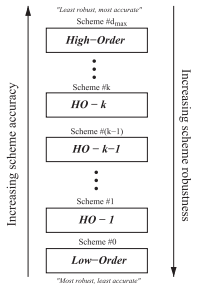


Fig. 1. Schematic representation of an ordered list of numerical schemes used in the MOOD method. The bottom scheme is the most robust but least accurate one denoted "Low-Order". All over-topping schemes are successively more accurate but less robust. The MOOD method is designed to choose the more adapted scheme for each cell of the computational domain.

* Corresponding author.
E-mail addresses: steven.diot@math.univ-toulouse.fr (S. Diot), clain@math.uminho.pt (S. Clain), raphael.loubere@math.univ-toulouse.fr (R. Loubère).

0045-7930/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compfluid.2012.05.004

- $\mathcal{A}$ is a set of prescribed physical and/or stability constraints. If for each cell K_i the mean values of the numerical solution fulfill the constraints then the numerical solution is said to be $\mathcal{A}$-eligible.

The last item concerns the detecting procedure to distinguish if a candidate solution is eligible according to a set of constraints. In practice we decrement the d_i for any cell K_i which does not respect all the constraints. Such a cell is called *problematic*. Moreover since neighbor cells fluxes may be affected by this process, the decrementing is spread over the direct neighborhood. Such a polynomial degree decrementing for a problematic cell is repeated up to a $d_i > 0$ for which the set of constraints is fulfilled or to $d_i = 0$. At that ultimate step the robust and diffusive LO scheme is employed and its first-order solution is always taken as valid. In other words unlike traditional high-order schemes (using *a priori* limiting procedure), we introduce an *a posteriori* detecting procedure where the decision to alter the polynomial degree is carried out after computing the candidate solution.

We finally highlight that such a procedure may be interpreted as a *try and fail* algorithm. Such a generic strategy might be adapted to other classes of method such as the Discontinuous Galerkin method and detect the best polynomial degree in each cell or Finite Element method and detect the most appropriate finite element one can employ in a cell.

2.2. Framework

Let us consider a generic autonomous hyperbolic equation defined on a domain $\Omega \subset \mathbb{R}^2$, $t > 0$ which casts in the conservative form

$$\partial_t U + \nabla \cdot F(U) = 0, \tag{1a}$$
$$U(\cdot,0) = U_0, \tag{1b}$$

where $U = U(\mathbf{x},t)$ is the vector of unknown functions, $\mathbf{x} = (x,y)$ denotes a point of Ω, t is the time, F is the physical flux function and U_0 is the initial condition. Boundary conditions shall be prescribed in the following.

We assume that the computational domain Ω is a polygonal bounded set of $\mathbb{R}^2$ divided into convex polygonal cells K_i, $i \in \mathcal{E}_{el}$, $\mathbf{c}_i$ being the cell centroid and $\mathcal{E}_{el}$ the cell index set. For each boundary edge, $K_i \cap \partial\Omega$, we introduce a virtual cell K_j with $j \notin \mathcal{E}_{el}$ which represents the exterior side of Ω and denote by $\mathcal{E}_{bd}$ the index set of all virtual cells. $\widetilde{\mathcal{E}}_{el} = \mathcal{E}_{el} \cup \mathcal{E}_{bd}$ is the index set of all cells. This notation avoids a special treatment for boundary edges in the scheme, and provides a natural notation for ghost cells should they exist or not.

For each cell K_i, one denotes by e_{ij} the common edge between K_i and K_j, with $j \in \underline{\nu}(i) \subset \widetilde{\mathcal{E}}_{el}$, $\underline{\nu}(i)$ being the index set of all the elements which share an edge with K_i. The extended neighborhood is represented by the index set $\bar{\nu}(i) \subset \widetilde{\mathcal{E}}_{el}$ of all K_j such that $K_i \cap K_j \neq \emptyset$ (see Fig. 2).

Moreover $|K_i|$ and $|e_{ij}|$ measure the surface of K_i and the length of e_{ij} respectively while $\mathbf{n}_{ij}$ is the unit outward normal vector to e_{ij} pointing from K_i to K_j. At last, q^r_{ij}, $r = 1,\dots,R$ represent the Gaussian quadrature points employed for numerical integration on edge e_{ij}.

The generic first-order explicit finite volume scheme is given by

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \mathbb{F}\left(U_i^n, U_j^n, \mathbf{n}_{ij}\right), \tag{2}$$

where $\mathbb{F}\left(U_i^n, U_j^n, \mathbf{n}_{ij}\right)$ is a numerical flux which satisfies the classical properties of consistency and monotonicity. To provide higher-order accuracy, we substitute in Eq. (2) the first-order approximation

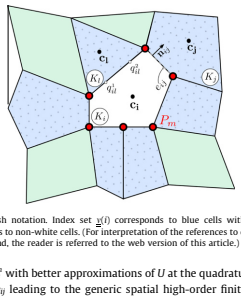


Fig. 2. Mesh notation. Index set $\underline{\nu}(i)$ corresponds to blue cells with dots, $\bar{\nu}(i)$ corresponds to non-white cells. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

U_i^n and U_j^n with better approximations of U at the quadrature points of edge e_{ij} leading to the generic spatial high-order finite volume scheme

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}\left(U^n_{ij,r}, U^n_{ji,r}, \mathbf{n}_{ij}\right), \tag{3}$$

where $U^n_{ij,r}$ and $U^n_{ji,r}$, $r = 1,\dots,R$ are high-order approximations of U at quadrature points $q^r_{ij} \in e_{ij}$, $r = 1,\dots,R$ on both sides of edge e_{ij} and ξ_r denote the quadrature weights.

For the sake of simplicity, let us write the scheme under the compact form

$$U_h^{n+1} = U_h^n + \Delta t \mathcal{H}^R(U_h^n), \tag{4}$$

with $U_h^n = \sum_{i \in \mathcal{E}_{el}} U_i^n 1_{K_i}$ the constant piecewise approximation of function U and operator $\mathcal{H}^R$ being defined as

$$\mathcal{H}^R(U_h^n) := -\sum_{i \in \mathcal{E}_{el}} \left(\sum_{j \in \underline{\nu}(i)} \frac{|e_{ij}|}{|K_i|} \sum_{r=1}^{R} \xi_r \mathbb{F}\left(U^n_{ij,r}, U^n_{ji,r}, \mathbf{n}_{ij}\right) \right) 1_{K_i}. \tag{5}$$

Finally to provide a high-order method in time, we use the third-order TVD Runge–Kutta method (RK3, see [36]) which corresponds to a convex combination of three explicit steps

$$U_h^{n+1} = \frac{U_h^n + 2U_h^{(3)}}{3} \quad \text{with} \quad \begin{cases} U_h^{(1)} = U_h^n + \Delta t \mathcal{H}^R(U_h^n) \\ U_h^{(2)} = U_h^{(1)} + \Delta t \mathcal{H}^R\left(U_h^{(1)}\right) \\ U_h^{(3)} = \widetilde{U}_h^{(2)} + \Delta t \mathcal{H}^R\left(\widetilde{U}_h^{(2)}\right) \end{cases} \tag{6}$$

where $\widetilde{U}_h^{(2)}$ is the convex combination $\left(3U_h^n + U_h^{(2)}\right)/4$.

2.3. Arbitrary degree polynomial reconstruction

In the introduction we have reminded one classical obstacle to reach higher-order of accuracy when polynomial reconstruction is to be used. It is well-known that the mean value U_i of a regular function U on K_i is approximated by the value of the solution at the cell centroid, $U(\mathbf{c}_i)$, with an error of $O(h^2)$ where h represents the characteristic length of the cell. It results that any reconstruction based on geometrical arguments using $U(\mathbf{c}_i)$ in place of U_i can only provide second-order approximation.

Therefore as classical higher-order finite volume methods the MOOD method is based on polynomial reconstructions from mean values on cells. Let us consider a generic reconstructed polynomial of degree d, given mean values U on a generic cell K, under the form

$$\widetilde{U}(\mathbf{x};\mathbf{d}) = U + \sum_{1 \leq |\boldsymbol{\alpha}| \leq d} \mathcal{R}_{\boldsymbol{\alpha}} \left((\mathbf{x} - \mathbf{c})^{\boldsymbol{\alpha}} - \frac{1}{|K|} \int_K (\mathbf{x} - \mathbf{c})^{\boldsymbol{\alpha}} d\mathbf{x} \right), \tag{7}$$

where $\mathbf{c}$ is the centroid of K, $\mathbf{x}$ a generic point in K and $\mathcal{R}_{\boldsymbol{\alpha}}$ are the unknowns polynomial coefficients where $\boldsymbol{\alpha} = (\alpha_x, \alpha_y) \in \mathbb{N}^2$ is a multi-index with $|\boldsymbol{\alpha}| = \alpha_x + \alpha_y$. Note that by construction, the mean value on K of the polynomial function is equal to U since the integral over K of the term between parenthesis in (7) vanishes. It thus fulfills the conservation property on K.

There exist several techniques [1,28] to determine the coefficients $\mathcal{R}_{\boldsymbol{\alpha}}$. Here, we consider a least square approximation of neighbor mean values U_j where K_j belongs to a compact stencil $\mathcal{S}(K)$. It amounts to minimizing the functional

$$E(\widetilde{U}) = \sum_{j \in \mathcal{S}(K)} \omega_k \left[\frac{1}{|K_j|} \int_{K_j} \widetilde{U} d\mathbf{x} - U_j \right]^2, \tag{8}$$

where ω_k are positive weights used to provide a better condition number. In particular, the condition number of the associated linear system dependents on the spatial characteristic length thus we use the solution proposed in [28] to overcome this problem.

In practice, we do not directly solve the symmetric linear system associated with the minimization problem. Instead we use the technique from [4,29] where an over-determined linear system is solved in a least-squares sense with a QR decomposition using Householder transformations.

The reconstructed polynomial $\widetilde{U}$ is thus exact for any polynomial function of degree lower than d which provides the consistency of the reconstruction method and further the status of a $(d+1)^{th}$-order numerical method.

Remark 1. In 2D, at least $\mathcal{N}(d) = (d+1)(d+2)/2 - 1$ neighbors are needed to provide the minimal number of equations. However for the sake of robustness more cells are involved. In details, we use at least 5 cells for $d = 1$, 8 cells for $d = 2$, 16 cells for $d = 3$, 20 cells for $d = 4$ and 26 cells for $d = 5$.

Remark 2. In the introduction we have stated that in the general case one should not identify the mean value of a non-linear combination with the non-linear combination of mean values. Let ρ and ϕ be two regular functions on cell K_i and ρ_i, ϕ_i, $(\rho\phi)_i$, denote their respective exact mean values. A Taylor expansion with respect to the centroid of the cell gives $(\rho\phi)_i = \rho_i\phi_i + O(h^2)$. For instance let us consider the one-dimensional variables ρ, ϕ and $(\rho\phi)$ and their mean values on cell $K_1 = [0,h]$

$$\rho(x) = 1 + x, \quad \phi(x) = 1 - x, \quad (\rho\phi)(x) = 1 - x^2,$$
$$\rho_1 = 1 + \frac{h}{2}, \quad \phi_1 = 1 - \frac{h}{2}, \quad (\rho\phi)_1 = 1 - \frac{h^2}{3}.$$

We then deduce that $|(\rho\phi)_1 - \rho_1\phi_1| = h^2/12$ leading to a second-order error. As instance it is well known that for Euler system of equations the non-linear transformation of the conservative mean values into primitive ones introduces a second-order error in the general case.

2.4. Algorithm

Let us assume that we have access to a given sequence $U_h^n = \left(U_i^n\right)_{i \in \mathcal{E}_{el}}$ of mean value approximations at time t^n, the goal is to build an eligible sequence $U_h^{n+1} = \left(U_i^{n+1}\right)_{i \in \mathcal{E}_{el}}$ at time $t^{n+1} = t^n + \Delta t^n$ in the sense that each approximation U_i^{n+1} respects a set of constraints $\mathcal{A}$. We only consider here a forward Euler time step without loss of generality. The MOOD method algorithm is the following:

1. *Initialization at* t^n. The MOOD procedure starts by initializing the CellPD to $d_i = d_{max}$ and by computing the coefficients of the polynomial reconstruction $\widetilde{U}_i(\mathbf{x};d_i)$ on each cell.
2. *Evaluation of EdgePD and values at Gauss points.* We compute the EdgePD d_{ij} on each edge and use polynomial function $\widetilde{U}_i(\mathbf{x};d_{ij})$ and $\widetilde{U}_j(\mathbf{x};d_{ij})$ to compute approximations of U at Gaussian points on e_{ij}.
3. *Computation of candidate solution* U_h^n. Numerical fluxes are computed using the reconstructed solution at Gauss points and one time step is carried out to provide a candidate U_h^n at time $t^{n+1} = t^n + \Delta t^n$.
4. *Check* U_i^n *for* $\mathcal{A}$*-eligibility.* If $d_i \neq 0$ we check the $\mathcal{A}$-eligibility of each mean value U_i^n with respect to the constraints set $\mathcal{A}$. In the case U_i^n is not $\mathcal{A}$-eligible then CellPD d_i is decremented. If all cells are $\mathcal{A}$-eligible then the candidate solution is valid and we set $U_h^{n+1} = U_h^n$ else the solution is recomputed following steps 2–4.

Remark 3. Only cells K_i where CellPD has been decremented and their neighbors in the compact stencil $\underline{\nu}(i)$ have to be re-updated. Consequently only these cells will have to be checked for the next iterations of the MOOD procedure within the current time step. This dramatically reduces computational cost.

Remark 4. Since polynomial reconstruction is costly in CPU time and memory, we proposed in [7] to truncate $\widetilde{U}_i(\cdot;d_{max})$ to obtain lower-order polynomials. However we found that for $d_{max} > 2$ this technique implies nondesirable behavior on discontinuous profiles as the reconstruction stencil remains large.

Moreover numerical experiments show that a one-by-one degree decrementing leads to avoidable computational effort since the decrementing procedure is usually performed around discontinuities. We thus slightly modify the decrementing algorithm by jumping from $d = d_{max}$ to $d = 2$ and then from $d = 2$ to $d = 0$ if needed. This also reduces the computational effort while providing equivalent results on a wide range of test cases compared to a one-by-one decrementing.

Remark 5. Polynomial reconstruction on boundary cells are treated using ghost cells in order to be consistent with the prescribed boundary conditions.

The major difficulty remains to determine a list of constraints which both provides a very high accurate solution while avoids numerical artifacts such as spurious oscillations in the vicinity of discontinuity. This is the purpose of the next section.

3. Detection process

The list of constraints $\mathcal{A}$ corresponds to eligible criteria that the numerical approximation has to fulfill. To end, detection process is necessary to list where the candidate numerical solution fails to respect the constraints. Such process must be very carefully designed to preserve high accuracy for regular solutions whereas discontinuities should be treated with the lower order scheme to avoid non-physical oscillations. The first subsection deals with the advection problem and a new detection process called *u2* and based on a smoothness detector. In the following subsection the Euler system is considered: Two detection processes are proposed and we show the positivity-preserving property of the MOOD method.

3.1. Advection problem: the u2 detection process

Solutions of autonomous scalar hyperbolic problems satisfy the Maximum Principle property. Such a property is also valid for

advection problem with divergence free velocity. Therefore the Discrete Maximum Principle (DMP) seems to be a good candidate to detect problematic cells. Unfortunately, as mentioned in the introduction, the strict DMP applied to mean values reduces the order of accuracy to two (see the appendix for an example), and thus cannot be used alone. Classical studies show that the accuracy discrepancy only occurs at extrema [33,30,21]. We will then mainly focus on extrema since the DMP detection process is still relevant where the solution is locally monotone. We propose the relaxation of the strict DMP at smooth extrema in order to avoid accuracy discrepancy. This leads to the introduction of an additional procedure to detect smooth extrema. Notice that in (W)ENO type of methods the DMP is not strictly enforced which implies that extrema are well approximated and consequently arbitrary high-order of accuracy is achieved. The first detection criteria is the DMP: No polynomial degree decrementing is performed for cells where the DMP is satisfied. Let us now consider a cell K_i where U_i° does not fulfill the DMP. Two situations may arise whether we deal with a discontinuity or a smooth extrema. The major difficulty is to give a concrete definition of the concept of a *smooth extrema* from a numerical point of view. Actually a function may be considered irregular for a coarse mesh but regular with a finer one. We try to overcome this difficulty by introducing the following definition.

Definition 6. Let K_i be a cell and $\widetilde{U}_i = \widetilde{U}_i(.;2)$ a polynomial reconstruction of degree 2 for an underlying function U. We define the second derivatives in x and y directions by $\mathcal{X}_i = \partial_{xx}\widetilde{U}_i \in \mathbb{R}$ and $\mathcal{Y}_i = \partial_{yy}\widetilde{U}_i \in \mathbb{R}$. We will refer to these second derivatives as "curvatures".

For all cell K_j, $j \in \underline{\nu}(i)$, we define the maximal and minimal curvatures as

$$\mathcal{X}_i^{min} = \min_{j\in\underline{\nu}(i)}(\mathcal{X}_i, \mathcal{X}_j) \quad \text{and} \quad \mathcal{X}_i^{max} = \max_{j\in\underline{\nu}(i)}(\mathcal{X}_i, \mathcal{X}_j),$$
$$\mathcal{Y}_i^{min} = \min_{j\in\underline{\nu}(i)}(\mathcal{Y}_i, \mathcal{Y}_j) \quad \text{and} \quad \mathcal{Y}_i^{max} = \max_{j\in\underline{\nu}(i)}(\mathcal{Y}_i, \mathcal{Y}_j).$$

We now introduce the new detection criterion to select smooth extrema.

Definition 7. A numerical solution U_i° in cell K_i which violates the DMP is nonetheless eligible if

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > 0 \quad \text{and} \quad \mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > 0, \tag{9}$$
$$\frac{|\mathcal{X}_i^{min}|}{|\mathcal{X}_i^{max}|} \geqslant 1-\varepsilon_i \quad \text{and} \quad \frac{|\mathcal{Y}_i^{min}|}{|\mathcal{Y}_i^{max}|} \geqslant 1-\varepsilon_i, \tag{10}$$

where ε_i is a cell dependent parameter defined by

$$\varepsilon_i = (\Delta x_i)^{\frac{m}{2}}, \quad \text{with } \Delta x_i = |K_i|^{\frac{1}{m}},$$

m being the spatial dimension ($m = 2$ here).

Such a detection criterion is motivated by the following considerations. For a given mesh, the solution is locally considered as non-oscillating if condition (9) is fulfilled meaning that, at the numerical level, the "curvatures" of the $\mathbb{P}_2$ approximation have the same sign.

Moreover for a given mesh, the solution is considered locally $\mathcal{C}^2$ from a numerical point of view if condition (10) is fulfilled. The parameter ε is a mesh dependent coefficient which prescribes the tolerance. Such criteria verifies if the "curvatures" are almost identical in the vicinity of cell K_i with respect to the local characteristic space length Δx_i.

The choice of ε derives from numerous tests. In fact our numerical experiments have shown that ε scales like a cell dependent characteristics length to a power depending on the dimension of space (tests have been carried out in 1D and 2D). It seems to the authors to be the best compromise to gain a very high-order of convergence while maintaining reasonable monotonicity. Finally we remark that at the limit $\varepsilon = 0$ we recover the DMP.

The set of constraints $\mathcal{A}$ for advection equation is thus constituted by the DMP relaxed by the smooth extrema detector described above. The detection process is called *u2* detection in reference to the second-order derivatives and is summarized in the sequel.

Being given a sequence $U_h^\circ = (U_i^\circ)_{i\in\mathcal{E}_d}$, the *u2* detection procedure in the case of the advection problem is given by the following algorithm.

1. The DMP criterion is first checked on each cell K_i
$$\min_{j\in\overline{\nu}(i)}\left(U_i^n, U_j^n\right) \leqslant U_i^\circ \leqslant \max_{j\in\overline{\nu}(i)}\left(U_i^n, U_j^n\right). \tag{11}$$
2. If U_i° does not satisfy (11) then.
   a. Compute $\mathcal{X}_k, \mathcal{Y}_k$ for $k \in \underline{\nu}(i) \bigcup \{i\}$ and coefficient ε_k,
   b. Check criteria (9) and (10). If cell i is not a smooth extrema then d_i is decremented, else U_i° is eligible.

3.2. Euler system: two detection processes and positivity-preserving

The compressible hydrodynamics Euler system of equations is the following hyperbolic unsteady non-linear system involving conservation of mass, momentum and total energy

$$\partial_t\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix} + \partial_x\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{pmatrix} + \partial_y\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{pmatrix} = 0. \tag{12}$$

The primitive variables are the density ρ, the velocity $\mathbf{U} = (u, v)$ and the pressure p. The pressure is linked to two thermodynamical variables such as density and specific internal energy ε through an Equation Of State (EOS) $p = p(\rho, \varepsilon)$. As instance the classical ideal gas law states that $p = (\gamma - 1)\rho\varepsilon$ where γ is the ratio of specifics heats. Moreover the total energy E is such that $E = \rho(\varepsilon + 1/2|\mathbf{U}|^2)$.

Even if the DMP property is used in most of limiting procedures (MUSCL technique as instance), the DMP property does not make sense in the case of the Euler system, for the density or the total energy for instance, since the velocity is not divergence free. Consequently we cannot rely only on DMP. We propose here two detecting procedures which we have been widely experimented and present in the next sections the pros and cons of such procedures.

3.2.1. Physical Admissible Detection (PAD)

The first and minimal detection criteria consists of ensuring the physical meaningfulness of the primitive variables, namely positivity of density and pressure.

Then the set of constraints $\mathcal{A}$ are used to test if the candidate solution satisfies $\rho_i^\circ > 0$ and $p_i^\circ > 0$. Note that p_i° is not a conservative variable and derives from nonlinear combinations of conservative ones. The PAD algorithm is the following.

1. The Physical Admissibility criterion is first checked on each cell K_i
$$\rho_i^\circ > 0, \quad p_i^\circ > 0. \tag{13}$$
2. If the PAD criterion is not satisfied then d_i is decremented, else U_i° is eligible.

The PAD procedure only consists of maintaining the physical meaningfulness of the numerical approximation. In other words, the high-order MOOD method coupled with the PAD Detection

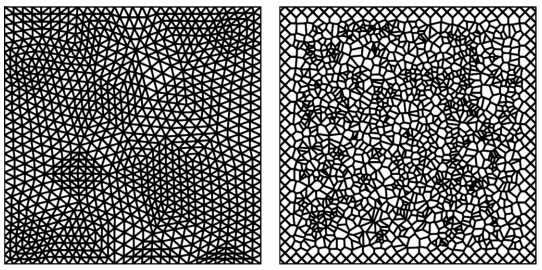


Fig. 3. Example of Delaunay (left) and Voronoi (right) meshes for the DST problem.

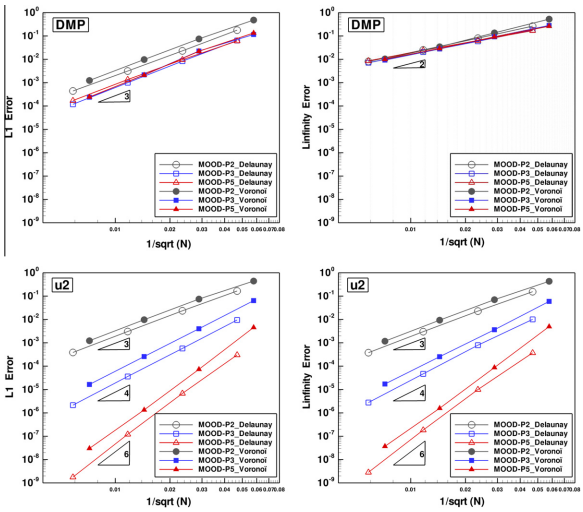


Fig. 4. Error curves for the DST problem for series of Delaunay meshes (empty symbols) and of Voronoi meshes (filled symbols) for the DMP detection process (top) and the *u2* one (bottom).

Table 1
L^1 and L^∞ errors and convergence rate for the DST problem for the MOOD method with DMP and *u2* detection process.

Deg./type	Cell Nb	DMP detec. process				*u2* detec. process			
		L^1 Error		L^∞ Error		L^1 Error		L^∞ Error	
$\mathbb{P}_2$/Delaunay	456	1.775E−01	–	2.629E−01	–	1.656E−01	–	1.549E−01	–
	1824	2.303E−02	2.95	8.016E−02	1.71	2.351E−02	2.82	2.283E−02	2.76
	7296	3.142E−03	2.87	2.522E−02	1.67	3.049E−03	2.95	2.995E−03	2.93
	29184	4.391E−04	2.84	8.082E−03	1.64	3.870E−04	2.98	3.784E−04	2.98
$\mathbb{P}_2$/Voronoi	300	4.804E−01	–	5.278E−01	–	4.423E−01	–	4.339E−01	–
	1200	7.483E−02	2.68	1.359E−01	1.96	7.482E−02	2.56	7.070E−02	2.62
	4800	9.779E−03	2.94	3.432E−02	1.99	9.788E−03	2.93	9.348E−03	2.92
	19200	1.244E−03	2.97	1.039E−02	1.72	1.233E−03	2.99	1.176E−03	2.99
Expected order			3		3		3		3
$\mathbb{P}_3$/Delaunay	456	6.383E−02	–	1.801E−01	–	9.474E−03	–	1.007E−02	–
	1824	8.369E−03	2.93	5.920E−02	1.61	5.751E−04	4.04	7.916E−04	3.67
	7296	9.916E−04	3.08	2.057E−02	1.53	3.611E−05	3.99	4.664E−05	4.09
	29184	1.185E−04	3.06	7.146E−03	1.53	2.140E−06	4.08	2.774E−06	4.07
$\mathbb{P}_3$/Voronoi	300	1.158E−01	–	2.826E−01	–	6.431E−02	–	5.961E−02	–
	1200	2.263E−02	2.36	9.234E−02	1.61	4.017E−03	4.00	3.632E−03	4.04
	4800	2.157E−03	3.39	2.787E−02	1.73	2.583E−04	3.96	2.539E−04	3.84
	19200	2.393E−04	3.17	9.295E−03	1.58	1.649E−05	3.97	1.718E−05	3.89
Expected order			4		4		4		4
$\mathbb{P}_5$/Delaunay	456	6.098E−02	–	1.691E−01	–	3.034E−04	–	3.715E−04	–
	1824	9.660E−03	2.66	6.383E−02	1.41	6.796E−06	5.48	9.939E−06	5.22
	7296	1.359E−03	2.83	2.399E−02	1.41	1.207E−07	5.82	1.831E−07	5.76
	29184	1.704E−04	3.00	8.574E−03	1.48	1.767E−09	6.09	2.836E−09	6.01
$\mathbb{P}_5$/Voronoi	300	1.352E−01	–	2.610E−01	–	4.584E−03	–	4.955E−03	–
	1200	2.213E−02	2.61	9.116E−02	1.52	7.327E−05	5.97	8.740E−05	5.83
	4800	2.119E−03	3.38	2.914E−02	1.65	1.341E−06	5.77	1.573E−06	5.80
	19200	2.449E−04	3.11	1.005E−02	1.54	3.017E−08	5.47	3.703E−08	5.41
Expected order			6		6		6		6

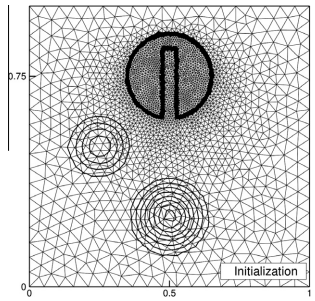


Fig. 5. Initial mesh and initial data for the SBR problem. The mesh is composed of 5190 triangles refined around the slotted disk. The resulting mesh is genuinely non-uniform.

end, we adapt the *u2* criterion to the density variable using local $\mathbb{P}_2$ polynomial reconstruction $\widetilde{\rho}_i = \widetilde{\rho}_i(.;2)$ to evaluate $\mathcal{X}_i = \partial_{xx}\widetilde{\rho}_i$ and $\mathcal{Y}_i = \partial_{yy}\widetilde{\rho}_i$. The *u2* detection algorithm for the Euler system is thus the following.

1. The PAD criterion is first checked on each cell K_i. If it is not satisfied then d_i is decremented and Steps 2 and 3 are skipped.
2. The DMP criterion of the density function is checked on each cell K_i
$$\min_{j\in\overline{\nu}(i)}\left(\rho_i^n, \rho_j^n\right) \leqslant \rho_i^\circ \leqslant \max_{j\in\overline{\nu}(i)}\left(\rho_i^n, \rho_j^n\right). \tag{14}$$
3. If ρ_i° does not satisfy (14) then
   a. Compute $\mathcal{X}_k, \mathcal{Y}_k$ for $k \in \underline{\nu}(i) \bigcup \{i\}$ and coefficient ε_k,
   b. Check criteria (9) and (10). If cell i is not a smooth extrema then d_i is decremented for any conservative variable, else U_i° is eligible.

The set of constraints $\mathcal{A}$ consists of the PAD, and the *u2* detection process on the density. Note that the density is thus the variable onto which the detection is performed. However there is a large number of possible choices of detection variables and decrementing procedures.

3.2.3. Positivity-preserving property

One important property a scheme must fulfill is to be positivity-preserving, that is given a set of physically admissible mean values the scheme provides another set of physically admissible ones. It is absolutely mandatory for the simulation to continue. In the case of the Euler equations density and pressure must be positive but this is not straightforwardly ensured by most of classical MUSCL or ENO/WENO schemes and most of simulation codes need a special treatment when the positivity is violated. Indeed designing a positivity-preserving scheme may be a difficult task and often leads to a more complex scheme because of the classical *a priori* limitation

Process is positivity-preserving for density and pressure. This point is further discussed in Section 3.2.3.

3.2.2. Extension of the *u2* detection process

Physical admissibility of the solution is not enough to prevent oscillations in the vicinity of discontinuities. It is a precondition but we require an supplementary detection criterion to decide whether the numerical solution is locally smooth or not. To this

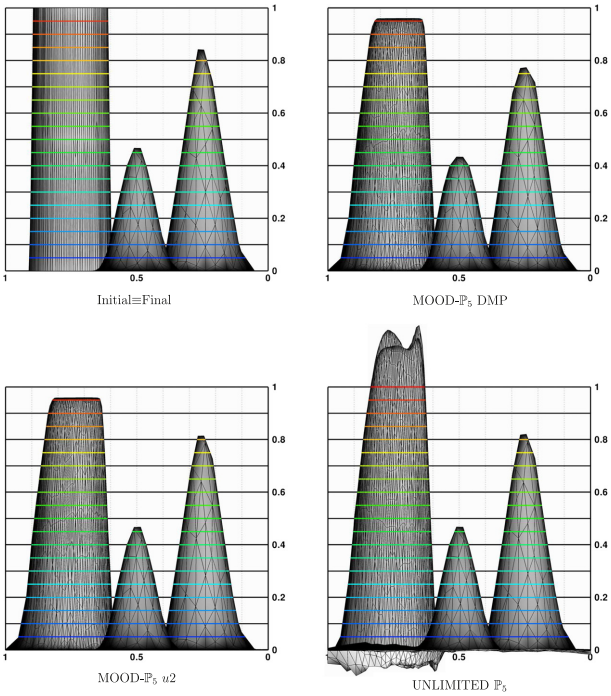


Fig. 6. Profiles of the SBR solution for the initial/final exact solution, MOOD-$\mathbb{P}_5$ with DMP detection process, MOOD-$\mathbb{P}_5$ with *u2* detection process, MOOD-$\mathbb{P}_5$ without any limitation.

philosophy. This classical difficulty is stated by the authors in [49] page 2754 as " It is very difficult to design a conservative high-order accurate scheme preserving the positivity". However the *a posteriori* treatment implies that the MOOD method is intrinsically positivity-preserving assuming the three following points:

1. The lowest order scheme is positivity-preserving, in our case it is the first-order finite volume one.
2. The positivity of density and pressure are parts of the set of constraints $\mathcal{A}$.
3. The EdgePD strategy is upper-limiting see [7] definition 9 page 4033. This implies that if the CellPD of a given cell is 0 then this cell is fully updated with the first-order scheme.

The proof that the MOOD method is positivity-preserving is analogous to the one in theorem 10 page 4033 of [7]. In short, given a candidate solution one checks the positivity of density and pressure. If a cell is problematic that is to say density or pressure is negative then the CellPD is decremented. The next candidate solution is computed and checked again: Either this next candidate is positive or the decrementing process carries on until the CellPD is zero. In this latter case points 1 and 3 necessarily imply the positivity of the candidate solution. As this process is the same for any cell it leads to a positivity-preserving solution in a finite number of MOOD iterations.

In the numerical section we propose the Noh test case for which our implementation of the classical MUSCL scheme generates a

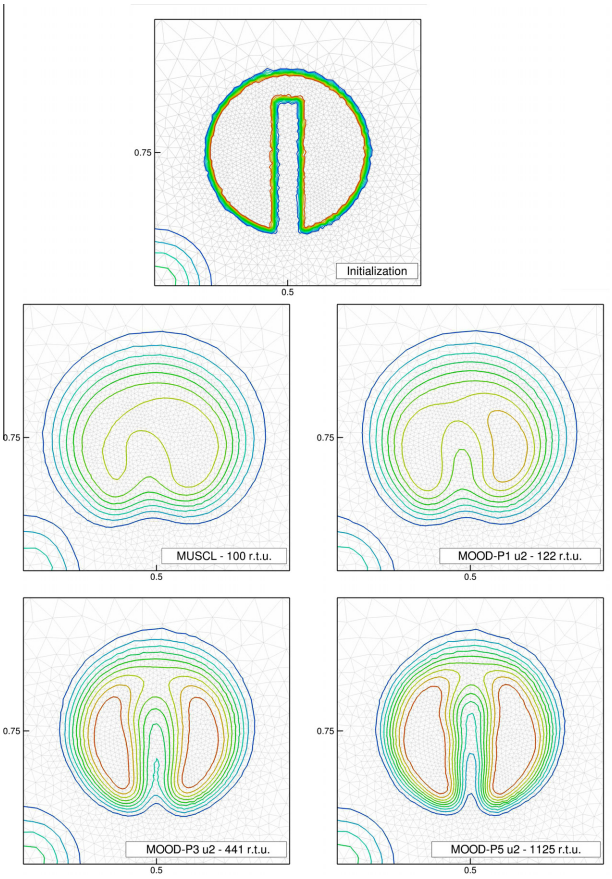


Fig. 7. Profiles of the SBR solution for the initial/final exact solution (top), for a limited MUSCL method (MLP) and MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ with *u*2 detection process.

Table 2
L^1 error for the SBR problem for different detection processes and polynomial degrees.

L^1 Error	DMP	*u*2	UNLIMITED
$\mathbb{P}_3$	3.219E−1	3.171E−1	3.734E−1
$\mathbb{P}_5$	2.690E−1	2.621E−1	3.223E−1

negative pressure and fails to complete the simulation whereas the MOOD method always gives a physical meaningful solution.

4. Numerical tests

MOOD has been implemented into a 2D unstructured (polygonal) code which can deal with advection equation and hydrodynamics equations. The polynomial reconstruction ranges from piecewise constant up to piecewise polynomial of fifth degree. Following Remark 4 one uses two decrementing sequences: $\mathbb{P}_5 - \mathbb{P}_2 - \mathbb{P}_0$ and $\mathbb{P}_3 - \mathbb{P}_2 - \mathbb{P}_0$. It implies that only two precomputed matrices for the reconstruction step per cell are only stored in memory for $d = d_{max}$ and $d = 2$. The flux computation involves integrals which are approximated using Gaussian numerical integration. We use two Gaussian points on edges for $\mathbb{P}_2$ and $\mathbb{P}_3$ reconstructions and three for $\mathbb{P}_5$ to reach the expected order of accuracy for numerical integrations. Time integration is performed with the RK3-TVD method given by system (6). We apply the MOOD procedure detailed in Section 2 to each sub-step of the RK3-TVD. The CellPD are thus reinitialized to d_{max} at the beginning of each time sub-step. By default we use classical time step control with CFL = 0.6. In the case of convergence study we use a fixed time step $\Delta t = \Delta x^{r/3}$ to reach *r*th-order of accuracy. Given a variable φ the relative L^1 and L^∞ errors are measured by:

$$err_1 = \frac{\sum_{i\in\mathcal{E}_d} |\varphi_i^N - \varphi_i^0||K_i|}{\sum_{i\in\mathcal{E}_d} |\varphi_i^0||K_i|} \quad \text{and} \quad err_\infty = \frac{\max_{i\in\mathcal{E}_d} |\varphi_i^N - \varphi_i^0|}{\max_{i\in\mathcal{E}_d} |\varphi_i^0|},$$

where $(\varphi_i^0)_i$ and $(\varphi_i^N)_i$ are respectively the cell mean values at initial time $t = 0$ and final time $t = t_{final} = N\Delta t$.

The unstructured meshes used in this paper are of different kinds, logically rectangular, Delaunay triangulation, Voronoi tessellation and non-conformal polygonal mesh. Contrarily to what was done in [7] the whole detection is made *a posteriori*, namely we do not check if the reconstructed values at Gauss points are physically admissible or not. If they are not, the flux and the cell mean values are usually undefined therefore the cell is flagged as problematic.

4.1. Advection equation

Let us consider the scalar linear advection of a quantity *u* with velocity $V(\mathbf{x})$

$$\begin{cases} \partial_t u + \nabla \cdot (Vu) = 0, \\ u(\cdot, t = 0) = u^0, \end{cases} \tag{15}$$

where $V(\mathbf{x})$ is a continuous function on $\Omega \in \mathbb{R}^2$ and u^0 is the initial condition. Boundary conditions are prescribed as periodic ones on $\partial\Omega$.

The Double Sine Translation (DST) is first tested on Delaunay triangulations and Voronoi tessellations in order to prove that on smooth solution MOOD can actually maintain very high-order of accuracy with the *u*2 detection criteria. Only second-order of accuracy is reached when DMP detection criterion is used. The second test is the Solid Body Rotation (SBR) that is used to prove that MOOD-*u*2 can preserve smooth extrema but can still limit discontinuous profiles. This problem is further used to show the improvement obtained when polynomial reconstruction degree is increased, in other word when high-order ($\mathbb{P}_1$) and very high-order ($\mathbb{P}_3$, $\mathbb{P}_5$) numerical schemes are used.

4.1.1. Double Sine Translation (DST)

Let Ω be the unit square. We consider a constant velocity $V = (2,1)$ and the C^∞ initial condition

$$u^0(x,y) = \sin(2\pi x)\sin(2\pi y).$$

The final time is $t_{final} = 2.0$. Periodic boundary conditions imply that the exact final solution coincides with the initial one. The solution is therefore always smooth during the computation.

The computations are carried out on series of successively refined Delaunay triangulations (from 456 up to 29,184 cells, see an example in Fig. 3 left panel) and polygonal Voronoi tessellations (from 300 up to 19,200 cells, see Fig. 3 right panel). Note that the meshes are far from being regular, see right panel of Fig. 3 fro instance. We plot in Fig. 4 the convergence curves obtained on the series of Delaunay triangulations and Voronoi tessellations. The MOOD method with the DMP detection process is displayed on top panels whereas the *u*2 detection process is on bottom panels. It clearly shows the strong limitation implied by the DMP since only 3rd-order and 2nd-order are reached in L^1 and L^∞ norms respectively independently of the polynomial degree. On the contrary the proposed *u*2 Detection Process reaches the expected order of convergence. This is actually explained by the fact that only polynomials of maximal degree are used during the whole computation, i.e. no CellPD decrementing is ever recorded.

L^1 and L^∞ errors and rates are given in Table 1 for the DMP and the *u*2 detection criteria. One observes that the optimal order of convergence is reached for the *u*2 detection criterion whereas only second-order accurate results are obtained when the DMP is used.

This accuracy test on smooth functions is passed by the MOOD method with *u*2 Detection Process, the next section is thus dedicated to the study of its behavior on non-smooth profiles.

4.1.2. Solid Body Rotation (SBR)

First introduced by Leveque in [25], the Solid Body Rotation test on the unit domain consists of one rotation of three shapes: a hump, a cone and a slotted cylinder. Each shape is located within a circle of radius $r^0 = 0.15$

Hump centered at $(x^0, y^0) = (0.25, 0.5)$

$$u^0(x,y) = \frac{1}{4}(1 + \cos(\pi \min(r(x,y), 1))).$$

Cone centered at $(x^0, y^0) = (0.5, 0.25)$

$$u^0(x,y) = 1 - r(x,y).$$

Table 3
Minimal and maximal mean values for the SBR problem for different detection processes and polynomial degrees.

Method	MUSCL	MOOD-$\mathbb{P}_1$		MOOD-$\mathbb{P}_3$		MOOD-$\mathbb{P}_5$	
Detec.		DMP	*u*2	DMP	*u*2	DMP	*u*2
Min	5.58E−10	0.00E+00	−2.45E−03	3.27E−08	−1.31E−03	1.10E−08	−5.60E−05
Max	7.48E−01	8.53E−01	8.51E−01	9.49E−01	9.54E−01	9.61E−01	9.60E−01

Table 4
L^1 and L^∞ errors and convergence rates for the isentropic vortex problem with MOOD and the physical admissible detection process. Comparison between conservative and primitive variables polynomial reconstructions for different polynomial degrees.

		Conservative variables				Primitive variables			
Deg.	Cell Nb	L^1 Error	–	L^∞ Error	–	L^1 Error	–	L^∞ Error	–
$\mathbb{P}_2$	200	1.850E−02	–	2.680E−01	–	2.002E−02	–	2.884E−01	–
	800	6.519E−03	1.50	1.255E−01	1.09	7.621E−03	1.39	1.771E−01	0.70
	3200	1.444E−03	2.17	2.208E−02	2.51	1.536E−03	2.31	4.054E−02	2.13
	12800	2.504E−04	2.53	3.631E−03	2.60	2.554E−04	2.59	6.060E−03	2.74
	51200	3.347E−05	2.90	4.923E−04	2.88	4.540E−05	2.49	8.756E−04	2.79
Expected order			3		3		3		3
$\mathbb{P}_3$	200	1.137E−02	–	1.880E−01	–	1.424E−02	–	2.384E−01	–
	800	2.504E−03	2.18	4.686E−02	2.00	3.530E−03	2.01	8.358E−02	1.51
	3200	3.524E−04	2.83	5.977E−03	2.97	5.666E−04	2.64	8.835E−03	3.24
	12800	1.947E−05	4.18	3.725E−04	4.00	1.377E−04	2.04	1.649E−03	2.42
	51200	1.069E−06	4.19	1.996E−05	4.22	3.460E−05	1.99	4.091E−04	2.01
Expected order			4		4		4		4
$\mathbb{P}_5$	200	8.193E−03	–	1.200E−01	–	1.161E−02	–	1.915E−01	–
	800	1.762E−03	2.22	3.433E−02	1.81	2.492E−03	2.22	3.740E−02	2.36
	3200	6.767E−05	4.70	1.133E−03	4.92	5.482E−04	2.18	6.112E−03	2.61
	12800	1.011E−06	6.06	2.237E−05	5.66	1.382E−04	1.99	1.598E−03	1.94
	51200	2.583E−08	5.29	4.809E−07	5.54	3.462E−05	2.00	4.039E−04	1.98
Expected order			6		6		6		6

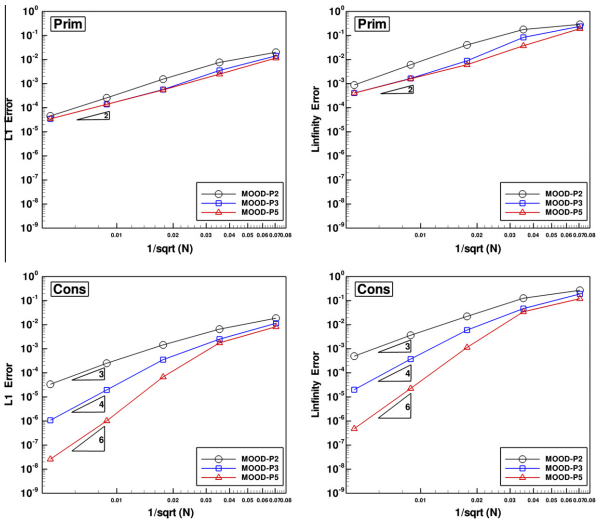


Fig. 8. Convergence curves for the isentropic vortex. Top figures correspond to the reconstruction with primitive variables while bottom figures use reconstruction with conservative variables. The left column represents the L^1-norm error and the right column the L^∞-norm error. The PAD detection process has been used.

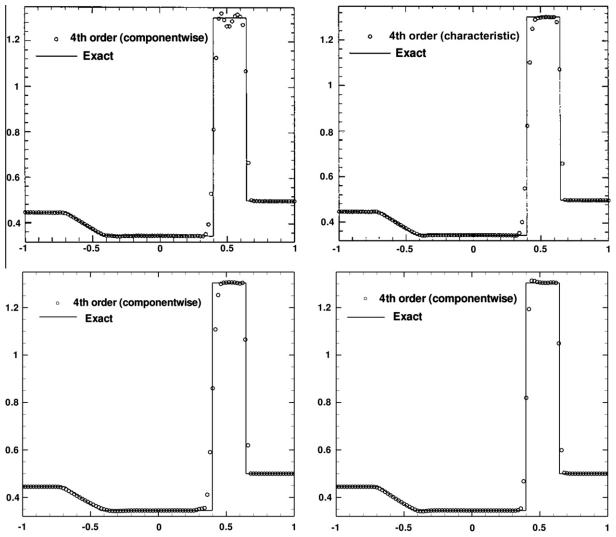


Fig. 9. Comparison between WENO and MOOD methods on 100 × 10 quadrangles split into triangles—Top: results of the 4th-order WENO method using Lax-Friedrichs flux from [18] on conservative variables (left) and on characteristic variables (right)—Bottom: results of the 4th-order MOOD-$\mathbb{P}_3$ method with *u*2 + PAD detection process using Lax-Friedrichs (left) and HLLC (right) fluxes.

Slotted cylinder centered at $(x^0, y^0) = (0.5, 0.75)$

$$u^0(x,y) = \begin{cases} 1 & \text{if } |x - 0.5| < 0.25, \quad \text{or } y > 0.85, \\ 0 & \text{elsewhere,} \end{cases}$$

where $r(x,y) = \frac{1}{2}\sqrt{(x-x^0)^2 + (y-y^0)^2}$. To perform the rotation, we use the velocity $V(\mathbf{x}) = (-y + 0.5,\ x - 0.5)$ and the final time $t_{final} = 2\pi$ corresponds to one full rotation.

For this test case we use a genuinely unstructured and non-uniform mesh made of 5190 triangles see Fig. 5 where we also display the initial data in isolines view, see also Fig. 6 top-left panel where a side view of the initial data is provided. This mesh is refined around the slotted disk, the ratio between the largest and smallest edge length is approximately seven. The three shapes while rotating move across the refined and coarse zones. The purpose is to emphasize the effects on the numerical results of using a truly non-regular mesh.

We plot in Fig. 6 profile views of the solution obtained from three methods but all with a $\mathbb{P}_5$ polynomial reconstruction. First the MOOD method with the DMP Detection Process, then the MOOD method with the *u*2 Detection Process, and finally the unlimited version of the FV scheme. These results show on one hand that the solution with *u*2 Detection Process on the non-smooth slotted cylinder is almost the same as for the DMP. On the other hand it shows that the *u*2 solution on the two smooth profiles are exactly the ones obtained by the unlimited scheme. In other words, the *u*2 Detection Process maintains the same accuracy as an unlimited scheme on smooth profiles and almost the monotonicity of a limited scheme on non-smooth ones. The same conclusion applies for any other polynomial degrees tested hence we have skipped these figures. In Fig. 7 are displayed a zoom on the slotted disk at the final time for the initial/final, the limited MUSCL scheme (MLP [31]), MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ with *u*2 detection process.

In Table 2 are gathered the errors for $\mathbb{P}_3$ and $\mathbb{P}_5$ in order to show that the *u*2 Detection Process provides a slightly better accuracy than the DMP detection process. Finally we display in Table 3 the min/max values of the final numerical solution for the limited MUSCL method (MLP), MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ all with DMP detection or *u*2 Detection Process. This table shows that the *u*2 detection process permits slight undershoots which is one of the reasons MOOD can reach high-order accuracy.

4.2. Euler system

In this section we test the MOOD method on unstructured meshes for hydrodynamics problems governed by the Euler sys-

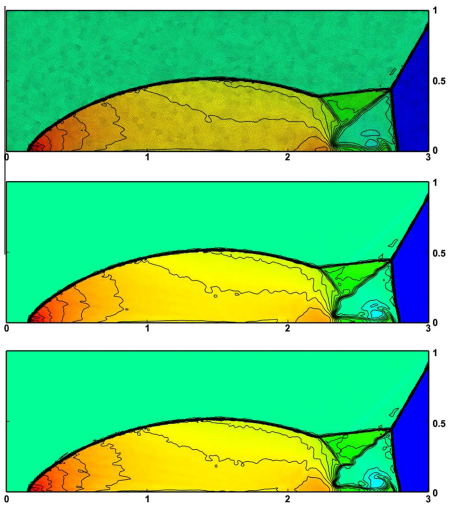


Fig. 10. Comparison between the $\mathbb{P}_2$ (top), $\mathbb{P}_3$ (middle) and $\mathbb{P}_5$ (bottom) polynomial reconstructions with the conservative variables using the same mesh. Physical Admissible Detection (PAD) and $u2$ Detector have been both used to prevent numerical oscillations.

tem. First we need to assess the effective numerical accuracy of the method on a smooth problem for which an exact solution exists. We choose an isentropic vortex which presents a smooth profile during the entire simulation and, as such, permits the estimation of errors and convergence orders. In a second test we run the Lax shock tube to assess the essentially non-oscillatory behavior of MOOD compared to classical WENO results. Then we run the Double Mach reflection problem to highlight the good capacity of the MOOD method to capture strong shocks and contact discontinuities. Moreover we provide CPU cost and memory storage tables. Next the Noh problem is used to assess the positivity-preserving property of the MOOD method. Last we propose a genuine physical problem extracted from [37] for which experimental results are available.

4.2.1. Isentropic vortex

The isentropic vortex problem is detailed in [35,47], therefore we only mention the basic data for the sake of consistency. The simulation domain Ω is the square $[-5,5]\times[-5,5]$ and we consider an initial gas flow given by the following condition (ambient gas) $\rho_\infty = 1.0$, $u_\infty = 1.0$, $v_\infty = 1.0$, $p_\infty = 1.0$, with a normalized ambient temperature $T^*_\infty = 1.0$ computed with the perfect gas equation of state and $\gamma = 1.4$.

A vortex centered at $\mathbf{x}_{\text{vortex}} = (x_{\text{vortex}}, y_{\text{vortex}}) = (0,0)$ is added to the ambient gas at the initial time $t = 0$ with the following conditions $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, and $T^* = T^*_\infty + \delta T^*$

$$\delta u = -y'\frac{\beta}{2\pi}\exp\left(\frac{1-r^2}{2}\right), \quad \delta v = x'\frac{\beta}{2\pi}\exp\left(\frac{1-r^2}{2}\right),$$
$$\delta T^* = -\frac{(\gamma-1)\beta}{8\gamma\pi^2}\exp(1-r^2).$$

with $r = \sqrt{x'^2+y'^2}$, $(x' = x - x_{\text{vortex}}, y' = y - y_{\text{vortex}})$ and vortex strength is given by $\beta = 5.0$. Consequently, the initial density is given by

$$\rho = \rho_\infty\left(\frac{T^*}{T^*_\infty}\right)^{\frac{1}{\gamma-1}} = \left(1 - \frac{(\gamma-1)\beta}{8\gamma\pi^2}\exp(1-r^2)\right)^{\frac{1}{\gamma-1}} \quad (16)$$

We assume periodic condition on the boundary and the exact solution at any time t is the same vortex but translated.

The goal of the present test is to highlight the stagnation of the rate of accuracy when primitive variables are used for the polynomial reconstructions instead of conservative ones. As pointed out in the previous section, nonlinear operations on means values reduces the method order up to at most a second-order one. We have performed the numerical simulations of the isentropic vortex problem with the same mesh using the less restrictive Physical Admissible Detection (PAD) procedure to provide effective very high-order. A series of refined meshes (from 200 up to 51200 cells) are successively used to compute the numerical solution.

In Table 4 are gathered the L^1 and L^∞ errors and rates of convergence for MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ using the Physical Admissible Detection Clearly, conservative variable

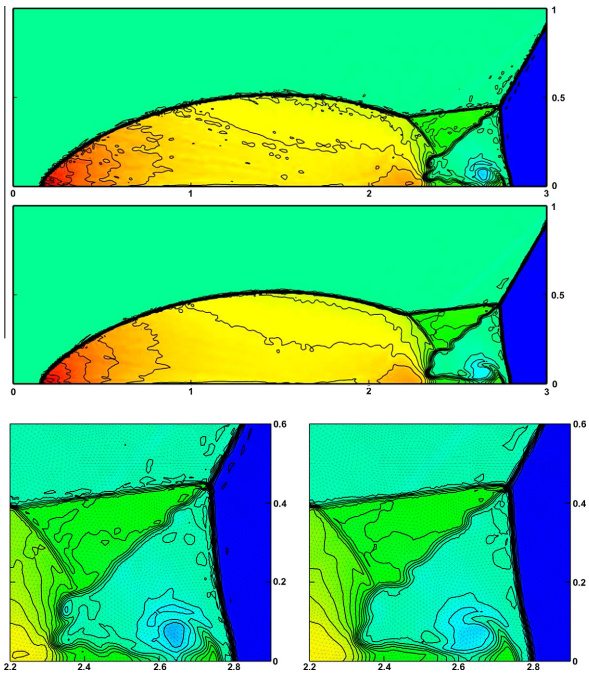


Fig. 11. Results of the MOOD method with $\mathbb{P}5$. In the top figure, simulation has been carried out with the Physical Admissible Detection (PAD) Detection Process while we have both employed the PAD and $u2$ Detection in the middle figure. The left bottom and right bottom figure give a zoom of the solution with the PAD and $u2$ + PAD Detection Process respectively.

Table 5
Memory storage and total number of iterations for the double mach problem according to the different configurations with the MOOD method.

Mesh	$\mathbb{P}_2$-3rd-order		$\mathbb{P}_3$-4th-order		$\mathbb{P}_5$-6th-order		Detection
	Memory	Iterations	Memory	Iterations	Memory	Iterations	
57,600 qua.	240 Mo	1012	385 Mo	998	840 Mo	1004	$u2$ + PAD
	180 Mo	1016	270Mo	1010	572 Mo	1031	PAD
17,624 tri.	60 Mo	1264	105 Mo	1265	250 Mo	1265	$u2$ + PAD
	50 Mo	1268	67Mo	1272	165 Mo	1275	PAD

reconstructions provide the optimal convergence rate whereas the reconstruction with primitive variables is systematically reduced to a second-order one. We also display in Fig. 8 the convergence curves corresponding to the errors of Table 4. Finally we also

Table 6
Total time and cost for one complete time step of a single cell for the double mach problem.

Machine	Mesh	$\mathbb{P}_2$-3rd-order		$\mathbb{P}_3$-4th-order		$\mathbb{P}_5$-6th-order		Detection
		Total (s)	Per iter. (μs)	Total (s)	Per iter. (μs)	Total (s)	Per iter. (μs)	
M1	57,600 qua.	2157	37	3162	55	8964	155	$u2$ + PAD
		1346	23	2327	40	8314	140	PAD
	17,624 tri.	601	27	1003	45	1650	74	$u2$ + PAD
		492	22	762	34	1573	70	PAD
M2	57,600 qua.	2228	38	4785	83	12,371	214	$u2$ + PAD
		1629	28	3830	66	11,629	196	PAD
	17,624 tri.	615	27	922	41	1292	58	$u2$ + PAD
		521	23	707	32	1079	48	PAD
M3	57,600 qua.	683	12	1089	19	3696	66	$u2$ + PAD
		490	8	859	15	3604	61	PAD
	17,624 tri.	265	12	397	18	594	27	$u2$ + PAD
		230	10	308	14	492	22	PAD

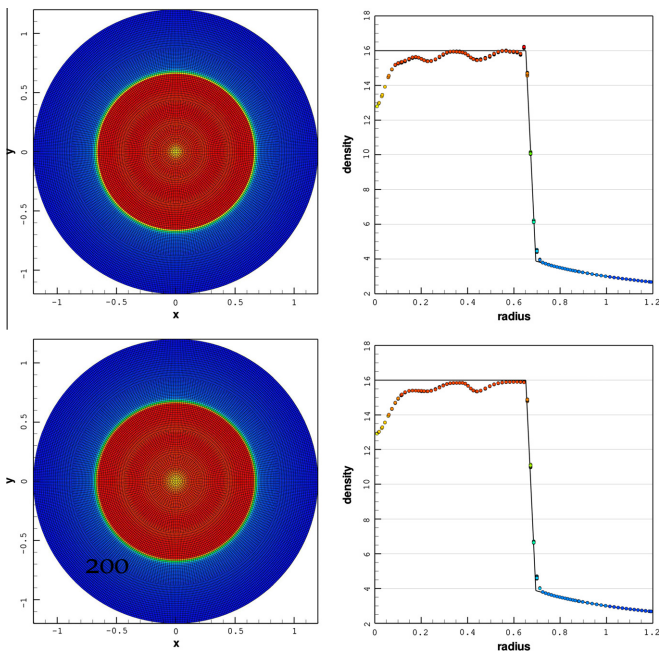


Fig. 12. Noh problem at $t_{\text{final}} = 2.0$ on a polygonal grid—Left: Density map and mesh— Right: Cell density as a function of cell radius r_i vs exact solution—Top panels correspond to the PAD detection process—Bottom panels correspond to the $u2$ + PAD detection process.

mention that when the vortex is not in motion, *i.e.* $(u_\infty, v_\infty) = (0,0)$, then the reconstruction using primitive variables does produce the correct order of convergence.

4.2.2. Lax shock tube

The 1D Lax shock tube consists of two states $(\rho_L, u_L, p_L) = (0.445, 0.698, 3.528)$ and $(\rho_R, u_R, p_R) = (0.5, 0, 0.571)$ separated by the interface $x = 0$. In order to compare with the finite volume multi-dimensional WENO results of [18], we run the problem on the domain $\Omega = [-1;1]\times[0,0.2]$ until final time $t = 0.26$ using a mesh made of 100×10 quadrangles split into two triangles through the same diagonal for all cells (see Fig. 5.5 of [18]).

The goal of this test is to compare the essentially non-oscillatory behavior of the MOOD method using the $u2$ detection process with the classical genuinely multi-dimensional finite volume WENO results based on conservative or characteristic variables.

On the top panels of Fig. 9 we reproduce the density profiles from [18] obtained with the 4th-order WENO method based on conservative (left) and characteristic (right) variables. We recall that this method uses combinations of $\mathbb{P}^2$ (3rd-order) polynomials to reach 4th-order. The bottom panel presents the MOOD-$\mathbb{P}_3$ density profiles on conservative variables with the $u2$ + PAD detection process where only one line of triangles is displayed to comply with Fig. 5.6 of [18]. On the left we plot the result obtained with the Lax-Friedrichs numerical flux (which is used for the WENO results) and on the right the result using the HLLC flux. We observe that the non-oscillatory behavior of the MOOD method with $u2$ + PAD detection process is equivalent to the WENO on characteristic variables while it is clearly better than the WENO on conservative variables. Moreover we see that the use of HLLC gives better a result for a negligible additional cost, with only three points in the contact discontinuity instead of five and remains essentially non-oscillatory.

4.2.3. Double Mach reflection of a strong shock

The double mach reflection of a strong shock was first proposed in [46]. This test problem involves a Mach 10 shock in a perfect gas with $\gamma = 1.4$, which is initially positioned at $x = 1/6$, $y = 0$ and makes a 60° angle with the x-axis. The gas ahead of the shock is at rest and has uniform initial density $\rho^0 = 1.4$ and pressure $p^0 = 1$. The reflecting wall lies along the bottom of the domain, beginning at $x = 1/6$. The region from $x = 0$ to $x = 1/6$ along the bottom boundary at $y = 0$ is always assigned values for the initial post-shock flow. Inflow boundary condition on the left side and outflow condition on the right side are also set. At the top, the boundary conditions are set to describe the exact motion of the Mach 10 flow (see [11]).

The goal of the test is, on one hand, to quantitatively show the effect of the polynomial degree reconstruction when dealing with strong shock and, on the other hand, to observe the capacity of the method to reproduce the complex structure due to the contact discontinuities in the right part of the shock.

The mesh has been obtained using the free mesher Gmsh by a refinement of a coarser Delaunay ones, it is constituted of 102,720 triangles (see Fig. 10 top). Moreover for all Figs. 30 isolines between 1.39 and 23 have been drawn.

We depict in Fig. 10 the impact of the polynomial degree of the reconstruction on the numerical solution using the same mesh. The $u2$ + PAD Detection Processes has been employed to control the oscillations in the vicinity of the shock. Clearly the degree of the reconstruction has a strong impact on the solution accuracy and improve the shock capture. Most relevant parts are the contact discontinuities in the right zone $x \in [2.3, 2.7]$ which show the capacity of the scheme to reduce numerical viscosity when employing higher-order reconstructions.

Fig. 11 is a comparison between the Physical Admissible Detection (PAD) and the coupling $u2$ + PAD. The $u2$ Detection Process reduces the oscillations but increases the numerical viscosity close to contact discontinuities. It is worth noting that even with a weak Detection Process, namely the PAD procedure, the MOOD method is still very robust and provides a solution resembling the classical ones from the literature [46]. The choice of the detecting procedure depends of the simulation goal: Less oscillations with the $u2$ + PAD or less diffusive with the PAD alone.

To conclude with this test case, we provide in Tables 5 and 6 the cost of the MOOD method running on a single core of the three following machines (using -O3 flag for gfortran compiler).

- *M1*: A laptop with Intel Core2Duo P7550 (2 cores) @ 2.26 GHz, 3 MB of L2 Cache, 8 GB of RAM.
- *M2*: A server with two Intel Xeon E5335 (4 cores) @ 2.00Ghz, 8 MB of L2 Cache, 16 GB of RAM.
- *M3*: A desktop with Intel Core i5 2500 (4 cores) @ 3.30 GHz, 6 MB of L2 Cache, 8 GB of RAM.

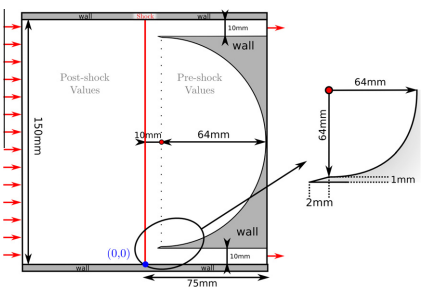


Fig. 13. Domain characteristics for the shock impacting a cylindrical cavity. Red arrows represent inflow and outflow boundary conditions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

This comparison is done on two different meshes, one made of 57,600 uniform quadrilaterals and one Delaunay triangulation with 17,624 cells. We compare MOOD-$\mathbb{P}2$, MOOD-$\mathbb{P}3$ and MOOD-$\mathbb{P}5$ for both the PAD and $u2+$PAD detection processes. We give in Table 5 the memory cost (in left column) and the total number of iterations (in right column) for all simulations, while we provide in Table 6 the total CPU time (in left column) and the time in micro-seconds needed for one complete time step of a single cell (in right column) including reconstruction, flux computation and time integration (RK3) of all variables.

It is fairly difficult to compare the cost of two methods running on different machines, for instance the method is faster on triangles with $M2$ compared to $M1$ but it is the opposite for quadrilaterals. However according to reference [15] the MOOD method is competitive when compared to truly unstructured methods of the same order.

4.2.4. Noh problem as a positivity-preserving test case

The goal of the Noh problem in Cartesian geometry is to numerically prove that the MOOD method is positivity-preserving, see Section 3.2.3 for a discussion on this point. It is a difficult problem well-known in the Lagrangian community, see as instance [27,26]. It is noticeable that our implementation of the classical MUSCL scheme is not able to simulate this problem without creating negative pressures.

The problem is run in the disk of radius 1.2 centered at $(0,0)$. We initialize a perfect gas with $\gamma = 5/3$, density $\rho_0 = 1$, pressure $p_0 = 10^{-10}$ and velocity $U_0(x,y) = (-x/\sqrt{x^2+y^2}, -y/\sqrt{x^2+y^2})$ such that $\|U_0(x,y)\| = 1$. A cylindrical shock wave generated at the origin further diverges until final time $t_{\text{final}} = 2.0$. The exact solution at t_{final} is thus given by

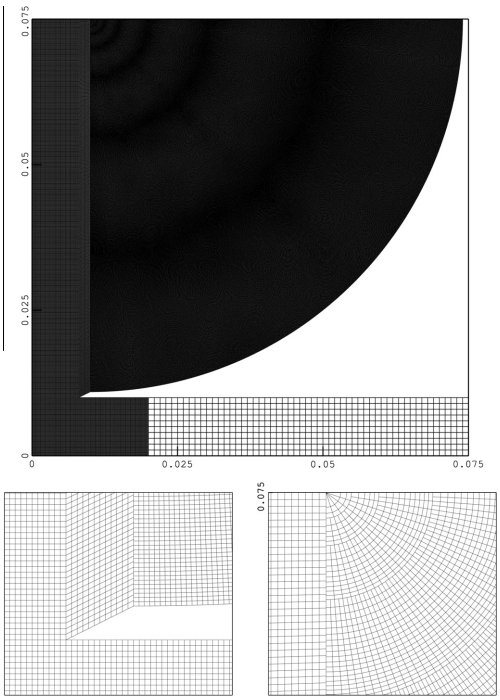


Fig. 14. On top, we display the global view of the mesh where the different mesh zones are clearly visible. On bottom, zooms on the non-convex part of the mesh (on left) and on the junction between the polar part of the mesh and the quasi-uniform one (right). Non-conformity are clearly visible.

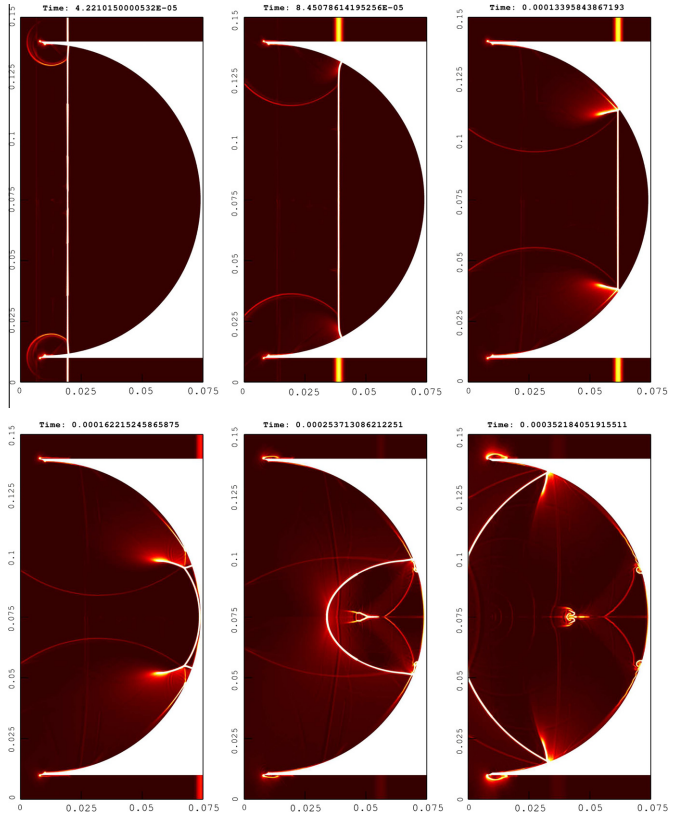


Fig. 15. Gradient density magnitude is shown at different times. Time 0 corresponds to the initial shock at position $x = 0$.

$$\{\rho, p, u_r\} = \begin{cases} \{16, \frac{16}{3}, 0\} & \text{if } r < r_s, \\ \left\{(1+\frac{2}{\Gamma}), 10^{-10}, -1\right\} & \text{if } r > r_s, \end{cases} \tag{17}$$

where r is the radius, u_r the radial velocity and $r_s = 2/3$ the shock wave position. This problem is simulated on a polygonal mesh made of 19,756 cells with about 100 cells in the radial direction. Notice that the mesh is made of seven layers of quadrangles separated with degenerated polygons, see Fig. 12. We display the MOOD-$\mathbb{P}3$ results for the density maps (left panels) and the density as a function of cell radius (right panels) in Fig. 12. The top panels correspond to the PAD detection process whereas the bottom ones correspond to the $u2+$PAD process. One observes that the symmetry is almost perfectly reproduces. Notice that the PAD detection process is only intended to ensure the physical meaningfulness of the solution but does not prevent oscillations to occur. Independently of the order of the scheme the PAD always provides a meaningful solution. As a consequence the $u2+$PAD not only provides a valid solution without negative pressure but also removes the oscillation after the shock wave.

4.2.5. Impact of a shock on a cylindrical cavity

We finally test the ability of MOOD method to capture physics in realistic conditions by simulating the experiment proposed in [37] where a planar shock impacts a cylindrical cavity. We consider the case of a nominal incident shock Mach number of 1.33 in ambient air (with $\gamma = 1.4$) at 0.95 bar pressure. Moreover we use the domain configuration A (following notation of [37]) we detail in Fig. 13.

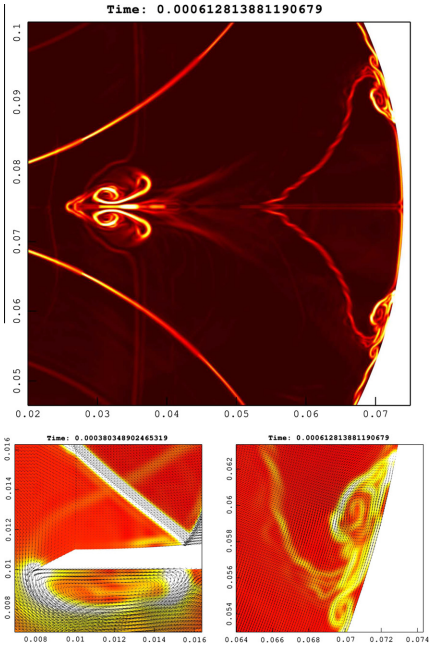


Fig. 16. Zooms on different parts of the solution. On top, gradient density magnitude is shown at a late time when instabilities are well developed. On bottom, vortices at the entry of the cavity (left) and the instabilities (right) along the wall are displayed with density gradient magnitude in color and velocity vectors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The variables initialization is split in two parts, the pre-shock values

$$(\rho, u, v, p) = (1.1175, 0.0, 0.0, 95000.0),$$

and the post-shock ones

$$(\rho, u, v, p) = (1.7522, 166.3435, 0.0, 180219.75),$$

leading to conditions of [37] at temperature $T = 296.15$ K.

The simulation is only preformed in the lower half part of the domain for symmetry argument, namely from $y = 0$ mm to $y = 75$ mm. The 193,615 cells mesh is composed of triangles, quadrangles but also more general polygons with non-conformal elements (see Fig. 14) to better suit with the complex geometry of the set-up. Notice that non-conformity is simply handled using polygons, i.e. no special treatment is used. We also deliberately use a heterogeneous mesh to highlight that the MOOD method is not much affected by the quality of the mesh.

The simulation are carried out with the MOOD-P3 method (fourth-order) using the PAD and the $u2$ Detection Process. Pictures are rendered as a full mesh by symmetry even if the computation was done on a half-domain to easier compare with physical results of [37]. Fig. 15 represents the density gradient magnitude at six different times to embrace the global behavior of the solution. In details, top right is chosen to be compared to Fig. 7(a) of [37], bottom center to Fig. 8(d) and bottom right to Fig. 9(c) of same paper. Our results are clearly in agreement with physical results. In Fig. 16 different zooms on solution at several times are plotted. On the top part, density gradient magnitude at a late time is given and is to be compared to Fig. 14(b) in [37] while we superpose, in the bottom figure, the velocity vectors on the density magnitude gradient to show the created vortices at the entrance of the cavity (left) and highlight the instabilities lying along the wall.

5. Conclusion and perspectives

The paper presents important new extensions of the MOOD method for unsteady advection and hydrodynamics equations, that ensure high-order approximations (up to the sixth-order) on unstructured meshes.

We introduced new efficient detection processes and proved that the MOOD method is intrinsically positivity-preserving for the hydrodynamics system of equations assuming that the first-order scheme is. This has been numerically assessed on the Noh problem for which our implementation of the MUSCL scheme fails due to negative pressures.

Then both for the advection equation and the hydrodynamics Euler system, we proposed numerical tests to confirm that the MOOD method provides very high-order of accuracy on unstructured meshes for smooth solutions (e.g. isentropic vortex in motion) and non-oscillatory behavior on discontinuous ones (e.g. Lax shock tube). Moreover the memory storage and CPU time have also been reported for the double Mach problem, proving that the MOOD method is competitive. The last numerical test showed that the MOOD method, on a relatively coarse and non-conformal polygonal mesh, is able to simulate complex physics from an experimental set-up of the impact of a shock wave on a cylindrical cavity.

Finally we plan to improve the detection procedure, especially for vectorial problems to achieve a very low diffusion but still preventing the oscillations from appearing. Application to full three-dimensional problem is also an attractive task since performing an efficient computational solution is always a challenging problem. The extension of the MOOD method to deal with steady-state solution needs also more investigations. Overall an important perspective is the polynomial reconstruction itself. We have observed that the main computational cost comes from the reconstruction stage and that the reconstruction quality strongly depends on the stencil employed. Such a point is of crucial importance from a computational point of view to obtain tractable complex numerical simulations.

Appendix A. The discrete maximum principle on mean values provides at most a second-order scheme

We recall that a time explicit scheme preserves the Discrete Maximum Principle (DMP) if for all cell K_i

$$\min_{j\in\mathcal{V}(i)}\left(U_i^n, U_j^n\right) \leqslant U_i^{n+1} \leqslant \max_{j\in\mathcal{V}(i)}\left(U_i^n, U_j^n\right). \tag{18}$$

It has been shown in [34,31,22] that any scheme based on the DMP property reduces the accuracy to second-order for regular functions due to inaccurate approximation at extrema. Indeed following [50], let us consider the advection problem in $\mathbb{R}$ to avoid boundary condition issues

$$\begin{cases} \partial_t U + \partial_x U = 0, \\ U(x, t=0) = \cos(x). \end{cases} \tag{19}$$

We consider a uniform discretization $x_i = ih,\ i \in \mathbb{Z}$ and $h > 0$ being the cell size and initialize the mean value on cell $K_0 = [0, h]$ as

$$U_0^{t=0} = \frac{1}{h}\int_0^h \cos(x)dx = \frac{\sin(h)}{h}. \tag{20}$$

Now, let us perform one time step with $\Delta t = h/2$ of a finite volume scheme which respects the DMP property. The exact solution at time $t = h/2$ is $U^{ex}(x, h/2) = \cos(x - h/2)$ and accordingly the exact mean value on K_0 is

$$U_0^{ex,t} = \frac{1}{h}\int_0^h \cos(x - h/2)dx = \frac{2\sin(h/2)}{h}. \tag{21}$$

However a Taylor expansion provides

$$U_0^{ex,t} = \frac{2\sin(h/2)}{h} = 1 - \frac{h^2}{24} + O(h^4).$$

But the initial mean values are bounded by

$$U_0^{t=0} = \frac{\sin(h)}{h} = 1 - \frac{h^2}{6} + O(h^4).$$

Clearly, the exact mean value $U_0^{ex,t}$ on cell K_0 is greater than the maximum mean values over all cells at time $t = 0$ with an error of $h^2/8$ as

$$\left|U_0^{ex,t} - U_0^{t=0}\right| \leqslant \left|\frac{h^2}{24} - \frac{h^2}{6} + O(h^4)\right| = \frac{h^2}{8} + O(h^4).$$

Therefore a scheme which fulfills the DMP property necessarily provides a solution lower than $\sin(h)/h$, hence after the first cycle the numerical solution verifies $U_0^1 \leqslant U_0^{t=0} = 1 - \frac{h^2}{6} + O(h^4)$. It follows that the approximation of the mean value has an error of order $O(h^2)$ compared to the exact mean value on cell K_0. Consequently the scheme is at most second-order accurate and DMP-type of criteria cannot be used strictly for higher than second-order schemes and has to be relaxed.

References

[1] Abgrall R. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. J Comput Phys 1994;114:45–58.
[2] Abgrall R. Essentially non-oscillatory residual distribution schemes for hyperbolic problems. J Comput Phys 2006;214:773–808.
[3] Barth TJ. Numerical methods for conservation laws on structured and unstructured meshes. VKI March 2003 lectures series.
[4] Barth TJ, Fredrickson PO. Higher-order solution of the euler equations on unstructured grids using quadratic reconstruction. In: AIAA conference paper 90-0013; 1990.

[5] Buffard T, Clain S. Monoslope and multislope MUSCL methods for unstructured meshes. J Comput Phys 2010;229:3745–76.
[6] Clain S, Clauzon V. L^∞ stability of the MUSCL methods. Numer Math 2010;116:31–64.
[7] Clain S, Diot S, Loubère R. A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD). J Comput Phys 2011;230(10):4028–50.
[8] Cockburn B, Shu CW. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: general framework. Math Comput 1989;52:411–35.
[9] Cockburn B, Lin SY, Shu CW. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. J Comput Phys 1989;84:90–113.
[10] Cockburn B, Hou S, Shu CW. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. Math Comput 1990;54:545–81.
[11] Cockburn B, Shu C-W. The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J Comput Phys 1998;141:199–224.
[12] Csik A, Ricchiuto M, Deconinck H. A conservative formulation of the multidimensional upwind residual distribution schemes for general nonlinear conservation laws. J Comput Phys 2002;179:286–312.
[13] Dumbser M, Castro M, Parés C, Toro EF. ADER schemes on unstructured meshes for nonconservative hyperbolic systems: applications to geophysical flows. Comput Fluids 2009;38:1731–48.
[14] Dumbser M, Kser M. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. J Comput Phys 2007;221:693–723.
[15] Dumbser M, Kser M, Titarev VA, Toro EF. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. J Comput Phys 2007;226:204–43.
[16] Harris R, Wang ZJ, Liu Y. Efficient quadrature-free high-order spectral volume method on unstructured grids: theory and 2D implementation. J Comput Phys 2008;227:1620–42.
[17] Harten A, Engquist B, Osher S, Chakravarthy S. Uniformly highorder accurate nonoscillatory schemes III. J Comput Phys 1987;71:279–309.
[18] Hu C, Shu CW. Weighted essentially non-oscillatory schemes on triangular meshes. J Comput Phys 1999;150:97–127.
[19] Hubbard ME. Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids. J Comput Phys 1999;155(1):54–74.
[20] Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126:202–28.
[21] Jiang G-S, Tadmor E. Non-oscillatory central schemes for multidimensional hyperbolic conservative laws. SIAM J Sci Comput 1998;19:1892–917.
[22] Kolgan VP. Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics. Trans Central Aerohydrodynam Inst 1972;3:68–77 [in Russian].
[23] Kolgan VP. Finite-difference schemes for computation of three dimensional solutions of gas dynamics and calculation of a flow over a body under an angle of attack. Trans Central Aerohydrodynam Inst 1975;6:1–6 [in Russian].
[24] Kolgan VP. Application of the principle of minimizing the derivative to the construction of finite-difference schemes for computing discontinuous solutions of gas dynamics. J Comput Phys 2010;230:2384–90.
[25] Leveque Randall J. High-resolution conservative algorithms for advection in incompressible flow. SIAM J Numer Anal 1996;33:627–65.
[26] Loubère R, Maire P-H, Vachal P. Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme. Commun Comput Phys 2011;10(4):940–78.
[27] Maire Pierre-Henri. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. J Comput Phys 2009;228:2391–425.
[28] Friedrich O. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. J Comput Phys 1998;144:194–212.
[29] Ollivier-Gooch CF. Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. J Comput Phys 1997;133:6–17.
[30] Osher S, Chakravarthy S. High resolution schemes and the entropy condition. SIAM J Numer Anal 1984;21:955–84.
[31] Park JS, Yoon S-H, Kim C. Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids. J Comput Phys 2010;229:788–812.
[32] Ricchiuto M, Bollermann A. Stabilized residual distribution for shallow water simulations. J Comput Phys 2009;228:1071–115.
[33] Sander R. A third-order accurate variation non-expansive difference scheme for single nonlinear conservation law. Math Comput 1988;51:535–58.
[34] Shi J, Hu C, Shu CW. A technique of treating negative weights in WENO schemes. J Comput Phys 2002;175:108–27.
[35] Shu C-W. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: Cockburn B, Johnson C, Shu C-W, Tadmor E, Quarteroni A, editors. Advanced numerical approximation of nonlinear hyperbolic equations. Lecture notes in mathematics, vol. 1697. Springer; 1998. p. 325–432.
[36] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing scheme. J Comput Phys 1988;77:439–71.
[37] Skews BW, Kleine H. Flow features resulting from shock wave impact on a cylindrical cavity. J Fluid Mech 2007;580:481–93.
[38] Titarev VA, Toro EF. ADER schemes for three-dimensional non-linear hyperbolic systems. J Comput Phys 2005;204:715–36.
[39] Toro EF, Hidalgo A. ADER finite volume schemes for nonlinear reaction–diffusion equations. Appl Numer Math 2009;59:73–100.
[40] Tsoutsanis P, Titarev VA, Drikakis D. WENO schemes on arbitrary mixed-element unstructured meshes in three space dimensions. J Comput Phys 2011;230:1585–601.
[41] Van Leer B. Towards the ultimate conservative difference scheme I. The quest of monotonicity. In: Proceedings of the third international conference on numerical methods in fluid mechanics, Lecture notes in physics, vol. 18; 1973. p. 163–8.
[42] Van Leer B. Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme. J Comput Phys 1974;14:361–70.
[43] Wang ZJ. Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation. J Comput Phys 2002;178:210–51.
[44] Wang ZJ, Liu Y. Spectral (finite) volume method for conservation laws on unstructured grids: extention to two dimensional scalar equation. J Comput Phys 2002;179:665–97.
[45] Wolf WR, Azevedo JLF. High-order ENO and WENO schemes for unstructured grids. Int J Numer Methods Fluids 2007;55:917–43.
[46] Woodward P, Colella P. The numerical simulation of two-dimensional fluid flow with strong shocks. J Comput Phys 1984;54:115–73.
[47] Yee HC, Vinokur M, Djomehri MJ. Entropy splitting and numerical dissipation. J Comput Phys 2000;162:33–81.
[48] Zhang Y-T, Shu C-W. Third-order WENO scheme on three dimensional tetrahedral meshes. Commun Comput Phys 2009;5:836–48.
[49] Zhang X, Shu C-W. Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. Proc Roy Soc A 2011;467:2752–76.

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2012; **00**:1–34
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld

The MOOD method in the three-dimensional case: Very-High-Order Finite Volume Method for Hyperbolic Systems.

S. Diot*[,1], R. Loubère[1], S. Clain[1,2]

[1]*Institut de Mathématiques de Toulouse, Université de Toulouse, France.*
[2]*Departamento de Matemática e Aplicações, Campus de Gualtar - 4710-057 Braga, Campus de Azurém - 4800-058 Guimarães, Portugal.*

SUMMARY

The Multi-dimensional Optimal Order Detection (MOOD) method for two-dimensional geometries has been introduced in "A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD)", J. Comput. Phys. 230 (2011), and enhanced in "Improved Detection Criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials", Comput. & Fluids 64 (2012). We present in this paper the extension to 3D mixed meshes composed of tetrahedra, hexahedra, pyramids and prisms. In addition, we simplify the $a2$ detection process previously developed and show on a relevant set of numerical tests for both the convection equation and the Euler system that the optimal high-order of accuracy is reached on smooth solutions while spurious oscillations near singularities are prevented. At last, the intrinsic positivity-preserving property of the MOOD method is confirmed in 3D and we provide simple optimizations to reduce the computational cost such that the MOOD method is very competitive compared to existing high-order Finite Volume methods.
Copyright © 2012 John Wiley & Sons, Ltd.

Received …

KEY WORDS: Finite Volume; high-order; conservation law; polynomial reconstruction; 3D; unstructured; Euler; MOOD; positivity-preserving

CONTENTS

*Correspondence to: steven.diot@math.univ-toulouse.fr

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls *[Version: 2010/05/13 v2.00]*

1. INTRODUCTION

First introduced in [5], the Multi-dimensional Optimal Order Detection (MOOD) method proposes a new strategy to provide third-order approximations to hyperbolic scalar or vectorial problems for two-dimensional geometry with structured meshes. The author then gave an extension in [11] to general unstructured 2D meshes where they achieved a sixth-order convergence in space introducing new detection-limitation procedure. The issue we address in the present paper is to extend the MOOD method to three-dimensional geometries with general polyhedral unstructured meshes for the scalar advection equation and the hydrodynamics Euler system. The method casts in the generic framework of the finite volume method but fundamentally differs from the traditional techniques by the specific detection-limitation procedure implemented by the authors. Indeed, classical high-order polynomial reconstruction-based schemes such as MUSCL [21, 35, 36, 22, 18, 3, 23] or ENO/WENO methods [15, 16, 29, 39, 27] rely on an *a priori* limiting procedure to achieve some stability properties. For instance, in MUSCL-like methods unlimited slopes are reduced through the use of limiters to respect some Discrete Maximum Principle or Total Variation Diminishing properties. In the same way, ENO/WENO-like methods employ an essentially non-oscillatory polynomial which provides an accurate solution while preventing undesirable oscillations from appearing.

We state that such limitation strategies are *a priori* in the sense that only the data at time t^n are used to first perform the limitation procedure and then compute an approximation at time t^{n+1}. Generally, the "worst case scenario" (speculative approach) has to be considered as plausible and, consequently a "precautionary principle" is applied. It results that most of the time the limitation mechanism unnecessarily operates and may reduce the scheme accuracy due to restrictive assessments. The MOOD principle lies in a different approach since we first compute a candidate solution for time t^{n+1} and use this *a posteriori* information to check if the proposed approximation is valid. Roughly speaking, we compute a candidate solution without any limitation using local polynomial reconstructions to provide accurate approximation of the flux (the degree is set to a prescribed maximal value). We then detect if this solution locally fails to fulfill some stability criteria (detection of problematic cells) and further decrement polynomial degree only on problematic regions (limitation step) before recomputing a new candidate solution. An iterative procedure (the MOOD algorithm) is carried out by successively decrementing the degree to provide the *optimal* local polynomial reconstruction for each cell to satisfy the given stability criteria. At the end of the MOOD algorithm, the candidate solution is eligible and turns out to be the approximation at time t^{n+1}. The *a posteriori* strategy brings new benefits. We dramatically reduce the number of polynomial reconstructions regarded to the ENO/WENO method since our technique only requires one polynomial function for each cell. Most of the time, the polynomial with maximal degree is employed since the limitation mechanism is only activated for problematic cells (objective approach). From a physical point of view, the positivity preserving property (for the Euler equations as instance) is simply guaranteed by the *a posteriori* strategy applying a simple detection procedure which checks the physical admissibility of the solution.

The paper is organized as follows. In section 2, we detail the concept of the MOOD method, while the detection criteria are developed in section 3 both for the advection equation and the hydrodynamics Euler system of equations. Numerical tests are proposed in section 4 to prove the efficiency of the method: we first consider the scalar advection equation and show effective high-order of accuracy for regular solutions with the fourth- and sixth-order schemes considering meshes made of hexahedra and pyramids. We then propose an H-shaped discontinuous profile in rotation to verify the non-oscillatory property of the MOOD method. Finally, numerical simulations are carried out for the Euler system to test the method with a nonlinear vectorial problem. As preliminary experiments, the classical 1D test cases, namely the Sod and Lax shock tubes and the Shu-Osher and Woodward-Collela problems, are run on 3D tetrahedral and pyramidal meshes. Then the numerical order of accuracy is checked on the 2D isentropic vortex in motion extended by

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

invariance for a 3D mesh and a realistic 2D test case (introduced in [11]), namely the *impact of a shock wave on a cylindrical cavity*, is carried out on a mesh made of triangular and quadrangular prisms. At last, we present results for the 3D explosion problem on a pyramidal mesh and the interaction of a shock wave with a quarter of cone on a mesh of 1.1 millions of tetrahedra. We moreover provide computational cost (CPU and memory storage) for the 3D explosion problem for the MOOD method for different polynomial degrees. We conclude with section 5 and delineate some future perspectives.

2. THE MOOD CONCEPT

We consider the generic hyperbolic equation defined on a domain $\Omega \subset \mathbb{R}^3$, $t > 0$ cast in the conservative form

$$\begin{aligned} \partial_t U + \nabla \cdot F(U) &= 0, && \text{(1a)} \\ U(\cdot,0) &= U_0, && \text{(1b)} \end{aligned}$$

where $U = U(\mathbf{x}, t)$ is the vector of unknown functions depending on $\mathbf{x} = (x, y, z) \in \Omega$ and on the time t. We denote by F the so-called physical flux where we shall consider the autonomous case $F = F(U)$ (Euler system as instance) and the non-autonomous situation $F = F(\mathbf{x}, U)$ such as $\nabla_{\mathbf{x}}.F(\mathbf{x}, .) = 0$ (scalar convection case). Function u_0 stands for the initial condition while the boundary conditions will be prescribed in section devoted to the numerical simulations.

2.1. Framework

In order to design the numerical scheme, we introduce the following notation illustrated in Figure 1. The computational domain Ω is assumed to be a polyhedron bounded set of $\mathbb{R}^3$ divided into polyhedral cells K_i, $i \in \mathcal{E}_{el}$ where $\mathcal{E}_{el}$ is the cell index set. For each cell K_i, we denote by $\mathbf{c}_i$ the cell centroid, and define the set $\underline{\nu}(i)$ of all the indexes $j \in \mathcal{E}_{el}$ such that elements K_j share a common face f_{ij} with K_i and the set $\overline{\nu}(i)$ of all the indexes $j \in \mathcal{E}_{el}$ such that $K_i \cap K_j \neq \emptyset$ (see illustrations in Figure 2). Moreover for each face $f_{ij} = K_i \cap K_j$, $\mathbf{n}_{ij}$ stands for the unit normal vector going from K_i to K_j and we denote by $(\xi_{ij,r}, q_{ij,r})$, $r = 1, ..., R_{ij}$ the quadrature rule for the numerical integration on f_{ij} where $\xi_{ij,r}$ is the weight associated to the r^{th} quadrature point $q_{ij,r}$ with $\sum_{r=1}^{R} \xi_{ij,r} = 1$, $\forall i \in \mathcal{E}_{el}$ and $\forall j \in \underline{\nu}(i)$ (see Figure 1).

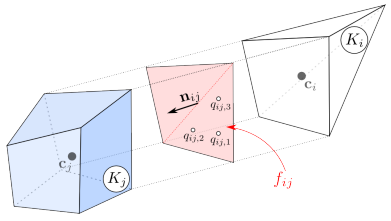


Figure 1. Notation for a three-dimensional mesh: exploded view of the face f_{ij} between two cells K_i and K_j. Centroids are respectively denoted by $\mathbf{c}_i$ and $\mathbf{c}_j$. Three quadrature points $q_{ij,r}$, $r = 1, 2, 3$, on f_{ij} are drawn for illustration. The unit normal vector pointing from K_i to K_j is denoted $\mathbf{n}_{ij}$.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

To avoid a specific treatment of the boundary faces we introduce the notion of virtual cell. To this end, assuming that cell K_i has a face $f_{ie} = K_i \cap \partial\Omega$ on $\partial\Omega$, we introduce the virtual cell K_j where $j \notin \mathcal{E}_{el}$ obtained by symmetrical transformation of the original cell K_i which represents the exterior side of Ω. We shall denote by $\mathcal{E}_{bd}$ the index set of all virtual cells such that $\widetilde{\mathcal{E}_{el}} = \mathcal{E}_{el} \cup \mathcal{E}_{bd}$ is the index set of all cells (including the virtual ones).

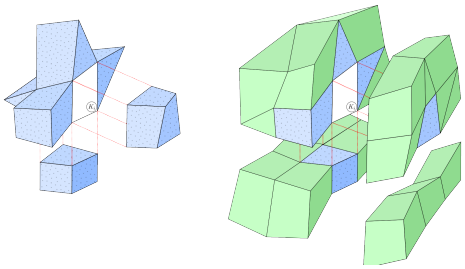


Figure 2. Illustrations for index sets $\underline{\nu}(i)$ (left) and $\overline{\nu}(i)$ (right) in 3D.

The generic first-order Finite Volume scheme associated to equation (1) writes

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \mathbb{F}(U_i^n, U_j^n, \mathbf{n}_{ij}), \tag{2}$$

where U_i^n is an approximation of the mean value of U at time t^n on K_i, $\mathbb{F}(U_i^n, U_j^n, \mathbf{n}_{ij})$ is a numerical flux which satisfies the properties of consistency and monotonicity for the scalar case, Δt stands for the time step while $|f_{ij}|$ and $|K_i|$ are the area of face f_{ij} and the volume of cell K_i respectively.

To provide high-order finite volume schemes, we use convex combinations of the initial building-block (2) with better approximations at the quadrature points to compute the numerical flux (see [5, 11] for instance). The high-order schemes are thus obtained from an original first-order Finite Volume and that is of crucial importance from a computational and implementation point of view (re-use of the original first-order code to achieve high-order approximations). We substitute the first-order approximations of the flux integral by higher-order versions, the scheme then writes

$$U_i^{n+1} = U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \sum_{r=1}^{R_{ij}} \xi_{ij,r} \mathbb{F}(U_{ij,r}^n, U_{ji,r}^n, \mathbf{n}_{ij}), \tag{3}$$

where $U_{ij,r}^n, U_{ji,r}^n$ are high-order approximations of U at quadrature points $q_{ij,r}$ on both side of f_{ij}.

For meshes constituted of tetrahedral cells, all faces are triangles. Consequently R_{ij} and $\xi_{ij,r}$ are independent of i and j and the previous scheme rewrites as a convex combination of the first-order scheme (2)

$$U_i^{n+1} = \sum_{r=1}^{R} \xi_r \left(U_i^n - \Delta t \sum_{j \in \underline{\nu}(i)} \frac{|f_{ij}|}{|K_i|} \mathbb{F}(U_{ij,r}^n, U_{ji,r}^n, \mathbf{n}_{ij}) \right). \tag{4}$$

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

Remark 2.1. *When dealing with general polyhedral cells, the quadrature rules may be different from a face to another and such a convex combination is not valid anymore. However, since each polygonal face can be split into triangles, one can recover equation (4) by considering each polyhedron as a polyhedron only constituted by triangular faces (and consequently with more faces than the original one).*

Let denote by $U_h^n = \sum_{i\in\mathcal{E}_{el}} U_i^n \mathbf{1}_{K_i}$ the constant piecewise representation of approximation $(U_i^n)_{i\in\mathcal{E}_{el}}$, we introduce operator $\mathcal{H}^R(U_h^n)$ such that relation (3) rewrites as

$$U_h^{n+1} = U_h^n + \Delta t \mathcal{H}^R(U_h^n). \tag{5}$$

From the original forward Euler discretization in time (5) we derive a high-order approximation in time using a Runge–Kutta 3 TVD method:

$$U_h^{n+1} = \frac{U_h^n + 2U_h^{(3)}}{3} \quad \text{with} \quad \left\{ \begin{array}{rcl} U_h^{(1)} & = & U_h^n + \Delta t\ \mathcal{H}^R(U_h^n) \\ U_h^{(2)} & = & U_h^{(1)} + \Delta t\ \mathcal{H}^R(U_h^{(1)}) \\ U_h^{(3)} & = & \widehat{U}_h^{(2)} + \Delta t\ \mathcal{H}^R(\widehat{U}_h^{(2)}) \end{array} \right. \tag{6}$$

where $\widehat{U}_h^{(2)}$ is the convex combination $(3U_h^n + U_h^{(2)})/4$.

The time discretization introduces a 3^{rd}-order error which makes the whole scheme to be formally 3^{rd}-order accurate. However setting $\Delta t = \Delta \mathbf{x}^{r/3}$ where r is the spatial order of accuracy and $\Delta\mathbf{x}$ is a characteristic length provide same order for spatial and time errors.

2.2. Reconstruction

We have formally defined an arbitrary high-order accurate Finite Volume scheme, providing that $U_{ij,r}$ is a high-order accurate point-wise approximation of $U(q_{ij,r})$ computed from cell K_i. In this subsection, we briefly describe the technique to produce such approximations and we refer to [5, 11] and references herein for details. Let us consider a scalar variable u and denote by $\widetilde{u}_i(\cdot, \mathsf{d})$ a local polynomial approximation of degree d reconstructed on cell K_i from the mean values of function u on a set of neighboring cells $\mathcal{S}_i^{\mathsf{d}}$ called stencil. For the sake of conservation, *i.e.* $\frac{1}{|K_i|}\int_{K_i} \widetilde{u}_i(\mathbf{x};\mathsf{d})\ d\mathbf{x} = u_i$, we assume that the polynomial has the following structure

$$\widetilde{u}_i(\mathbf{x};\mathsf{d}) = u_i + \sum_{1\leq|\boldsymbol{\alpha}|\leq\mathsf{d}} \mathcal{R}_i^{\boldsymbol{\alpha}} \left((\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}} - \frac{1}{|K|}\int_K (\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}}\ d\mathbf{x} \right), \tag{7}$$

where the polynomial coefficients $\mathcal{R}_i^{\boldsymbol{\alpha}}$ are fixed by solving a least-squares problem equivalent to minimizing the functional

$$E = \sum_{j\in\mathcal{S}_i^{\mathsf{d}}} \left(\frac{1}{|K_j|}\int_{K_j} \widetilde{u}_i(\mathbf{x};\mathsf{d})\ d\mathbf{x} - u_j \right)^2.$$

In practice, the polynomial coefficients are obtained by multiplying the pseudoinverse of the least-square problem matrix (that we store in memory) with the vector of mean values on the stencil, see [11] for details. We moreover recall that for the vectorial case, the reconstructions are performed for all the conservative components independently.

Finally, considering that polynomial reconstructions $\widetilde{u}_i(\mathbf{x};\mathsf{d})$ are provided for all cells $K_i, i \in \mathcal{E}_{el}$, we compute the approximation at each quadrature point of each face f_{ij} by $u_{ij,r} = \widetilde{u}_i(q_{ij,r};\mathsf{d})$. The so-called ($\mathsf{d}$-)unlimited scheme (3) is thus defined by employing the reconstructed values in the numerical flux without any restriction (*i.e.* no limitation).

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

Remark 2.2. *We recall that the reconstruction process is very time and memory consuming and would like to emphasize that contrarily to WENO methods we consider only one reconstruction stencil per cell and per degree, so that a lot of computational resources are saved.*

2.3. The MOOD concept

It is well-known that the first-order scheme (2) is robust but tremendously diffusive, while unlimited schemes of higher-order produce spurious oscillations in the vicinity of steep gradients. Limitation mechanisms have been developed to prevent the oscillations from appearing, such as the slope or flux limitation in MUSCL methods [21, 35, 36, 22, 18, 3, 23] or the computation of an Essentially Non-Oscillatory polynomial reconstruction in WENO methods [15, 16, 29, 39, 27]. As we mention in the introduction, all the classical techniques act *a priori* in the sense that we determine the limitation process in function of the current data (*i.e.* the solution at time t^n). As a consequence, the *a priori* strategy imposes very drastic accuracy reduction due to strong and unnecessary limitations (the worst case scenario has to be considered). Moreover, computational resources are allocated to perform the limitation process where most of the time it is useless. In the ENO/WENO case for instance, several polynomial reconstructions are required even if the solution is locally regular and can be approximated with only one polynomial function.

In two recent papers [5, 11], we have introduced a new approach based on an *a posteriori* evaluation of the solution to determine if the limitation procedure has to be applied and where. The technique is *a posteriori* in the sense that we compute a candidate solution (a potential approximation at time t^{n+1}) and we use the data of the candidate solution to determine if the solution is valid. More precisely, the detection-limitation mechanism operates in several steps. A candidate solution is first computed with the highest-order unlimited scheme (the polynomials with maximal degree). Then a detection procedure is performed to determine the problematic cells, *i.e.* all cells where the approximation does not respect some given criteria (see next section). For problematic cells, the solution is recomputed with a lower-order unlimited scheme (using polynomials with lower degree) and we repeat the procedure detection-degree decrementing (the MOOD algorithm) till the cell satisfy the detection criteria or the polynomial degree is zero. In the last case, a robust first-order scheme (2) is triggered and a meaningful solution is thus provided. Note that we need to guarantee that the MOOD algorithm stops after a finite number of iterations.

We now set some fundamental notions to define the MOOD method. We name Cell Polynomial Degree, shortened as CellPD and denoted by d_i, the degree of the polynomial reconstruction on cell K_i. We name Face Polynomial Degrees, shortened as FacePD and denoted by d_{ij} and d_{ji}, the degrees of the polynomial reconstructions actually used to compute approximations, $U_{ij,r}$ and $U_{ji,r}$, of the solution on face f_{ij} at quadrature points $q_{ij,r}$ respectively from K_i and K_j. The computation of d_{ij} and d_{ji}, named FacePD strategy, consists in evaluating the FacePD $\mathsf{d}_{ij}, \mathsf{d}_{ji}$ that we employ on both sides of the interface f_{ij} with respect to the CellPD of the neighboring cells. In previous studies (see [5] for details), we have proposed and experimented several strategies and introduced the *upper-limiting* property for a FacePD strategy which states that for any degree $\bar{\mathsf{d}}$, the following property holds

$$\mathsf{d}_i = \bar{\mathsf{d}} \Longrightarrow \mathsf{d}_{ij} \leq \bar{\mathsf{d}} \quad \text{and} \quad \mathsf{d}_{ji} \leq \bar{\mathsf{d}}, \quad \forall j \in \underline{\nu}(i).$$

This guarantees (see [5]) that the MOOD algorithm stops after a finite number of iterations. In practice, we use the simple rule $\mathsf{d}_{ij} = \mathsf{d}_{ji} = \min(\mathsf{d}_i, \mathsf{d}_j)$.

As mentioned above, the detection mechanism is performed on the candidate solution $U_h^\star$ and criteria have to be set to specify what is a good solution. To this end, we denote by $\mathcal{A}$ the set of detection criteria (*e.g.* positivity of a variable or a maximum principle) that the numerical approximation has to respect on each cell and we say that a candidate solution is $\mathcal{A}$-eligible if it fulfills all the criteria of $\mathcal{A}$. If the candidate solution is not $\mathcal{A}$-eligible on cell K_i, then we decrement the polynomial degree. However the solution may not be $\mathcal{A}$-eligible regardless of the set $\mathcal{A}$ even if the polynomial degree is zero for the cell. Consequently, we shall consider the solution *acceptable*

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

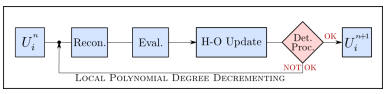


Figure 3. Flowchart of the MOOD algorithm: Recon. stands for polynomial reconstruction, Eval. for high-order evaluations at quadrature points, H-O Update for high-order update of the solution and Det. Proc. for detection process.

on the cell if either it is $\mathcal{A}$-eligible, or is a first-order solution (*i.e.* CellPD has been decremented to zero). To sum up, the MOOD algorithm for the explicit discretization in time consists in the following stages depicted in Figure 3:

0. Initialize $\mathsf{d}_i = \mathsf{d}_{max}, \quad \forall i \in \mathcal{E}_{el}$.
1. Compute polynomial reconstruction of degree $\mathsf{d}_i, \quad \forall i \in \mathcal{E}_{el}$.
2. Compute FacePD d_{ij} and d_{ji} and evaluate high-order approximations at quadrature points on face $f_{ij}, \quad \forall j \in \underline{\nu}(i), \ \forall i \in \mathcal{E}_{el}$.
3. Compute candidate solution mean values through unlimited scheme (3), $\forall i \in \mathcal{E}_{el}$.
4. Detection process: decrement CellPD of cells where solution is not acceptable.
5. Stop if the solution is acceptable else go to stage 1.

Following [5, 11] we extend the MOOD algorithm initially designed for a one-time step scheme to the RK3-TVD scheme by applying it to each sub-step of the RK3-TVD (6) procedure. The MOOD method is now completely defined except from the detection criteria that have to be suited to the problem we intend to solve. Such a difficult task requires the complete next section.

3. DETECTION CRITERIA

The crucial point of the MOOD method is the elaboration of the detection criteria set $\mathcal{A}$ which characterizes the properties we want the numerical solution to fulfill. A fundamental purpose of the detection criteria is to obtain higher-order of accuracy for regular solutions while preventing numerical oscillations in the vicinity of discontinuous profiles. This would consequently provide an efficient and robust method. We face several difficulties to design such a set since accuracy and robustness are antagonist objectives. Moreover, in the Euler problem, a physically admissible solution is mandatory since the positivity of the density and the pressure is required to compute the numerical flux. It results that the detection criteria would cover a wide spectrum of properties and restrictions. A key point we shall detail in the following is the notion of "numerical regularity" in the sense that we have to determine if, for a local stencil and a set of data (for instance the mean values), we can associate a regular or a irregular function. This point is really important since the choice of the reconstruction (namely the polynomial degree) depends on it.

The present section intends to extend and improve detection criteria initially introduced in [11] to evaluate the local "numerical regularity" of the approximation. We first begin the study for the advection equation in section 3.1 and address the hydrodynamics Euler system in section 3.2.

3.1. Advection equation

The scalar advection problem is characterized by the physical flux $F(U) = VU$ where $V \in \mathbb{R}^3$ stands for the velocity that we assume to be a regular function on Ω and satisfies $\nabla_{\mathbf{x}} V(\mathbf{x}) = 0$ while $U = U(t, \mathbf{x}) \in \mathbb{R}$ is the passive scalar quantity transported by the fluid.

When dealing with a constant velocity, the exact solution is simply given by $U(\mathbf{x}, t) = U_0(\mathbf{x} - Vt)$ and clearly fulfills a maximum principle, *e.g.* the minimum of the solution can not be lower

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

than the initial condition minimum (and a similar property for the maximum). Consequently, it seems natural to impose such a condition at the numerical level and, as proposed in [5], we integrate in the set $\mathcal{A}$ the Discrete Maximum Principle (DMP) on mean values for the candidate solution $U_h^\star$ formulated like this:

$$\min_{j\in\overline{\nu}(i)} (U_i^n, U_j^n) \leq U_i^\star \leq \max_{j\in\overline{\nu}(i)} (U_i^n, U_j^n). \tag{8}$$

A solution is $\mathcal{A}$-eligible if condition (8) is satisfied for all the cells and we have proved in [5] that the scheme equipped with such set $\mathcal{A}$ provides a numerical solution which, under first-order scheme CFL condition, satisfies the DMP. However a strict application of relation (8) at smooth extrema unavoidably reduces the scheme accuracy to two. It suggests that relation (8) is too restrictive and should be relaxed.

In [11] we have relaxed the condition on cells which violate the DMP. More specifically, the relation (8) has been supplemented with a new criteria, the so-called $u2$ detection criteria which provides an effective arbitrary high-order of accuracy. As mention in the beginning of the section, the key point is to determine if the numerical solution is regular enough to be approximated by a high-order polynomial reconstruction and avoid the Gibbs phenomena. To this end, let assume that the candidate solution does not satisfy the DMP criteria on cell K_i. A first step consists in reconstructing quadratic polynomials on K_i denoted by $\widetilde{U}_i$ and on its neighbors K_j for $j \in \overline{\nu}(i)$ denoted by $\widetilde{U}_j$. In a second step, we define approximations to the local minimal and maximal curvatures, namely

$$\mathcal{X}_i^{min} = \min_{j\in\overline{\nu}(i)} \left(\partial_{xx}\widetilde{U}_i,\ \partial_{xx}\widetilde{U}_j \right) \text{ and } \mathcal{X}_i^{max} = \max_{j\in\overline{\nu}(i)} \left(\partial_{xx}\widetilde{U}_i,\ \partial_{xx}\widetilde{U}_j \right), \tag{9}$$

$$\mathcal{Y}_i^{min} = \min_{j\in\overline{\nu}(i)} \left(\partial_{yy}\widetilde{U}_i,\ \partial_{yy}\widetilde{U}_j \right) \text{ and } \mathcal{Y}_i^{max} = \max_{j\in\overline{\nu}(i)} \left(\partial_{yy}\widetilde{U}_i,\ \partial_{yy}\widetilde{U}_j \right), \tag{10}$$

$$\mathcal{Z}_i^{min} = \min_{j\in\overline{\nu}(i)} \left(\partial_{zz}\widetilde{U}_j,\ \partial_{zz}\widetilde{U}_j \right) \text{ and } \mathcal{Z}_i^{max} = \max_{j\in\overline{\nu}(i)} \left(\partial_{zz}\widetilde{U}_i,\ \partial_{zz}\widetilde{U}_j \right), \tag{11}$$

where we emphasize that the second derivatives are constant and naturally referred to as *curvatures*. The $u2$ detection criterion holds in the following definition.

Definition 3.1 ($u2$ detection criterion)**.** *A candidate solution $U_i^\star$ in cell K_i which violates the DMP is nonetheless eligible if the following holds*

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > 0 \quad \textit{and} \quad \left| \frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \right| \geq 1 - \varepsilon,$$

$$\textit{and} \quad \mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > 0 \quad \textit{and} \quad \left| \frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}} \right| \geq 1 - \varepsilon,$$

$$\textit{and} \quad \mathcal{Z}_i^{max}\mathcal{Z}_i^{min} > 0 \quad \textit{and} \quad \left| \frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}} \right| \geq 1 - \varepsilon,$$

where ε is a smoothness parameter.

The definition derives from the idea that the comparison of local second derivatives of the quadratic reconstructions on a neighborhood provides a relevant information on the numerical smoothness of the underlying solution. More precisely, we consider that the underlying solution (characterized by the piecewise constant mean value) is (ε-)smooth if for each direction the curvatures have the same sign (no oscillation or inflection point) and are (ε-)*close* enough to each-other. Such a definition lies in a fitting of the parameter ε the value of which defines the threshold

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

between what is considered as smooth extrema or as discontinuity. Therefore the determination of ε is of crucial importance since it rules the decrementing process activation.

From a practical point of view, the $u2$ detection criteria operates in two stages. First the test on the sign of curvatures (left inequalities) is performed. If oscillations are detected, *i.e.* the product is negative, the cell is considered as problematic and the decrementing procedure must be applied. The second stage is performed only if the product is positive. It consists in computing the ratios between minimal and maximal curvatures and comparing it to $1-\varepsilon$ (right inequalities). If the curvatures ratio does not respect the inequality, the cell is considered as problematic and the decrementing process must be applied.

In [11], we propose a parameter ε depending on a local characteristic length and on the spatial dimension of the domain. This was a first attempt to the determination of ε and deeper investigations have shown that a simpler definition provides same quality results. To set the ε value, we extend the parameter as a new function $\varepsilon_x = \varepsilon_x\left(\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right)$ (for the x-direction) with respect to the curvatures which have to satisfy the restriction

$$\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \geq 1-\varepsilon_x. \tag{12}$$

The goal is to determine a relevant function ε_x which enables high-order approximation and robustness. We first note that the curvatures ratio ranges between zero and one so that ε_x must range in $[0,1]$ to make sense. Moreover, the ratio is expected to be close to zero on discontinuities and close to one on smooth functions which are the two extreme cases. For a non-smooth function, we expect that the limiting procedure operates and since the closer to zero ε is, the less smooth the underlying function is considered, ε_x is expected to be close to zero on discontinuities, *i.e.* $\lim_{r\to 0^+}\varepsilon_x(r) = 0^+$. On the other hand, a ratio close to one indicates smooth functions, so we expect $\lim_{r\to 1^-}\varepsilon_x(r) = 1^-$ to relax the restriction. We thus propose to define ε_x as a continuous increasing function of the curvatures ratio such that $\varepsilon_x(0)=0$ and $\varepsilon_x(1)=1$. After several attempts, it appears that the simple function $\varepsilon_x(r)=r$ is an excellent choice. When substituting expression of $\varepsilon_x = \mathcal{X}_i^{min}/\mathcal{X}_i^{max}$ in relation (12), the x-direction curvatures criterion becomes

$$\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \geq 1-\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}},$$

and yields

$$\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} \geq 1/2.$$

Finally we apply the same reasoning for y- and z-directions and obtain

$$\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}} \geq 1/2 \quad \text{and} \quad \frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}} \geq 1/2.$$

The linearity of function ε_x simplifies the final inequalities and leads to the constant value $\varepsilon = 1/2$ in definition 3.1.

Remark 3.2. *The definition of ε is really simpler than the one proposed in [11]. However numerous numerical test cases have been carried out and no change in the quality of results have been reported.*

Remark 3.3. *Numerical experiments show that the choice of the neighborhood where the curvatures are computed should define a convex hull which contains the reference cell K_i. To constitute such a stencil, we used the index set of cells $\underline{v}(i)$ in 2D (see [11]) but this choice is not relevant for three-dimensional meshes and we use the index set $\overline{v}(i)$ in equations (9)-(11) to provide the expected results even for large form factor meshes.*

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

To conclude the section we propose in Figure 4 an algorithmic view of the complete detection process [DMP$\to u2$] for the advection equation constituted of the DMP of equation (8) relaxed by the $u2$ detection criteria of definition 3.1. We emphasize that the algorithm is given in the case of a cell K_i with $U_i^\star$ its associated candidate solution mean value.

If $\left[U_i^\star \leq \min_{j\in\overline{\mathcal{V}}(i)}(U_i^n, U_j^n)\right]$ **or** $\left[U_i^\star \geq \max_{j\in\overline{\mathcal{V}}(i)}(U_i^n, U_j^n)\right]$ **then**
- decrement the CellPD d_i

else
→ compute the curvatures $\mathcal{X}_i^{min}, \mathcal{X}_i^{max}, \mathcal{Y}_i^{min}, \mathcal{Y}_i^{max}, \mathcal{Z}_i^{min}, \mathcal{Z}_i^{max}$
If $\left[\mathcal{X}_i^{min}\mathcal{X}_i^{max} < 0\right]$ **or** $\left[\mathcal{Y}_i^{min}\mathcal{Y}_i^{max} < 0\right]$ **or** $\left[\mathcal{Z}_i^{min}\mathcal{Z}_i^{max} < 0\right]$ **then**
- decrement the CellPD d_i

else
→ compute the curvatures ratios $\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}, \frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}, \frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}$
If $\left[\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} < 1-\varepsilon\right]$ **or** $\left[\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}} < 1-\varepsilon\right]$ **or** $\left[\frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}} < 1-\varepsilon\right]$ **then**
- decrement the CellPD d_i

end if
end if
end if

Figure 4. Algorithmic view of the [DMP$\to u2$] detection process for the advection equation.

3.2. Hydrodynamics Euler system

The Euler system for three-dimensional geometries writes

$$\partial_t \begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{pmatrix} + \partial_x \begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ u(E+p)\end{pmatrix} + \partial_y \begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ v(E+p)\end{pmatrix} + \partial_z \begin{pmatrix}\rho v\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ w(E+p)\end{pmatrix} = 0, \tag{13}$$

where ρ stands for the density, u, v and w for the velocity components in the x, y and z directions respectively, p for the pressure and E for the total energy. This system is closed by the Equation Of State (EOS) of a perfect gas $p=(\gamma-1)\rho\epsilon$, where ϵ is the specific internal energy, γ the ratio of specific heats and the total energy is constituted of the internal and kinetic energy

$$E = \rho\left((u^2+v^2+w^2)/2+\epsilon\right).$$

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

At last, vector $U=(\rho,\rho u,\rho v,\rho w,E)$ represents the conservative variables of the system while $W=(\rho,u,v,w,p)$ are the primitive ones. Note that contrarily to WENO methods we do not use the characteristic variables.

To provide accurate and oscillation-free solutions we use on the one hand polynomial reconstruction and apply, on the other hand the MOOD algorithm. We mention that all the polynomial reconstructions are performed on conservative variables and only one CellPD is used for all variables (see [11] for the motivations and justifications). We now turn to the detection-limitation procedure and we have to design specific detection criteria for the Euler problem.

Following [11], a first and also mandatory detection criteria corresponds to ensuring the physical meaningfulness of the primitive variables. We then introduce the Physical Admissibility Detection (PAD in short) which considers that the candidate solution on a cell K_i is not valid if we have $\rho_i^\star$ or $p_i^\star$ are negative (after having computed pressure $p_i^\star$). We underline the important property that a high-order scheme (whichever the degree of the polynomial reconstruction) equipped with the PAD and a first-order scheme which preserves the positivity (of density and pressure) under a CFL condition is automatically positivity preserving. This property straightforwardly derives from the *a posteriori* nature of the MOOD method and has been proved in [11].

However the PAD detection process does not prevent spurious oscillations from appearing and we turn to the adaptation of the [DMP$\to u2$] detection process proposed in [11]. Initially defined for scalar quantity, we apply the [DMP$\to u2$] on the density ρ (detection) and recall that the decrementing is performed for all variables (limitation). Note that the smoothness parameter ε is still set as $1/2$ in the $u2$ definition as in previous section.

The set of constraints $\mathcal{A}$ for Euler system is thus constituted by the PAD followed by the [DMP$\to u2$] detection process applied to the density variable since we first check the PAD and if the cell is valid we continue with the [DMP$\to u2$|detection. In Figure 5 we give an algorithmic view of the complete detection process [PAD$\to$DMP$\to u2$] for the hydrodynamics Euler system constituted of the PAD detection criteria, the DMP of equation (8) on the density relaxed by the $u2$ detection criteria of definition 3.1. We emphasize that the algorithm is given in the case of a cell K_i with $U_i^\star=\left(\rho_i^\star,(\rho u)_i^\star,(\rho v)_i^\star,(\rho w)_i^\star,E_i^\star\right)$ its associated candidate solution mean value and that the candidate pressure $p_i^\star$ has to be computed.

We now highlight some implementation aspects about the detection process which enable to improve the solution accuracy. Actually in the above algorithm, the [PAD$\to$DMP$\to u2$] performs well but does not, in some cases, fully reach the optimal order of accuracy for smooth solutions. Deeper investigations on the isentropic vortex in motion problem have shown that the detection process inappropriately decrements some cells of the flat region while it operates well in the area where curvatures are not negligible. The undesirable limitation derives from the extra-small *curvatures* treatment by the $u2$ detection where some spurious micro-oscillations take place on the flat area and wrongly activate the curvature sign detection. It results that the sign criterion is not relevant when all the *curvatures* sizes are too small with respect to a mesh parameter δ. To overcome the over-detection phenomena, we introduce a relaxation parameter in the $u2$ criterion to fix the problem.

Definition 3.4 (**u2** detection criterion)**.** *A candidate solution $U_i^\star$ in cell K_i for which the density $\rho_i^\star$ violates the DMP is nonetheless eligible if*

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > -\delta \quad \textit{and} \quad \left(\max\left(|\mathcal{X}_i^{max}|,|\mathcal{X}_i^{min}|\right) < \delta \quad \textit{or} \quad \left|\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right| \geq 1/2\right),$$

$$\textit{and} \quad \mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > -\delta \quad \textit{and} \quad \left(\max\left(|\mathcal{Y}_i^{max}|,|\mathcal{Y}_i^{min}|\right) < \delta \quad \textit{or} \quad \left|\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}\right| \geq 1/2\right),$$

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

$$\textit{and} \quad \mathcal{Z}_i^{max}\mathcal{Z}_i^{min} > -\delta \quad \textit{and} \quad \left(\max\left(|\mathcal{Z}_i^{max}|,|\mathcal{Z}_i^{min}|\right) < \delta \quad \textit{or} \quad \left|\frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}\right| \geq 1/2\right),$$

where δ is the greatest length of geometrical entity of dimension one defined by the length of the cells in $\mathbb{R}$, the maximal length of the cell interfaces in $\mathbb{R}^2$ and the maximal length of edges of the cell interface for three-dimensional meshes.

The correction only damps extra-small oscillations such that minimal and maximal curvatures product satisfies the left condition. When maximal curvatures are larger than δ, the condition on the ratios of curvatures implies that the underlying function will be considered as non-smooth.

Remark 3.5. *The value of δ has been determined after numerous simulation experiments. It enables to fully reach the optimal order for the Euler system but does not affect the method in wisely capturing discontinuous profiles. The correction has even been tested for the convection equation and accuracy losses have not been reported.*

In the same way, we slightly relax the DMP criteria to reduce the computational effort to avoid the waste of resources when performing the $u2$ detection criterion on plateaus. We consider that a DMP violation is not relevant if

$$\max_{j\in\overline{\mathcal{V}}(i)}(\rho_i^{RK}, U_j^{RK}) - \min_{j\in\overline{\mathcal{V}}(i)}(\rho_i^{RK}, U_j^{RK}) < \delta^3$$

where index ^{RK} corresponds to one of the Runge-Kutta sub-steps.

The MOOD method for the Euler hydrodynamics system is now completely defined and numerical simulations are carried out for three-dimensional geometries presented in section 4.

If $[\rho_i^\star \leq 0]$ **or** $[p_i^\star \leq 0]$ **then**
- decrement the CellPD d_i

else
If $\left[\rho_i^\star \leq \min_{j\in\overline{\mathcal{V}}(i)}(\rho_i^n, \rho_j^n)\right]$ **or** $\left[\rho_i^\star \geq \max_{j\in\overline{\mathcal{V}}(i)}(\rho_i^n, \rho_j^n)\right]$ **then**
- decrement the CellPD d_i

else
→ compute the curvatures $\mathcal{X}_i^{min}, \mathcal{X}_i^{max}, \mathcal{Y}_i^{min}, \mathcal{Y}_i^{max}, \mathcal{Z}_i^{min}, \mathcal{Z}_i^{max}$ from density ρ at t^n
If $\left[\mathcal{X}_i^{min}\mathcal{X}_i^{max} < 0\right]$ **or** $\left[\mathcal{Y}_i^{min}\mathcal{Y}_i^{max} < 0\right]$ **or** $\left[\mathcal{Z}_i^{min}\mathcal{Z}_i^{max} < 0\right]$ **then**
- decrement the CellPD d_i

else
→ compute the curvatures ratios $\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}, \frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}, \frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}}$
If $\left[\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}} < 1-\varepsilon\right]$ **or** $\left[\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}} < 1-\varepsilon\right]$ **or** $\left[\frac{\mathcal{Z}_i^{min}}{\mathcal{Z}_i^{max}} < 1-\varepsilon\right]$ **then**
- decrement the CellPD d_i

end if
end if
end if
end if

Figure 5. Algorithmic view of the [PAD$\to$DMP$\to u2$] detection process for the Euler system.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

3.3. Implementation and optimizations

To conclude the section we detail two important and simple optimizations that we apply to drastically improve the efficiency of the MOOD method.

Local re-updating. The MOOD method may seem computationally expensive since the MOOD algorithm we run for each time step, recompute the candidate solution several times whereas polynomial degrees have been only modified for a small number cells. At the first stage, an initial candidate solution is computed on all cells. Then the MOOD algorithm successively detects and limits the problematic cells. The evaluation of a new candidate solution during the MOOD algorithm by means of scheme (4) only involves the fluxes at the interfaces of corrected cells. Consequently only problematic cells and their neighbors by face must be recomputed. It drastically reduces the computational effort since in most cases the solution is acceptable on more than 80–90% of cells, even when shocks are present.

Reduced polynomial degree decrementing. The original decrementing procedure consists in dropping one-by-one polynomial degrees until zero is reached. Such an approach may be both costly in CPU and memory resources since reconstruction matrices must be stored for all degrees. It would nonetheless still be less memory consuming than for the WENO method due to the large number of polynomial functions involved in the WENO technique. Moreover numerical experiments suggest the following alternative: whether the solution is very smooth, whether the solution presents some discontinuities. To take advantage of it, we change the decrementing strategy by starting from the highest degree, reducing to degree 2 if any and setting degree equal to 0 if the candidate solution is still not $\mathcal{A}$-eligible. We then manage to reduce the number of decrementing stages and save computational resources. We point out that the size of the reconstruction stencil is also an important parameter since a large stencil (required for the maximal degree) will be influenced by a discontinuity located in the second or third layer of cells around the reference one while a more compact one (for a $\mathbb{P}_2$ reconstruction) still preserves the local regularity of the underlying function. Another reason to use the $\mathbb{P}_2$ reconstruction is that it is also used for the $u2$ detection process and always has to be stored.
Therefore in practice, we only store two reconstruction matrices per cell, one for the maximal degree and one for the degree two. It is thus important to remark that the storage cost of the matrix for degree two is always much lower than the one for the maximal degree. Indeed for two-dimensional situations, the memory cost of the pseudoinverse matrix associated to polynomial of degree 2 represents about 10 times 5 elements, about 16 times 9 for $\mathbb{P}_3$ reconstruction and around 28 times 20 for $\mathbb{P}_5$. Analogically for three-dimensional situations, the $\mathbb{P}_2$ reconstruction matrix represents about 16 times 9 elements while it is about 38 times 19 for $\mathbb{P}_3$ and 110 times 55 for $\mathbb{P}_5$.

To conclude this section, we would like to draw some remarks about the potentiality of the MOOD method to be parallelized. Within the MOOD algorithm, only classical unlimited schemes are used without modification so that the parallelization of this part of the method can be done as efficiently as the state-of-the-art methods (WENO method for instance). The only novelty brought by the MOOD method is the iterative process constituting the MOOD algorithm. A potential difficulty comes from the fact that the number of cells on which the numerical scheme acts changes from an iteration of the MOOD algorithm to another, since the procedure is only applied to problematic cells. However it may not dramatically affect the parallelization efficiency: firstly, because an efficient treatment of the list of problematic cells can be achieved and secondly, because the time spent to recompute new candidate solutions is negligible compared to the time to compute the initial one since the number of problematic cells is (in general) very low compared to the total number of cells. The parallelization capacity of the MOOD method is thus as good as the state-of-the-art higher-order finite volume methods.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

4. NUMERICAL RESULTS

The MOOD method has been implemented into a 3D unstructured code dealing with polyhedra having coplanar faces: tetrahedron, hexahedron, pyramid and prism. The polynomial reconstruction procedure is implemented independently of the degree d_{max} and we provide in the present paper numerical results up to $d_{max} = 5$. Following [11] and remarks in section 3.3, we use the decrementing sequence $\mathbb{P}_{d_{max}} - \mathbb{P}_2 - \mathbb{P}_0$. The reconstruction matrices are computed and stored in a preprocessing step since they only depend on geometry. Moreover fluxes across faces are approximated by the mean of Gaussian quadrature formulae on a triangular decomposition of the faces (see Figure 1). At last concerning the time discretization, the first-order time step Δt is controlled by a CFL coefficient equal to 0.5. For the convergence studies on smooth solutions we use the time step $\Delta t = \Delta x^{r/3}$ to achieve a global r^{th}-order of accuracy and compute the relative L^1 and L^∞ errors for a a bounded, L^1 function φ by

$$L^1 \text{ error: } \frac{\sum_{i\in\mathcal{E}_{el}} |\varphi_i^N - \varphi_i^{ex}||K_i|}{\sum_{i\in\mathcal{E}_{el}} |\varphi_i^{ex}||K_i|} \quad \text{and} \quad L^\infty \text{ error: } \frac{\max_{i\in\mathcal{E}_{el}} |\varphi_i^N - \varphi_i^{ex}|}{\max_{i\in\mathcal{E}_{el}} |\varphi_i^{ex}|},$$

where $(\varphi_i^{ex})_{i\in\mathcal{E}_{el}}$ and $(\varphi_i^N)_{i\in\mathcal{E}_{el}}$ are respectively the exact and the approximated cell mean values at final time $t = t_{\text{final}}$.

4.1. Advection equation

For the scalar advection equation, the MOOD method is employed with the [DMP→$u2$] detection process and two test cases are carried out: the Triple Sine Translation (TST) to assess the effective very high-order of accuracy and the rotation of a discontinuous H-like shape to test its ability to damp the spurious oscillations.

4.1.1. Triple Sine Translation Let Ω be the unit cube. We consider a constant translation velocity $V = (1, 1, 1)$ and the C^∞ initial condition

$$U_0(x, y, z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z).$$

The final time is $t_{\text{final}} = 2.0$ and periodic boundary conditions imply that the exact final solution coincides with the initial one. The computations are first carried out on a series of successively refined regular hexahedral meshes from 8^3 to 64^3 cells. To underline the capability of the MOOD method to handle mixed element meshes, we also consider a series of meshes built from a series of regular hexahedral meshes from 4^3 to 48^3 cells into which we regularly split half of cells into 6 pyramids (see top line of Figure 6).

In Figure 6, we display the convergence curves for the L^1 and L^∞ errors of the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ methods and give in Table I, the corresponding errors and rates of convergence. As expected, the optimal rate of convergence is achieved. Notice that on the coarsest meshes the initial mean values are not representative of the underlying smooth function and are coherently handled by the method as discontinuous profiles. As such the sine function is under-resolved; for instance in 1D, averaging the function $\sin(2\pi x)$ or an Heaviside-like function on $[0; 1]$ using four cells provides to same mean values.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

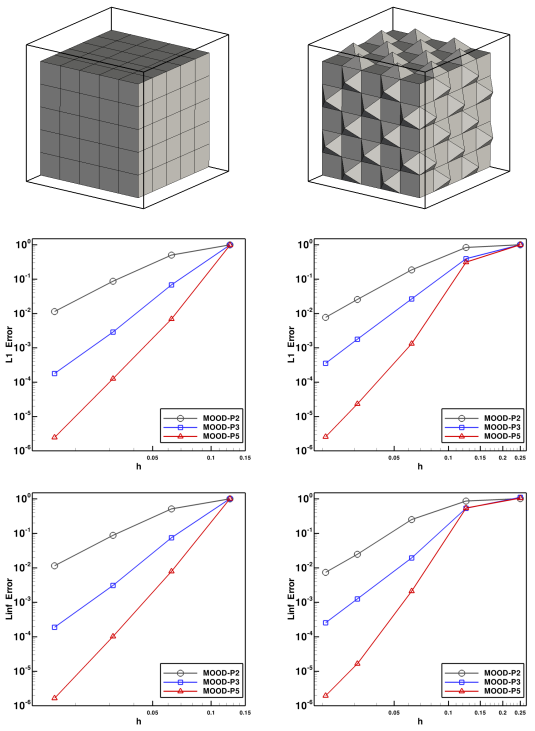


Figure 6. Triple sine translation: convergence curves for L^1 (middle) and L^∞ (bottom) errors for series of hexahedral (left) and hexahedral/pyramidal (right) meshes. Examples of such meshes are given on top line.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

MOOD on hexahedra

Deg.	h	L^1 error		L^2 error		L^∞ error	
$\mathbb{P}_2$	0.125	9.9682484294e-1	—	9.9680699621e-1	—	1.0038484635	—
	0.0625	5.0100730661e-1	0.99	5.0179199583e-1	0.99	5.1327069642e-1	0.97
	0.03125	8.6946371321e-2	2.53	8.6934402652e-2	2.53	8.7634409525e-2	2.55
	0.015625	1.1429255953e-2	2.93	1.1424251727e-2	2.93	1.1459635442e-2	2.93
Expected order			3		3		3
$\mathbb{P}_3$	0.125	9.8673015719e-1	—	9.8688657294e-1	—	1.0110942145	—
	0.0625	6.8019691177e-2	3.86	6.9068026293e-2	3.84	7.4325569597e-2	3.77
	0.03125	2.8693653411e-3	4.57	2.7741019990e-3	4.64	3.0819084097e-3	4.59
	0.015625	1.7856449887e-4	4.01	1.5709922281e-4	4.14	1.8874795858e-4	4.03
Expected order			4		4		4
$\mathbb{P}_5$	0.125	9.7842521971e-1	—	9.7924733246e-1	—	1.0169454936	—
	0.0625	6.9230110414e-3	7.14	6.9967747247e-3	7.13	7.9478234947e-3	7.00
	0.03125	1.2666634416e-4	5.77	1.1021776542e-4	5.99	1.0247433118e-4	6.28
	0.015625	2.4614368833e-6	5.69	2.0386571852e-6	5.76	1.6605387870e-6	5.95
Expected order			6		6		6

MOOD on mixed hexahedra/pyramids

Deg.	h	L^1 error		L^2 error		L^∞ error	
$\mathbb{P}_2$	0.25	1.0000027468	—	1.0000052406	—	1.0143912168	—
	0.125	8.3799247906e-1	0.25	8.3412664416e-1	0.26	8.6799172420e-1	0.20
	0.0625	1.8662020042e-1	2.17	1.8646014762e-1	2.16	2.5210598518e-1	1.78
	0.03125	2.5647018453e-2	2.86	2.4729005004e-2	2.91	2.4798614346e-2	3.35
	0.020833	7.6897099615e-3	2.97	7.4071918780e-3	2.97	7.3982102275e-3	2.98
Expected order			3		3		3
$\mathbb{P}_3$	0.25	9.9952627605e-1	—	1.0018017690	—	1.1083447073	—
	0.125	3.9219135702e-1	1.35	4.1718119801e-1	1.26	5.3531820180e-1	1.05
	0.0625	2.6501056786e-2	3.89	2.2797888150e-2	4.19	1.9364138004e-2	4.79
	0.03125	1.7829686262e-3	3.90	1.5178093945e-3	3.91	1.2521397100e-3	3.95
	0.020833	3.5401059785e-4	3.99	3.0038884987e-4	4.00	2.5551614705e-4	3.92
Expected order			4		4		4
$\mathbb{P}_5$	0.25	1.0009285907	—	1.0025919881	—	1.0496962436	—
	0.125	3.1141644019e-1	1.68	4.0086400280e-1	1.32	5.4249781220e-1	0.95
	0.0625	1.3246287256e-3	7.88	1.1541322861e-3	8.44	2.1098853389e-3	8.00
	0.03125	2.3443624169e-5	5.82	2.0015998229e-5	5.85	1.6596915121e-5	6.99
	0.020833	2.0207215760e-6	6.05	1.7188485947e-6	6.05	1.5127171717e-6	5.91
Expected order			6		6		6

Table I. L^1, L^2 and L^∞ errors and convergence rates for the TST problem with the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ methods. Top lines: hexahedral meshes. Bottom lines: mixed hexahedral/pyramidal meshes.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

4.1.2. H-like shape rotation We now turn to the rotation of an H-like shape in the unit cube Ω. The initial shape is given by

$$U_0(x,y,z) = \begin{cases} 1 & \text{if } (|x-0.5| > 0.1) \text{ or } (|y-0.5| < 0.1), \\ 0 & \text{elsewhere,} \end{cases}$$

in the cube $[0.2;0.8]^3$ and 0 elsewhere. The rotation axis is the diagonal line joining the origin $(0,0,0)$ and the point $(1,1,1)$. We stop the simulation after one full rotation when the shape is back to its original position. Note that the velocity depends on the spatial position but is divergence-free so that the maximum principle also applies in that case. Numerical simulations are carried out on a 86215 tetrahedra mesh generated by the free mesher Gmsh. Results are displayed with an extruded view on the cut plane $z = 1/2$. Initialization details are illustrated in Figure 7.

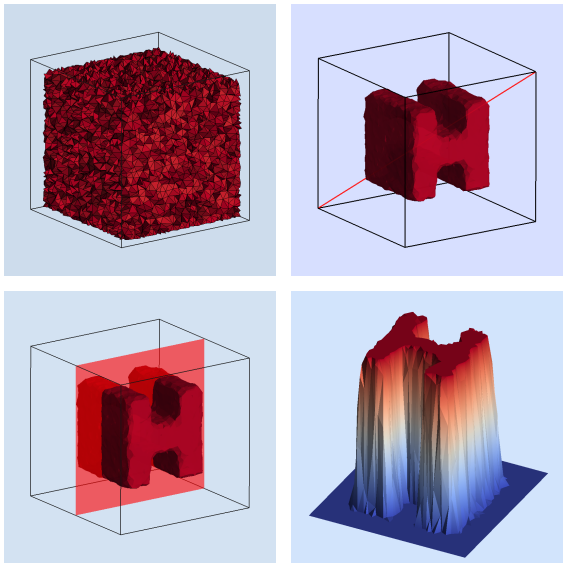


Figure 7. Initialization of the H-like shape rotation problem. Top left: interior view of the tetrahedral mesh. Top right: initialization of the H-like shape (isosurface $1/2$, rotation axis is the red line). Bottom left: cut plane $z = 1/2$. Bottom right: extruded initial values from the cut plane.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

We plot in Figure 8 the solution on the cut plane $z = 1/2$ for the unlimited $\mathbb{P}_3$ and $\mathbb{P}_5$ schemes and the MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ methods. We notice that the unlimited schemes produce oscillations, depicted in green in the figure, whereas the MOOD method provides an oscillation-free solution even for polynomials of degree 5. It highlights the capacity of the [DMP$\to u2$] detection process to correctly treat discontinuous shapes on genuinely unstructured 3D meshes.

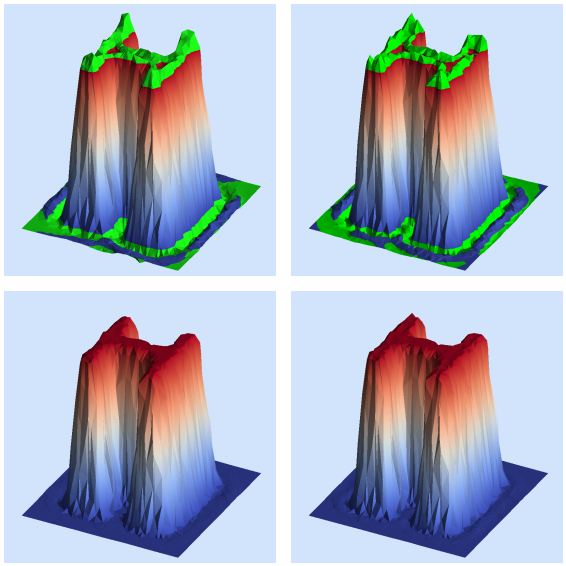


Figure 8. Results of the H-like shape rotation problem on the cut plane $z = 1/2$ for the unlimited $\mathbb{P}_3$ and $\mathbb{P}_5$ schemes (top line) and for the MOOD $\mathbb{P}_3$ and $\mathbb{P}_5$ methods (bottom line). The highlighted green cells correspond to values below 0 or above 1.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

4.2. Euler system

We now consider the three-dimensional hydrodynamics Euler system on unstructured meshes. The first test cases proposed in section 4.2.1 deal with the Sod and Lax shock tubes following the Ox axis (invariant with respect to the other directions). The simulations are carried out on a tetrahedral mesh to study the MOOD method capacity to handle simple waves. Section 4.2.2 is dedicated to the Shu-Osher and Blastwave problems approximated on pyramidal cells which respectively involve a complex oscillatory solution and strong interactions between simple waves along the Ox direction. We address in section 4.2.3 the effective numerical accuracy of the method with the isentropic vortex problem for which an exact smooth solution exists. Then in section 4.2.4, we assess the ability of the MOOD method to simulate complex realistic physics on a mesh of triangular and quadrangular prisms by carrying out the impact of a shock wave on a cylindrical cavity proposed in [11]. At last, we provide the MOOD method results for two genuinely three-dimensional test cases. First we compare the behavior and computational cost (CPU and memory storage) of the MOOD method with different degrees and detection processes by simulating the so-called explosion problem [33] using unstructured pyramidal meshes in section 4.2.5; Then in section 4.2.6, we consider the interaction of a shock wave with a quarter of cone on a mesh of 1.1 millions of tetrahedra with the 4^{th}-order MOOD method.

4.2.1. Sod and Lax shock tubes The original Sod [31] and Lax [24] problems concern one-dimensional Riemann shock tubes whose solutions consist of a left-moving rarefaction fan, a right-moving contact discontinuity and a right-moving shock wave. In the three-dimensional context, we reproduce the expansion following the Ox axis setting initial condition invariant in y, z and we prescribe reflecting boundary conditions on the cylinder sides. The domain is filled with an ideal gas with $\gamma = 1.4$ and the discontinuity is located in $x = 0.5$ at $t = 0$. The initial density/velocity/pressure values and final time t_{final} are given by

- Sod: $(\rho,u,p)_L = (1.0, 0.0, 1.0)$ and $(\rho,u,p)_R = (0.125, 0.0, 0.1)$, $t_{\text{final}} = 0.2$,
- Lax: $(\rho,u,p)_L = (0.445, 0.698, 3.528)$ and $(\rho,u,p)_R = (0.5, 0.0, 0.571)$, $t_{\text{final}} = 0.13$.

The computational domain we consider is a cylinder of unit length and radius $R = 0.025$ with $0x$ line as symmetry axis which is paved with 7517 unstructured tetrahedra as shown in figure 9.

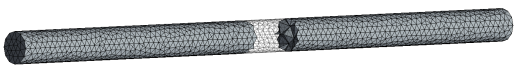


Figure 9. Mesh constituted of 7517 tetrahedra used for the Sod and Lax problems. Some cells are drawn non-opaque to see some interior tetrahedra.

We display in Figure 10 the numerical approximations of the density computed with the MOOD-$\mathbb{P}_3$ method using the [PAD$\to$DMP$\to u2$] detection process and the exact solution (red line). In order to provide a clear and relevant representation of the solution along the Ox axis, we slice the whole cylinder in 100 uniform cylinders (since the average characteristic length is 10^{-2}) and plot the average of the solution on each of them. As expected the MOOD-$\mathbb{P}_3$ method provides a very good approximation of the solution and maintains sharp discontinuities. In particular, we underline the very few numbers of points in the contact discontinuity.

4.2.2. Shu-Osher and Blastwave problems The Shu-Osher problem has been introduced in [29] to test the ability of a scheme to capture both small-scale smooth flow along with shock wave. The one-dimensional computational domain is $\Omega = [-5;5]$ and the final time is $t_{\text{final}} = 1.8$. An initial x-directional shock wave located at $x = -0.4$ separates the domain into a left post-shock state

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

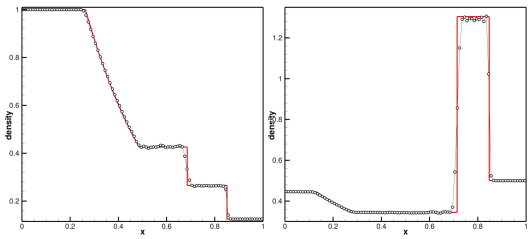


Figure 10. The MOOD-$\mathbb{P}_3$ density results are displayed for the Sod (left) and Lax (right) problems on tetrahedral mesh *vs* the exact solution (red line).

$(\rho,u,p)_L = (3.857143, 2.629369, 10.333333)$ and a right state $(\rho,u,p)_R = (1 + 0.2\sin(5x), 0, 1.0)$. We consider a perfect gas with $\gamma = 1.4$. Reflecting boundary conditions are used to preserve the invariance following axis Oy, Oz except from the left boundary condition which is an inflow one.

The Blastwave problem has been introduced by Collela and Woodward in [10] to test the performance of numerical schemes on problems involving strong and thin shock structures. The initial conditions consist of two parallel planar flow discontinuities on domain $\Omega = [0,1]$ separated by the planes $x_1 = 0.1$ and $x_2 = 0.9$. The density is unity on the whole domain and the gas is assumed initially at rest. The pressure is given by $p_L = 1000$ on the left, $p_C = 0.01$ in the center and $p_R = 100$ on the right. Reflecting boundary conditions are prescribed and the final time is $t_{\text{final}} = 0.038$.

We consider a 21600 regular pyramids mesh (see Figure 12-top right for a pattern example) obtained from a $400 \times 3 \times 3$ regular hexahedral mesh for which each cell is split into six pyramids. The original hexahedral mesh is built by setting $\Delta x = \Delta y = \Delta z$ with $\Delta x = 0.075$ for Shu-Osher problem and $\Delta x = 0.0075$ for the Blastwave problem. Since there is no exact solution for both tests we have computed reference solutions using a first-order finite volume scheme with very fine meshes. As in the previous simulations, the solutions are plotted following the Ox direction considering an underlying 400 points uniform one-dimensional mesh and circles in Figure 11 represent the mean density on three-dimensional slices of thickness Δx.

Density approximations obtained with the MOOD-$\mathbb{P}_3$ are presented in Figure 11 and compared to the reference solution (red line). For the Shu-Osher problem (left) we report that the [PAD$\to$DMP$\to u2$] detection criteria does not over-smooth the oscillatory solution and accurately capture the high-frequencies waves. On the other hand, for the Blastwave problem (right) we observe sharp contact discontinuities and shock waves are well-preserved. No spurious oscillations are generated and the central structure of the solution is very well approximated.

4.2.3. Isentropic vortex The isentropic vortex problem was initially introduced for the two-dimensional space [28, 38] to test the accuracy of numerical methods since the exact solution is smooth and has an analytical expression. We simply extend the original problem for the three-dimensional situation taking the two-dimensional solution invariant following Oz. Let us consider the computational domain $\Omega = [-5,5] \times [-5,5] \times [0, z_{max}]$ and an ambient flow characterized with $\rho_\infty = 1.0$, $u_\infty = 1.0$, $v_\infty = 1.0$, $w_\infty = 1.0$, $p_\infty = 1.0$, with a normalized ambient temperature

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

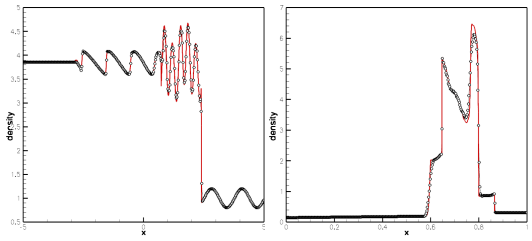


Figure 11. Results for the Shu-Osher (left) and Blastwave (right) problems on pyramids. MOOD-$\mathbb{P}_3$ density results are displayed on the left and right columns respectively *vs* the reference solution (red line).

$T^*_\infty = 1.0$ computed with the perfect gas equation of state and $\gamma = 1.4$.
A z-invariant vortex is centered on the axis line $\mathbf{x}_{\text{vortex}} = (x_{\text{vortex}}, y_{\text{vortex}}, z) = (0, 0, z)\ z \in \mathbb{R}$ and supplemented to the ambient gas at the initial time $t = 0$ with the following conditions $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, $T^* = T^*_\infty + \delta T^*$ where

$$\delta u = -y' \frac{\beta}{2\pi} \exp\left(\frac{1-r^2}{2}\right), \quad \delta v = x' \frac{\beta}{2\pi} \exp\left(\frac{1-r^2}{2}\right), \quad \delta T^* = -\frac{(\gamma-1)\beta}{8\gamma\pi^2} \exp\left(1-r^2\right),$$

with $r = \sqrt{x'^2 + y'^2}$ and $x' = x - x_{\text{vortex}}$, $y' = y - y_{\text{vortex}}$. The vortex strength is given by $\beta = 5.0$ and the initial density follows relation

$$\rho = \rho_\infty \left(\frac{T^*}{T^*_\infty}\right)^{\frac{1}{\gamma-1}} = \left(1 - \frac{(\gamma-1)\beta}{8\gamma\pi^2} \exp\left(1-r^2\right)\right)^{\frac{1}{\gamma-1}}. \tag{14}$$

The domain is paved either with $N \times N \times 4$ hexahedra , $N = 20$, 40, 60, 80, 120 or with $N \times N \times 24$ pyramids (each hexahedron from the previous mesh is split into 6 pyramids, see Figure 12). To reduce the computational effort, only four cells are considered in the z-direction and z_{max} is taken such that $\Delta x = \Delta y = \Delta z$, that is to say $z_{max} = 4\Delta x = 40/N$. The minimal/maximal number of cells is 1600/57600 hexahedra and 9600/153600 pyramids. We prescribe periodic boundary conditions everywhere.

In Figure 12 we display the convergence curves for the L^1 and L^∞ errors on the density approximations for MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ methods, while we provide in Table II the corresponding errors and convergence rates. We report effective orders corresponding to the expected optimal rates of convergence for both types of meshes and underline the MOOD method capacity to provide effective high-order of accuracy on a smooth but non-trivial solution for the three-dimensional Euler system.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

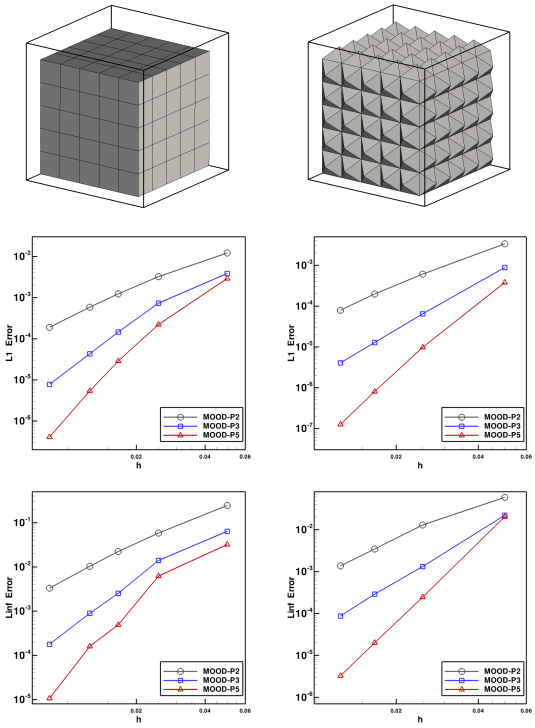


Figure 12. Isentropic vortex in motion: convergence curves for L^1 (middle) and L^∞ (bottom) errors for series of hexahedral (left) and pyramidal (right) meshes. Examples of such meshes are given on top line.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

Deg.	Cell nb	L^1 error		L^2 error		L^∞ error	
MOOD on hexahedra							
$\mathbb{P}_2$	$20 \times 20 \times 4$	1.2149125472e-2	—	3.1900165943e-2	—	2.4441600308e-1	—
	$40 \times 40 \times 4$	3.2245099895e-3	1.91	8.0410260168e-3	1.98	5.8071095773e-2	2.07
	$60 \times 60 \times 4$	1.2274201731e-3	2.38	3.0952199533e-3	2.35	2.2241472168e-2	2.37
	$80 \times 80 \times 4$	5.8449248920e-4	2.57	1.4891720330e-3	2.54	1.0333244490e-2	2.66
	$120 \times 120 \times 4$	1.8870676632e-4	2.78	4.8726522430e-4	2.75	3.2966795901e-3	2.82
Expected order			3		3		3
$\mathbb{P}_3$	$20 \times 20 \times 4$	3.8426301161e-3	—	9.2866634542e-3	—	6.3860302401e-2	—
	$40 \times 40 \times 4$	7.2909293814e-4	2.39	1.6463735019e-3	2.49	1.3928173243e-2	2.19
	$60 \times 60 \times 4$	1.4537313954e-4	3.97	3.4725838178e-4	3.84	2.5316808689e-3	4.20
	$80 \times 80 \times 4$	4.3014601762e-5	4.23	1.1403422006e-4	3.87	8.9234157884e-4	3.62
	$120 \times 120 \times 4$	7.7653485653e-6	4.22	2.0827671718e-5	4.19	1.7793186459e-4	3.98
Expected order			4		4		4
$\mathbb{P}_5$	$20 \times 20 \times 4$	2.8991068920e-3	—	4.8543664172e-3	—	3.2038381504e-2	—
	$40 \times 50 \times 4$	2.2151699683e-4	3.71	5.5851141683e-4	3.12	6.2194475329e-3	2.36
	$60 \times 60 \times 4$	2.8610132561e-5	5.04	7.5286576723e-5	4.94	4.9068468256e-4	6.26
	$80 \times 80 \times 4$	5.4168534310e-6	5.78	1.5519206048e-5	5.49	1.5955744462e-4	3.90
	$120 \times 120 \times 4$	4.0840597698e-7	6.38	1.1795674119e-6	6.36	1.0709587465e-5	6.66
Expected order			6		6		6
MOOD on pyramids							
$\mathbb{P}_2$	$20 \times 20 \times 24$	3.3660908651e-3	—	8.2020368268e-3	—	5.8966752971e-2	—
	$40 \times 40 \times 24$	6.0800306087e-4	2.47	1.4780372369e-3	2.47	1.2917288297e-2	2.19
	$60 \times 60 \times 24$	1.9831385885e-4	2.76	5.0256415975e-4	2.66	3.4489695638e-3	3.25
	$80 \times 80 \times 24$	7.9096059248e-5	3.19	2.0028509695e-4	3.19	1.3642153624e-3	3.22
Expected order			3		3		3
$\mathbb{P}_3$	$20 \times 20 \times 24$	8.8005733635e-4	—	2.0405839361e-3	—	2.2060839273e-2	—
	$40 \times 40 \times 24$	6.4460987694e-5	3.77	1.4763173293e-4	3.78	1.3204082077e-3	4.06
	$60 \times 60 \times 24$	1.2809782719e-5	3.98	2.9223354775e-5	3.99	2.8960192576e-4	3.74
	$80 \times 80 \times 24$	4.0713141263e-6	3.98	9.3121356054e-6	3.97	8.6899534957e-5	4.18
Expected order			4		4		4
$\mathbb{P}_5$	$20 \times 20 \times 24$	3.7944742185e-4	—	9.8016940506e-4	—	2.0273963181e-2	—
	$40 \times 40 \times 24$	9.7451977113e-6	5.28	2.4664732108e-5	5.31	2.4540502338e-4	6.36
	$60 \times 60 \times 24$	8.0304771455e-7	6.15	2.0569058735e-6	6.12	2.0022941712e-5	6.18
	$80 \times 80 \times 24$	1.2520658320e-7	6.46	3.1436119294e-7	6.53	3.2667224251e-6	6.30
Expected order			6		6		6

Table II. L^1, L^2 and L^∞ errors and convergence rates for the isentropic vortex problem with the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ methods. Top lines: hexahedral meshes. Bottom lines: pyramidal meshes.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

4.2.4. Impact of a shock wave on a cylindrical cavity Based on the experiment proposed in [30], we have introduced this test case in [11] for the two-dimensional case. We here extend it to 3D by invariance along the z-direction. It consists in a planar shock wave which impacts a cylindrical cavity creating complex structures and instabilities. The original purpose of this stringent numerical test is to prove the ability of the MOOD method to capture physics in realistic conditions. In this paper it moreover assesses the capacity of the MOOD method to deal with mixed triangular and quadrangular prisms since the mesh is obtained by extrusion (only two layers) along the Oz axis of a 2D mesh containing 101127 cells (triangles and quadrangles). We moreover point out that important differences between cell sizes are present in the domain, since the largest characteristic length is 0.008 and the smallest one is 0.00015. At last, we run the simulation on the lower half part of the domain but plot a full domain using a symmetry argument. Details of the mesh are provided in Figure 13.

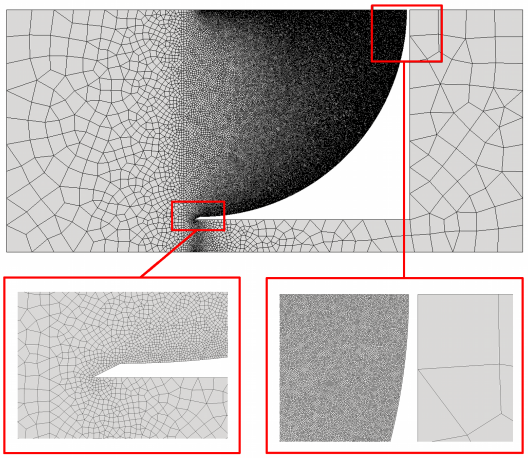


Figure 13. Impact of a shock on a cylindrical cavity: details of the mesh containing 202254 triangular and quadrangular prisms.

The detailed configuration and boundary conditions are provided in [11], and we recall that we consider the case of a nominal incident shock Mach number of 1.33 in ambient air (with $\gamma = 1.4$) at 0.95 bar pressure and that the variables initialization consists in the pre-shock values $(\rho, u, v, w, p) = (1.1175, 0.0, 0.0, 0.0, 95000.0)$ and the post-shock ones $(\rho, u, v, w, p) = (1.7522, 166.3435, 0.0, 0.0, 180219.75)$ leading to conditions of [30] at temperature $T = 296.15K$.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

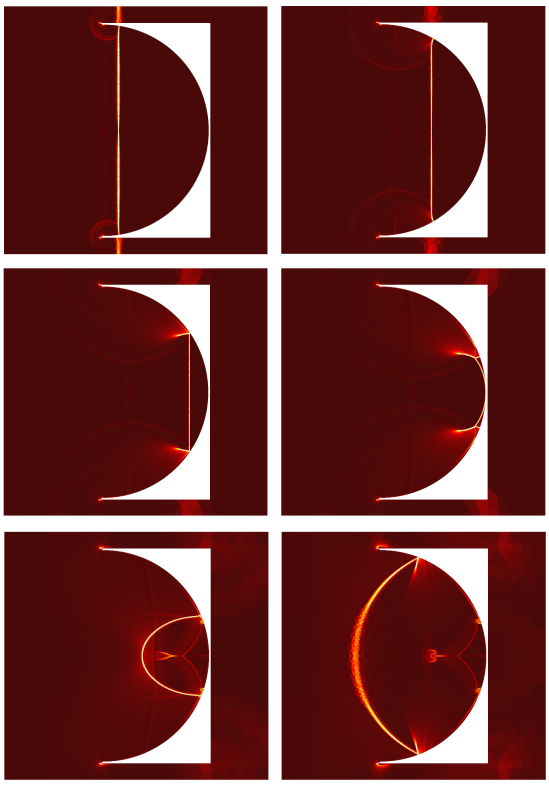


Figure 14. Impact of a shock on a cylindrical cavity: magnitude of the density gradient at different times from left to right and top to bottom.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

In Figure 14, we plot the magnitude of the density gradient computed with the MOOD $\mathbb{P}_2$ method equipped with the [PAD$\rightarrow$DMP$\rightarrow u2$] detection process at different times of the simulation in order to give an overview of the physical phenomena. We emphasize that the instabilities along the cylindrical wall are very well captured. Finally in Figure 15, we provide a zoom of the final solution on the created instabilities which perfectly match the experimental results of [30].

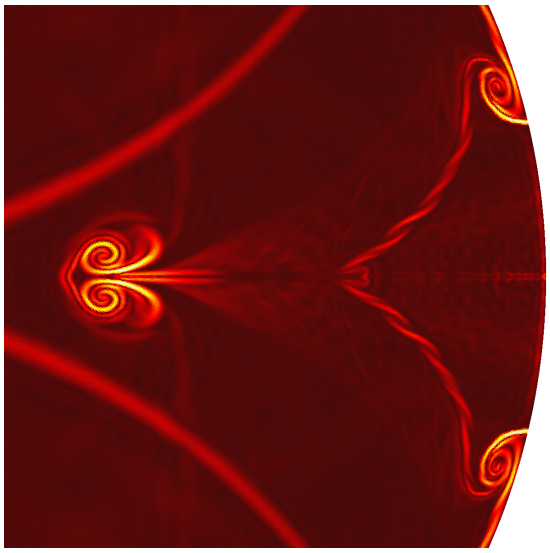


Figure 15. Impact of a shock on a cylindrical cavity: Zoom on the created instabilities at final time.

4.2.5. The explosion problem We consider the so-called explosion problem [33] given by a gas initially at rest in the unit cube where a quarter of the ball of radius $r_c = 0.4$ centered at the origin has a density $\rho_b = 1.0$, a pressure $p_b = 1.0$ whereas the exterior is characterized by $\rho_e = 0.125$, $p_e = 0.1$. The domain is partitioned into 20^3 hexahedral cells for which each hexahedron is split into 6 pyramids leading to a mesh of 48000 pyramids. Simulations are carried out till the final time $t_{\text{final}} = 0.25$. A reference solution has been computed with a two-dimensional cylindrical staggered numerical Lagrangian scheme [25].

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

We report in Figure 16 the density approximations in function of the radius for a classical MUSCL scheme [26], the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ methods equipped with the [PAD$\rightarrow$DMP$\rightarrow u2$] detection and the reference solution. Note that we use the same type of representation than for the previous test cases by slicing the radius in 100 uniform cells.

The solution shape is well reproduced by all methods and the higher the polynomial degree is the sharper the contact discontinuity and the shock wave are. In a three-dimensional context with discontinuous solutions, the MOOD-$\mathbb{P}_3$ method seems to be the right balance between accuracy and cost. The slight improvement gained by the MOOD-$\mathbb{P}_5$ compared to MOOD-$\mathbb{P}_3$ may not justify the computational over-cost (see further). We also notice that the head of the rarefaction wave is badly resolved by the MUSCL method whereas the MOOD-$\mathbb{P}_2$ and especially the MOOD-$\mathbb{P}_3$ method give accurate approximations.

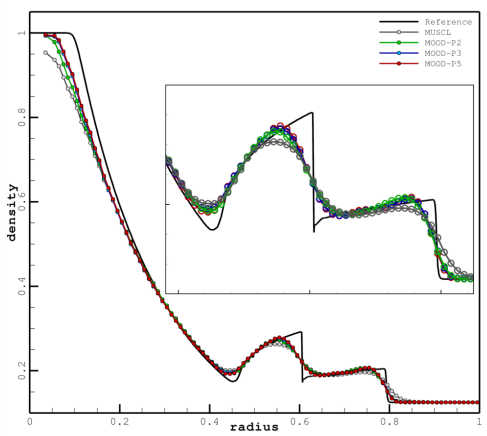


Figure 16. Density results for the explosion problem in 3D. Comparison between a classical MUSCL method and the MOOD-$\mathbb{P}_2$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ methods with [PAD$\rightarrow$DMP$\rightarrow u2$] detection process on tetrahedral mesh. The straight line corresponds to the reference solution.

To compare the different detection strategies, we present in Figure 17 the final solutions obtained by the MOOD method with the PAD alone and the [PAD$\rightarrow$DMP$\rightarrow u2$] detection processes using $\mathbb{P}_5$ polynomial reconstructions. Note that contrary to previous figures, we plot the density values for all cells by associating them with the radius corresponding to the cell centroid. As expected, the PAD detection process does not damp spurious oscillations close to the shock wave (see the zoom panel) and extra oscillations are also visible on the head of the rarefaction. We recall that the numerical approximation using the PAD detection process is the most accurate one on smooth solutions since only the physical admissibility of the solution is required so that few numerical diffusion is produced. On the opposite, the [PAD$\rightarrow$DMP$\rightarrow u2$] detection process damps the oscillations close

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

to the shock and to the head of rarefaction but also maintain a very good accuracy with a slight non monotonic behavior.

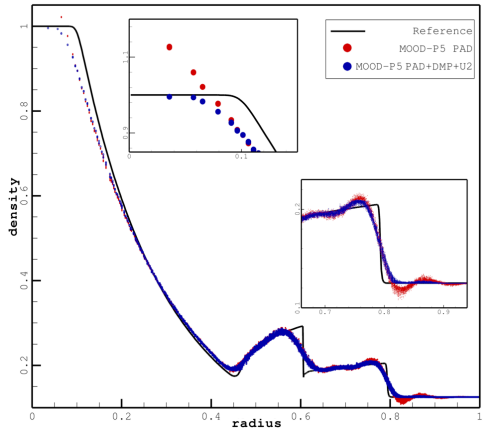


Figure 17. Density results for the explosion problem in 3D obtained with the MOOD-$\mathbb{P}_5$ method. Comparison between the PAD alone and [PAD$\rightarrow$DMP$\rightarrow u2$] detection process. The straight line corresponds to the reference solution and the symbols represent mean values of all cells.

Finally we provide in Tables III the computational cost of the MOOD method for this test case in this particular configuration when running on a single core of the three following machines (using -O3 flag for gfortran compiler):

M1: server with two Intel Xeon E5335 (4 cores) @ 2.00Ghz, 8MB of L2 Cache, 16GB of RAM
M2: laptop with Intel Core2Duo P7550 (2 cores) @ 2.26GHz, 3MB of L2 Cache, 8GB of RAM
M3: desktop with Intel Core i5 2500 (4 cores) @ 3.30GHz, 6MB of L2 Cache, 8GB of RAM

Note that the same three machines have been used in [11] to assess the computational cost of the MOOD method for two-dimensional geometries.

MOOD with [PAD$\rightarrow$DMP$\rightarrow u2$]	**Machine 1** Intel Xeon E5335 @ 2.00Ghz	**Machine 2** Intel Core2Duo P7550 @ 2.26GHz	**Machine 3** Intel Core i5 2500 @ 3.30GHz	**Memory storage**
MOOD-$\mathbb{P}_2$	66μs/it./cell	57μs/it./cell	30μs/it./cell	0.4 GB
MOOD-$\mathbb{P}_3$	163μs/it./cell	136μs/it./cell	69μs/it./cell	0.8 GB
MOOD-$\mathbb{P}_5$	439μs/it./cell	385μs/it./cell	185μs/it./cell	3.0 GB

Table III. CPU time in microseconds per iteration per cell and memory storage in Gigabytes for the MOOD-$\mathbb{P}_k$ methods ($k = 2, 3, 5$) with the [PAD$\rightarrow$DMP$\rightarrow u2$] detection process on three different computers.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

We first observe that the memory storage doubles when the polynomial degree is increased by one: 0.4 for $\mathbb{P}_2$, 0.8 for $\mathbb{P}_3$, 1.6 for $\mathbb{P}_4$ (not presented in the table) and 3.0 for $\mathbb{P}_5$. Notice that the memory consumption is very low since only two reconstruction pseudoinverse matrices (for $\mathbb{P}_2$ and $\mathbb{P}_{d_{max}}$) per cell are effectively stored. The CPU cost increases by a factor about 2.4 from $\mathbb{P}_2$ to $\mathbb{P}_3$ and about 2.7 from $\mathbb{P}_3$ to $\mathbb{P}_5$.

By extrapolation of these results we estimate the cost of the MOOD method for larger meshes. As instance for one million cells mesh and 1000 time steps the method cost should be:

- MOOD-$\mathbb{P}_2$ is 66000 seconds on M1, that is to say $\sim$ 18 hours ($\sim$ 16 hours on M2 and $\sim$ 8.3 hours on M3) with about 8 Gb of memory storage,
- MOOD-$\mathbb{P}_3$ is 163000 seconds, $\sim$ 2 days on M1 ($\sim$ 1.5 day and $\sim$ 19 hours on M2 and M3) with about 16 Gb of memory storage,
- MOOD-$\mathbb{P}_5$ is 439000 seconds, $\sim$ 5 days on M1 ($\sim$ 4.5 days and $\sim$ 2.1 days on M2 and M3) with about 62 Gb of memory storage.

Consequently simulations with nowadays sequential computers with a one million cells mesh (assuming one thousand time steps) can be obtained for about one day of computation with MOOD-$\mathbb{P}_3$ method. The MOOD method is thus a very competitive very high-order finite volume method, and these results shall be improved by an efficient parallelization.

4.2.6. Interaction of a shock wave with a quarter of cone To conclude the numerical tests section, we run the test case named *interaction of a shock wave with a quarter of cone* with the 4^{th}-order MOOD-$\mathbb{P}_3$ method equipped with the [PAD$\rightarrow$DMP$\rightarrow u2$] detection process. This 3D extension of the so-called *interaction of a shock wave with a wedge* has been proposed in [14] as instance.

The domain consists in a quarter on cylinder of radius $R = 2.25$ centered on the Ox axis which covers the interval $[-1.1; 3.0]$ in the x-direction. Note that three modifications have been made in comparison to [14] in order to reduce the computational cost: the test is run on a quarter of cylinder instead of a half one, the initial interface is placed at $x = -0.2$ instead of $x = -1.0$ and the domain covers in the x-direction the interval $[-1.1; 3.0]$ instead of $[-1.5; 3.0]$. Finally the mesh obtained by the free mesher Gmsh contains 1161854 tetrahedra in three refinement zones and exactly matches the initial interface, see top of Figure 19.

We recall that the circular cone under consideration is such that its length is 1, its tip and foot radii are 0.02 and 0.5 respectively while its tip is placed at the origin. Moreover wall boundary conditions are prescribed everywhere except from the top and bottom of the quarter of cylinder where the exact solution according to the Rankine-Hugoniot conditions is imposed. At last the initial pre- and post shock conditions are given by $(\rho, u, v, w, p) = (2.122, 0.0, 0.0, 0.0, 1.805)$ and $(\rho, u, v, w, p) = (1.4, 0.0, 0.0, 0.0, 1.0)$ respectively with $\gamma = 1.4$ and the final time is chosen such that it corresponds to the final time of [14].

In Figure 18, we propose numerical Schlieren-type images on the solution in the $Ox - Oy$ and $Ox - Oz$ planes. We remark that the symmetry is very well conserved since both images are almost identical and that all waves that are present in results of [14] are also resolved here although much less cells (more than 3.5 times less) are considered. This proves that the MOOD method performs very well on 3D unstructured meshes. Finally on bottom of Figure 19, we provide a 3D view for which isosurfaces have been chosen to represent the principal waves in the whole domain. It is thus clear that the method properly reproduces the cylindrical symmetry even on this fully unstructured 3D tetrahedral mesh.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

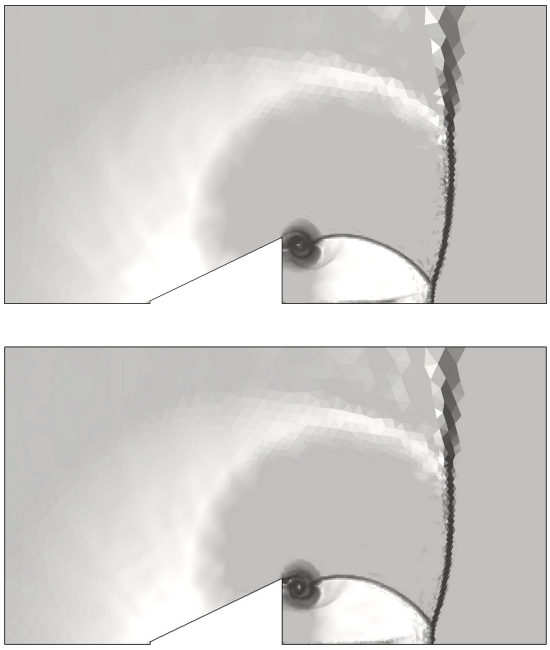


Figure 18. Interaction of a shock wave with a half cone: on top, numerical Schlieren-type image on the $Ox - Oy$ plane; on bottom, numerical Schlieren-type image on the $Ox - Oz$ plane.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

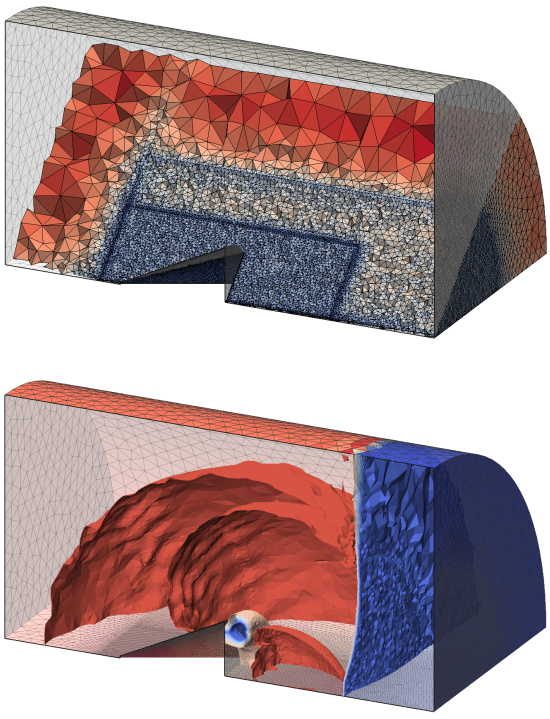


Figure 19. Interaction of a shock wave with a half cone: on top, view of the interior of the tetrahedral mesh with the different zones of refinement; on bottom, isosurfaces corresponding to the principal waves.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

5. CONCLUSION

In this paper we have proposed the three-dimensional extension of the so-called MOOD method [5, 11]. The *Multi-dimensional Optimal Order Detection* expression refers to an original way of determining the optimal local polynomial degree to be used in the reconstruction step of a classical high-order unlimited scheme. To each cell corresponds a polynomial reconstruction for which we *a posteriori* determine the degree according to given criteria (positivity as instance) against which we test each candidate solution. The detection criteria is based on a relaxed version of the discrete maximum principle (DMP) associated with a so-called $u2$ detection procedure which analyses the numerical curvatures in the neighborhood of a DMP violating cell and determine if the underlying function is regular or not. In the latter case the polynomial degree in the associated cell is decremented and the solution is locally recomputed. We have detailed the numerical method for three-dimensional unstructured meshes and improved the detection criteria both for the advection equation and the Euler system. Moreover some optimizations for the three dimensional case have been provided to significantly improve the efficiency of the method.

The MOOD method has been implemented on several kinds of unstructured meshes with $\mathbb{P}_k$ polynomial reconstructions (k varying from 1 to 5). We have provided some sanity checks with simple configurations and performed more advanced full three-dimensional tests to assess the ability of the MOOD method to accurately capture waves on real unstructured meshes. For the scalar convection equation with a regular initial shape the method gives an effective high-order of accuracy corresponding to the optimal one and we have shown that spurious oscillations are damped when discontinuous profiles are convected. The results for unidirectional problems for the Euler system with three-dimensional unstructured meshes show that small scaled structures are captured while shock waves are resolved within few cells. For the isentropic vortex test case extended to the three-dimensional context with non-trivial exact solution, effective high-orders of accuracy are measured and optimal orders are reported for $\mathbb{P}_2, \mathbb{P}_3$ and $\mathbb{P}_5$ polynomials. We prove that the MOOD method is able to capture the realistic physics of the impact of a shock wave on a cylindrical cavity on a non trivial mesh made of a mix of triangular and quadrangular prisms. At last, the three-dimensional explosion problem has been carried out to show the improvement gained with the use of high-order MOOD methods and the slight numerical diffusion generated by the $u2$ detection process which enables to prevent spurious numerical oscillations from appearing. We have also provided the solution computed with the PAD detection process alone to support the intrinsic positivity-preserving property of the MOOD method and measures of the CPU cost to underline that the MOOD method is effective on nowadays personal computers. Finally the interaction of a shock wave on a quarter of cone with the 4^{th}-order MOOD method proves that the MOOD method provides a very good reproduction of the physics on a unstructured non-regular 3D mesh of 1.1 millions of tetrahedra.

In a near future, we plan to adapt the MOOD within an ADER technique to avoid the multiple time steps of the Runge-Kutta approach and overcome the third-order accuracy restriction. Furthermore although the MOOD method significantly reduces the necessary computational resources (CPU and memory storage), a parallelized version is of crucial importance to treat huge size simulations. Finally the application of the MOOD method to more complex physics (multi-material, multi-phase, etc.) is also an important challenge that has to be tackled.

REFERENCES

1. R. Abgrall On Essentially Non-oscillatory Schemes on Unstructured Meshes: Analysis and Implementation, J. Comput. Phys. 114 (1994) 45–58.
2. R. Abgrall, Essentially non-oscillatory Residual Distribution schemes for hyperbolic problems, J. Comput. Phys. 214 (2006) 773–808.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

3. T. Buffard, S. Clain, Monoslope and Multislope MUSCL Methods for unstructured meshes, J. Comput. Phys. 229 (2010) 3745–3776.
4. S. Clain, V. Clauzon, L^∞ stability of the MUSCL methods, Numer. Math. 116 (2010) 31–64.
5. S. Clain, S. Diot, R. Loubère, A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD), J. Comput. Phys. 230 (2011) 4028–4050.
6. B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework, Math. Comp. 52 (1989), 411–435.
7. B. Cockburn, S. Y. Lin and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems, J. Comput. Phys. 84 (1989) 90–113.
8. B. Cockburn, S. Hou and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case, Math. Comp. 54 (1990) 545–581.
9. B. Cockburn, C.-W. Shu, The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems J. Comput. Phys. 141 (1998) 199–224.
10. P. Colella and P.R. Woodward, The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations, J. Comput. Phys. 54 (1984) 174–201.
11. S. Diot, S. Clain, R. Loubère, Improved Detection Criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials, Comput. & Fluids 64 (2012) 43–63.
12. M. Dumbser, M. Castro, C. Parés, E. F. Toro, ADER schemes on unstructured meshes for nonconservative hyperbolic systems: Applications to geophysical flows, Comput. & Fluids 38 (2009) 1731–1748.
13. M. Dumbser and M. Käser. Arbitrary High Order Non-Oscillatory Finite Volume Schemes on Unstructured Meshes for Linear Hyperbolic Systems, J. Comput. Phys. 221 (2007) 693–723.
14. M. Dumbser, M. Käser, V. A. Titarev and E. F. Toro, Quadrature-Free Non-Oscillatory Finite Volume Schemes on Unstructured Meshes for Nonlinear Hyperbolic Systems, J. Comput. Phys. 226 (2007) 204–243.
15. A. Harten, S. Osher, Uniformly highorder accurate nonoscillatory schemes I, SIAM J. Num. Anal. 24 (1987) 279–309.
16. A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly highorder accurate nonoscillatory schemes III, J. Comput. Phys. 71 (1987) 279-309.
17. C. Hu, C.W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.
18. M. E. Hubbard, Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids, J. Comput. Phys. 155 (1999) 54–74.
19. G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
20. G.-S. Jiang, E. Tadmor, Non-oscillatory central schemes for multidimensional hyperbolic conservative laws, SIAM J. Sci. Comput. 19 (1998) 1892–1917.
21. V. P. Kolgan, Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics, Transactions of the Central Aerohydrodynamics Institute 3 (1972) 68–77, in Russian.
22. V. P. Kolgan, Finite-difference schemes for computation of three dimensional solutions of gas dynamics and calculation of a flow over a body under an angle of attack, Transactions of the Central Aerohydrodynamics Institute 6 (1975) 1–6, in Russian.
23. V. P. Kolgan, Application of the principle of minimizing the derivative to the construction of finite-difference schemes fr computing discontinuous solutions of gas dynamics, J. Comput. Phys. 230 (2010) 2384–2390.
24. P. D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Comm. Pure and Appl. Math., (1954), 7, 159-193.
25. R. Loubère, P.-H. Maire, P. Vachal, Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme, Communication in Computational Physics, 2011, Vol. 10 (2011), Issue 4, pp. 940-978.
26. J. S. Park,S.-H. Yoon, C. Kim, Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids, J. Comput. Phys. 229 (2010) 788–812.
27. C.-W. Shu, High-order Weighted Essentially Non-Oscillatory schemes for convection dominated problems, SIAM Review 51 (2009) 82–126.
28. C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperin Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics 1697 Springer (1998) 325–432.
29. C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing scheme, J. Comput. Phys. 77 (1988) 439–471.
30. B.W. Skews, H. Kleine, Flow features resulting from shock wave impact on a cylindrical cavity, J. Fluid. Mech. 580 (2007) 481–493.
31. G. A. Sod, A survey of several finite difference methods for systems of non-linear hyperbolic conservation laws, J. Comput. Phys. 27 (1978) 1–31.
32. V.A. Titarev, E. F. Toro, ADER schemes for three-dimensional non-linear hyperbolic systems, J. Comput. Phys. 204 (2005) 715–736.
33. E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, 3^{rd} revision, Springer-Verlag Berlin and Heidelberg GmbH & Co. K (2009).
34. E. F. Toro, A. Hidalgo, ADER finite volume schemes for nonlinear reaction-diffusion equations, Applied Numerical Mathematics 59 (2009) 73–100.
35. B. Van Leer, Towards the ultimate conservative difference scheme I. The quest of monotonicity, Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Lecture Notes in Physics 18 (1973) 163–168.
36. B. Van Leer, Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme, J. Comput. Phys. 14 (1974) 361–370.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

37. P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.
38. H. C. Yee, M. Vinokur, M. J. Djomehri, Entropy Splitting and Numerical Dissipation, J. Comput. Phys. 162 (2000) 33–81.
39. Y.-T. Zhang and C.-W. Shu Third-order WENO scheme on three dimensional tetrahedral meshes, Com. Comput. Phys. 5 (2009) 836–848.
40. X. Zhang and C.-W. Shu Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments, Proc. R. Soc. A 467 (2011) 2752-2776.

Copyright © 2012 John Wiley & Sons, Ltd.
Prepared using fldauth.cls
Int. J. Numer. Meth. Fluids (2012)
DOI: 10.1002/fld

3.2 Ultra efficient 3D kinetic scheme

Recently with my colleague Giacomo Dimarco (IMT) we have designed an ultra efficient 3D kinetic scheme [39]. The main idea has been developed by Giacomo and my role has been reduced to implement the 3D version of this idea in an efficient numerical simulation code based on the straighforward collision operator, i.e. the BGK (Bhatnagar-Gross-Krook) relaxation operator. The purpose of this work is to show that simulating kinetic equations in seven dimensions ($\mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}$ respectively for 3D in space, 3D in velocity and 1D in time) is feasible with nowadays laptop with our new approach.

3.2.1 Quick refresher on the context

The kinetic equations provide a mesoscopic description of gases and more generally of particle systems. In many applications, the correct physical solution for a system far from thermodynamical equilibrium, such as rarefied gases or plasmas, requires the resolution of a kinetic equation [228]. However, the numerical simulation of these equations with deterministic techniques presents several drawbacks due to the large dimension of the problem. The distribution function depends on seven independent variables : three coordinates in physical space, three coordinates in velocity space and the time leading to the seven dimensions already mentioned. This "curse of dimentionality" is often used as a blanket excuse for not dealing with high-dimensions. This has led the researchers to find solution to avoid the use of the seven dimensions or, at least, to reduce the burden of dealing with them. Probabilistic techniques such as Direct Simulation Monte Carlo (DSMC) methods [229, 230, 231, 232] are extensively used in real situations due to their flexibility and low computational cost compared to finite volume, finite difference or spectral methods for kinetic equations [233, 234, 235, 236, 237]. On the other hand, DSMC solutions are affected by large fluctuations. Moreover, in non stationary situations it is impossible to use time averages to reduce these fluctuations and this leads to, either poorly accurate solutions, or again to computationally expensive simulations. For this reason, many different works have been dedicated to reduce some of the disadvantages of Monte Carlo methods, see as instance [230] for an overview on efficient and low variance Monte Carlo methods.
In this work, we consider the development of a new deterministic method to solve kinetic equations. The key point is an efficient discretization of the linear transport part of these equations. The proposed method is based on the so-called discrete velocity models (DVM) [235] and on the semi Lagrangian approach [238, 239]. The DVM models are obtained by discretizing the velocity space into a set of fixed discrete velocities [240, 235, 236, 241]. As a result of this discretization, the original kinetic equation is then represented as a set of linear transport equations plus an interaction term which couples all the equations. In order to solve the resulting set of equations, the most common strategy consists in an operator splitting strategy [242] : the solution in one time step is obtained by the sequence of two stages. First one integrates the space homogeneous equations and then, in the second stage, the transport equation using the output of the previous step as initial condition. More sophisticated splitting techniques can be employed, which permits to obtain high order in time discretizations of the kinetic equations as for instance the Strang splitting method [243]. In any case, the resulting method is very simple and robust but the main drawback is again the excessive computational cost. It is a matter of fact that the numerical solution through such microscopic models and deterministic schemes remains nowadays too expensive especially in multi-dimensions even with the use of super-computers.

3.2.2 Ultra Fast-Kinetic-Scheme (FKS)

To overcome this problem, we propose to use a Lagrangian technique which exactly solves the transport stage on the entire domain and then to project the solution on a grid to compute the contribution of the collision operator. The resulting scheme shares many analogies with semi-Lagrangian methods [238, 239, 234] and with Monte Carlo schemes [244], but on the contrary to them, the method is as fast as a particle method while the numerical solution remains fully deterministic, which means that there is no source of statistical error.
The main features of the method proposed in this work can be summarized as follows :

- The BGK equation is discretized in velocity space by using the discrete velocity models (DVM) method. The principle of Discrete Velocity Model (DVM) [235] is to set a grid in the velocity space and to transform the kinetic equation in as set of N linear hyperbolic evolution equations with source terms.
- A time splitting procedure is employed between the transport and the relaxation operators for each of the N evolution equations. First- and second-order Strang time splittings [243] are considered.
- The transport part is solved exactly, which means without using a spatial mesh. The initial data of this step is given by the solution of the relaxation operator.
- The relaxation part is solved on the grid. The initial data for this step is given by the value of the distribution function in the center of the cells after the transport step.

We refer the reader to [39] for the details and we only describe the key points of our fast algorithm. The algorithm relies on a very efficient transport step performed on a logical rectangular grid. First the 1D velocity bounded space $\mathcal{U} = [u_{min}, u_{max}]$ is represented by N_v particles uniformly distributed

$$u_p = u_{min} + (p-1)\Delta u + \Delta u/2, \tag{3.1}$$

with $\Delta u = (u_{max} - u_{min})/N_v$. The same process is made for v and w components of the velocity space leading to $N_p \times N_p \times N_p$ particles which pave $\mathcal{U} \times \mathcal{V} \times \mathcal{W}$. Because a generic particle moves with constant velocity $\boldsymbol{U}_p = (u_p, v_p, w_p)$ the transport step consists of solving the N_v^3 equations indexed by p with

$$\widetilde{\boldsymbol{X}}_p^{n+1} = \boldsymbol{X}_p^n + \Delta t\, \boldsymbol{U}_p \qquad \forall p = 1, \cdots, N_v^3. \tag{3.2}$$

The first key point in our approach is to work with a regular spacial mesh made of $N_i \times N_j \times N_k$ cells, all cells being the same. Each cell is indexed with three indexes i, j, k for each spacial direction. N_v^3 particles are localized and further will evolve within each spacial cell. Each particle carries its own moments, mass m_p, momentum $m_p \boldsymbol{U}_p$ and energy $\frac{1}{2} m_p \|\boldsymbol{U}_p\|^2$. The mass is computed *via* the distribution fonction feeded with the macroscopic state of the current cell.
The second key point is to set the initial position of all N_v^3 particles at the cell center of their associated cell $\boldsymbol{X}_{i,j,k}$, that is to say

$$\boldsymbol{X}_p^0 = \boldsymbol{X}_{i,j,k}, \qquad \forall p = 1, \cdots, N_v^3. \tag{3.3}$$

After the transport step (3.2) of particle p in cell $\Omega_{i,j,k}$ either $\widetilde{\boldsymbol{X}}_p^{n+1}$ remains in cell $\Omega_{i,j,k}$ or it lands in a neighbor cell Ω_{i_l,j_l,k_l} where $i_l = i + a$ with $a = -1, 0$, or $+1$ (idem for $j_l = j + b$ and $k_l = k + c$). If particle p remains within its cell then the cell moments are not modified. Contrarily if particle p leaves its cell then the cell $\Omega_{i,j,k}$ moments are decreased by particle p's moments whereas the cell Ω_{i_l,j_l,k_l} moments are increased. The new moments in each cell are therefore decreased due to the leaving particles and increased by the incoming particles, see Figure 3.3-left. By construction

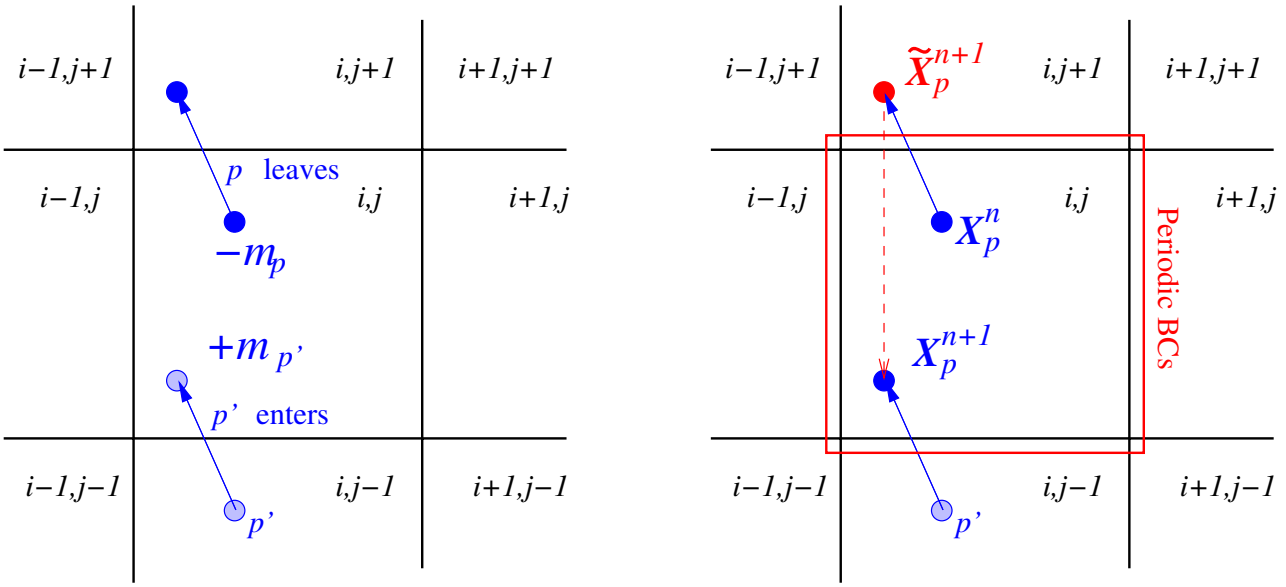


Figure 3.3 – *Sketch of the 2D transport step in the fast kinetic scheme — Left : a particle p leaves the current cell i, j with its mass m_p (the cell moments are therefore decreased) and it lands in cell $i, j+1$ (it contributes to the cell moments). If p leaves the cell then a sister particle p' enters and contributes to the moments in cell i, j — Right : the same situation for the particle positions. The true new position $\boldsymbol{X}_p^{n+1}$ is computed as if periodic boundary conditions are applied to cell i, j.*

conservation of moments is ensured.

Due to the initialization of particle positions the fact that p leaves its cell implies that a "sister" particle p' on the other side of the cell is entering, see Figure 3.3-right. This "sister" particle has a position $\widetilde{\boldsymbol{X}}_{p'}^{n+1}$ in $\Omega_{i,j,k}$ which is the same as $\widetilde{\boldsymbol{X}}_{p'}^{n+1}$ in its new cell Ω_{i_l,j_l,k_l}. It is then easy to see that this situation is equivalent to assume periodic boundary conditions on each cell. Therefore the new particle p position in cell $\Omega_{i,j,k}$ is

$$\boldsymbol{X}_p^{n+1} = \boldsymbol{X}_p^n + \Delta t\, \boldsymbol{U}_p \qquad \text{subject to periodic BCs on} \quad \Omega_{i,j,k}. \tag{3.4}$$

Doing so only the positions of particles need to be updated for only one spacial cell. In addition only one set of N_v^3 particle positions need to be stored, which drastically reduces the memory consumption of the method. The information which must be kept in the case particle p leaves the cell, is the integer vector (a, b, c) which determines in which cell particle p lands. Finally the algorithm simply consists of transporting the particles and marking the particles leaving their cell, computing the moments of the leaving particles, update the cell moments, compute the moments of the incoming particles and re-update the cell moments.

Thanks to this approach we are able to compute the solution of the full six dimensional kinetic equation on a laptop. This is, up to our knowledge, the first time that the full kinetic equation has been solved with a deterministic scheme on a single processor machine for acceptable mesh sizes and in a reasonable amount of time (around ten hours for 100^3 space $\times$ 12^3 velocity space mesh points).

3.2.3 Numerical experiments in 3D/3D

Here we report some simulations of the full 3D/3D problem [3]. As already mentioned the goal is to numerically show that such a kinetic scheme can reasonably perform on six dimensions on a mono-processor laptop. All simulations have been carried out on a HP EliteBook 8740W Intel(R) Core(TM) i7 Q840@1.87GHz running under a Ubuntu (oneiric) version 11.10. The code has been compiled with gfortran 4.6 compiler with -O3 optimization flags.
The 3D Sod shock tube has been run with the 3D/3D FKS method. The left state of the 1D Sod problem is set for any cell c with cell center radius $r_c \leq 1/2$, conversely the right state is set for cell radius $r_c > 1/2$. The final time is $t_{\text{final}} = 0.1$. The domain is the unit cube and the mesh is composed of $N_x \times N_x \times N_x$ cells with $\Delta x = 1/N_x$ and $\Delta x = \Delta y = \Delta z$. The problem is run with $N_x = 50$ (125000 cells), $N_x = 100$ (1 million cells) and $N_x = 200$ (8 millions cells). The velocity space is either $[-10;10]$ discretized with 12^3 points, or $[-15;15]$ discretized with 13^3 points. This leads to consider up to $200^3 \times 13^3 \simeq 17.7$ milliards of particles. The time step is fixed to 95% of the maximum time step allowed, as prescribed by the CFL condition, apart from the last time step. Symmetric boundary conditions are considered. In Figure 3.4 the density is plotted as a function of the radius (left panel) and the colored density on a 3D view (right panel) for $N_x = 50$ (middle panels) and $N_x = 200$ (bottom panels). The two different choices for the bounds and the mesh points in velocity space do not significantly change the results hence only the solution with bounds $[-10;10]$ and with 12^3 mesh points is reported. The reference solution is obtained with ALE INC(ubator) code [5] with 1000 cells in radial and 20 cells in angular directions. Moreover in Figure 3.4 (top panel) we present the convergence of the density as a function of cell center radius for all cells for the $50 \times 50 \times 50$, $100 \times 100 \times 100$ and $200 \times 200 \times 200$ cells meshes. These curves are compared to the reference solution in straight thick line and they show that the results are converging towards the reference solution. In table 3.1 we gather the number of time steps and the total CPU time T for 50^3 and 100^3 cell meshes for the two different configurations : one with $N_v = 13$ and the velocity space $[-15,15]$ and the second one with $N_v = 12$ and the velocity space $[-10,10]$. For the 50^3 mesh the simulation takes 45 minutes or 1.36 hour depending on the configuration. For the finer 100^3 mesh the simulation takes either 11 hours or 24 hours The memory consumption ranges from 124Mb to 924Mb depending on the configurations and it scales with the number of cells N_c.
Then, we compute the cost per cycle T_{cycle} and per cycle per cell T_{cell}. One observe that the cost per cycle per cell is an almost constant equal to 4×10^{-4}s or 5.5×10^{-4}s. The extrapolation of the CPU time T for a 200^3 mesh at T_{cell} fixed leads to one or two weeks computation for the two configurations and a memory storage of about 900MB. In Figure 3.5 we plot the CPU time (red or blue symbols for each configuration and mesh points of the velocity space) and the extrapolation curves $CPU(N_x, N_c, T_{\text{cell}}) = \frac{N_{\text{cycle}}}{N_x} N_c T_{\text{cell}}$ for the 3D Sod problem up to time $t_{\text{final}} = 0.1$ for single processor laptop computation on a fixed mesh in velocity space of $N_v = 12^3$ points. We deduced that the FKS method can be used at most on a single processor machine up to a $200 \times 200 \times 200$ cells for roughly one week of computation. One also notices that the CPU time linearly scales on a log/log graph as expected (right panel of Figure 3.5)
In the future we would like to extend the method to non uniform meshes, more advanced boundary conditions and different discretization of the velocity space. One expects with this last point to increase the accuracy of the schemes without losing its attractive efficiency. To avoid the loss of accuracy close to the fluid limit, we want to couple the FKS method to an high order solver for the

3. We consider the case in which the projection is made towards the equilibrium at each time step. We recall that, in this regime, the numerical method gives the worst results in terms of precision, on the other hand, exact solution are known and this permits to make fair comparisons.

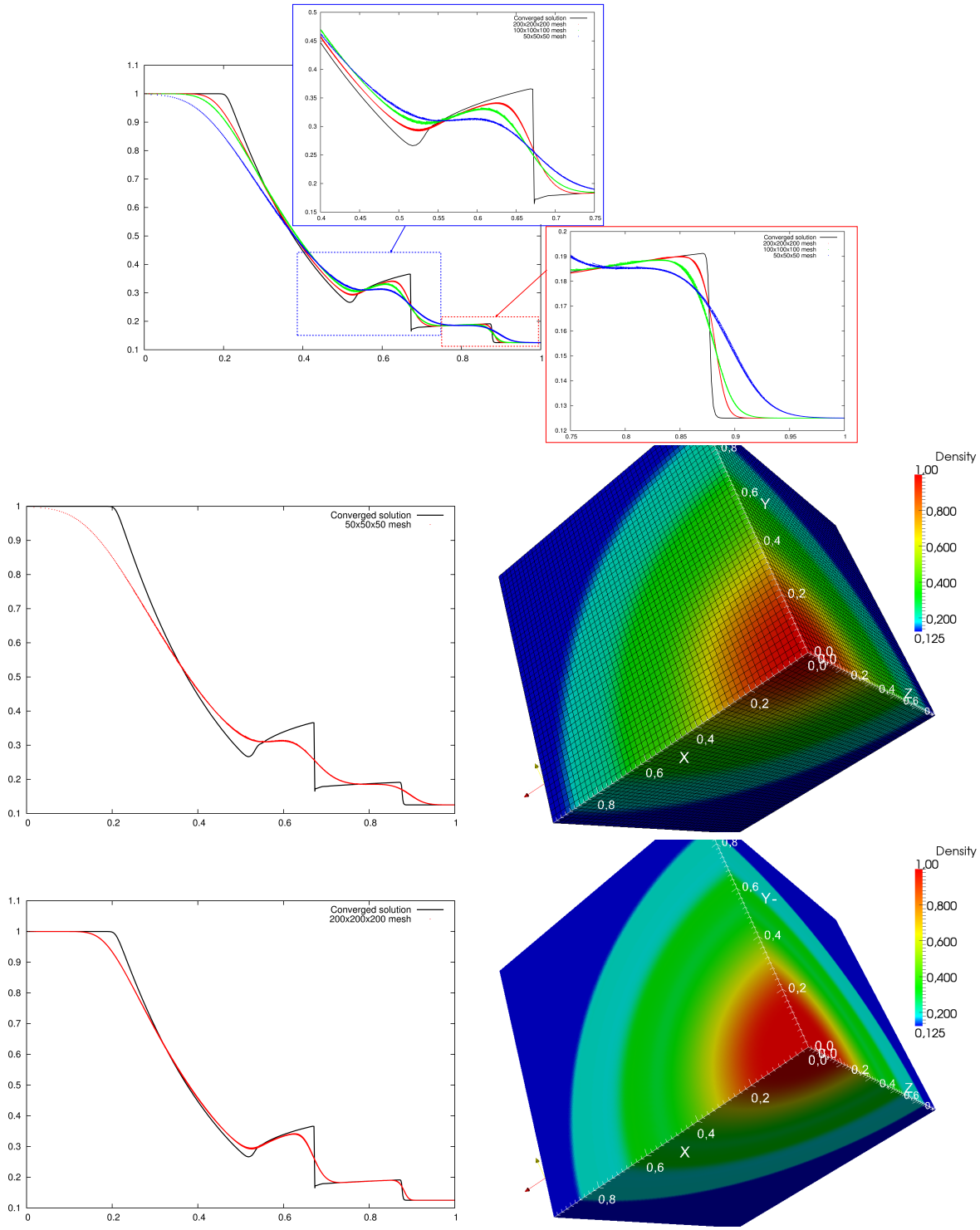


Figure 3.4 – *Sod problem at* $t_{final} = 0.1$ *for* $N_x \times N_x \times N_x$ *cells (for* $N_x = 50, 100, 200$*) for the velocity space* $[-10; 10]$ *discretized with* 12^3 *mesh points. — Top : Convergence of density as a function of cell center radius for all cells vs converged solution (straight thick line) for the three meshes with zooms on contact and shock waves. Left : Density as a function of cell center radius (middle :* $N_x = 50$*, bottom :* $N_x = 200$*) Right : 3D view of density on the unit cube* $N_x = 50$ *(middle) and* $N_x = 200$ *(bottom) (the mesh is only shown for* $N_x = 50$*).*

N_v^3 Bnds	**Cell #** $N_c \times N_v^3$ $N_x \times N_y \times N_z \times N_v^3$	**Cycle** N_{cycle}	**Time** T (s)	**Time/cycle** T_{cycle} (s)	**Time/cell** T_{cell} (s)	**Mem** (MB)
13^3 ± 15	$25^3 \times 13^3$ $= 3.4328125 \times 10^6$	32	346s (5.76mn)	10.81	6.92×10^{-4}	2.4
	$50^3 \times 13^3$ $= 274.625000 \times 10^6$	81	4900s (1.36h)	60.50	4.84×10^{-4}	15.5
	100×13^3 $= 2.1970 \times 10^9$	160	85720s (23.8h)	535.75	5.36×10^{-4}	115.5
extrapol.	200×13^3 $= 1.7576 \times 10^{10}$	320	$\sim 1.4 \times 10^6$s (16d)	~ 4400	5.5×10^{-4}	~ 900
12^3 ± 10	$25^3 \times 12^3$ $= 27 \times 10^6$	27	218s (3.63mn)	8.07	5.17×10^{-4}	2.3
	$50^3 \times 12^3$ $= 125 \times 10^3$	54	2702s (45mn)	50.03	4.00×10^{-4}	15.4
	$100^3 \times 12^3$ $= 1.728 \times 10^9$	107	38069s (10.57h)	355.79	3.56×10^{-4}	115.4
extrapol.	$200^3 \times 12^3$ $= 1.3284 \times 10^{10}$	214	~ 633440s (7d)	~ 2960	3.7×10^{-4}	~ 900

TABLE 3.1 – *3D Sod shock tube. The time per cycle is obtained by $T_{cycle} = T/N_{cycle}$ and the time per cycle per cell by $T_{cell} = T/N_{cycle}/N_c$. The lines marked with extrapol. have been extrapolated by fixing N_c, N_{cycle} and T_{cell}.*

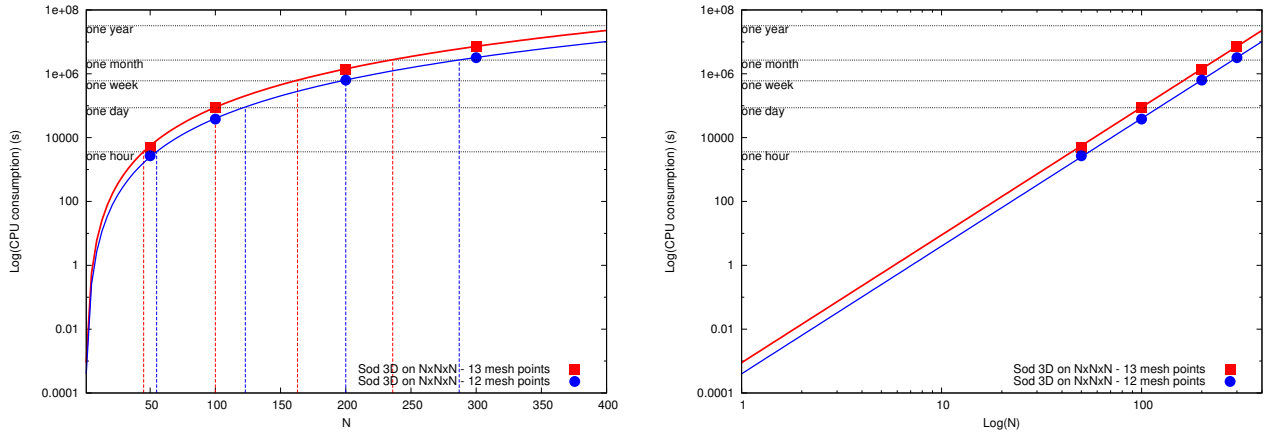


FIGURE 3.5 – *Left : Log of the CPU time consumption for the 3D Sod problem at $t_{final} = 0.1$ as a function of N (for $N \times N \times N$ cell meshes) on a single processor laptop The red/blue squares are taken from Table 3.1, the thick red/blue curves are the extrapolation curve from T_{cell}. The horizontal lines corresponding to one hour, one day, week, month and year are also plotted. $N = 100$ corresponds to the 'one million cells' in space — Right : Log/Log scale.*

system of equations which describes the fluid limit. Finally, we want to extend the method to other kinetic equations as the Boltzmann or the Vlasov equation and uses GPU infrastructure to speed-up such computation even more.

Towards an ultra efficient kinetic scheme Part I: basics on the BGK equation *

Giacomo Dimarco† Raphaël Loubere‡

April 13, 2012

Abstract

In this paper we present a new ultra efficient numerical method for solving kinetic equations. In this preliminary work, we present the scheme in the case of the BGK relaxation operator. The scheme, being based on a splitting technique between transport and collision, can be easily extended to other collisional operators as the Boltzmann collision integral or to other kinetic equations such as the Vlasov equation. The key idea, on which the method relies, is to solve the collision part on a grid and then to solve exactly the transport linear part by following the characteristics backward in time. The main difference between the method proposed and semi-Lagrangian methods is that here we do not need to reconstruct the distribution function at each time step. This allows to tremendously reduce the computational cost of the method and it permits for the first time, to the author's knowledge, to compute solutions of full six dimensional kinetic equations on a single processor laptop machine. Numerical examples, up to the full three dimensional case, are presented which validate the method and assess its efficiency in 1D, 2D and 3D.

Keywords: Kinetic equations, discrete velocity models, semi Lagrangian schemes, Boltzmann-BGK equation, 3D simulation.

1 Introduction

The kinetic equations provide a mesoscopic description of gases and more generally of particle systems. In many applications, the correct physical solution for a system far from thermodynamical equilibrium, such as rarefied gases or plasmas, requires the resolution of a kinetic equation [7]. However, the numerical simulation of these equations with deterministic techniques presents several drawbacks due to the large dimension of the problem.

*This work was supported by the ANR Blanc project BOOST and the ANR JCJC project ALE INC(ubator) 3D

†Université de Toulouse; UPS, INSA, UT1, UTM; CNRS, UMR 5219; Institut de Mathématiques de Toulouse; F-31062 Toulouse, France. (giacomo.dimarco@math.univ-toulouse.fr).

‡Université de Toulouse; UPS, INSA, UT1, UTM; CNRS, UMR 5219; Institut de Mathématiques de Toulouse; F-31062 Toulouse, France. (raphael.loubere@math.univ-toulouse.fr).

The distribution function depends on seven independent variables: three coordinates in physical space, three coordinates in velocity space and the time. As a consequence, probabilistic techniques such as Direct Simulation Monte Carlo (DSMC) methods [1, 5, 6, 29] are extensively used in real situations due to their large flexibility and low computational cost compared to finite volume, finite difference or spectral methods for kinetic equations [17, 18, 28, 31, 35]. On the other hand, DSMC solutions are affected by large fluctuations. Moreover, in non stationary situations it is impossible to use time averages to reduce these fluctuations and this leads to, either poorly accurate solutions, or again to computationally expensive simulations.

For this reason, many different works have been dedicated to reduce some of the disadvantages of Monte Carlo methods. We quote [5] for an overview on efficient and low variance Monte Carlo methods. For applications of variance reduction techniques to kinetic equation let us remind to the works of Homolle and Hadjiconstantinou [21] and [22]. We mention also the work of Boyd and Burt [4] and of Pullin [36] who developed a low diffusion particle method for simulating compressible inviscid flows. We finally quote the works of Dimarco and Pareschi [14, 15] and of Degond, Dimarco and Pareschi [12] who constructed efficient and low variance methods for kinetic equations in transitional and general regimes.

In this work, we consider the development of a new deterministic method to solve kinetic equations. In particular, we focus on the development of efficient techniques for the discretization of the linear transport part of these equations. The proposed method is based on the so-called discrete velocity models (DVM) [28] and on the semi Lagrangian approach [8, 9]. The DVM models are obtained by discretizing the velocity space into a set of fixed discrete velocities [3, 28, 31, 32]. As a result of this discretization, the original kinetic equation is then represented as a set of linear transport equations plus an interaction term which couples all the equations. In order to solve the resulting set of equations, the most common strategy consists in an operator splitting strategy [10]: The solution in one time step is obtained by the sequence of two stages. First one integrates the space homogeneous equations and then, in the second stage, the transport equation using the output of the previous step as initial condition. More sophisticated splitting techniques can be employed, which permits to obtain high order in time discretizations of the kinetic equations as for instance the Strang splitting method [37]. In any case, the resulting method is very simple and robust but the main drawback is again the excessive computational cost. It is a matter of fact that the numerical solution through such microscopic models and deterministic schemes remains nowadays too expensive especially in multi-dimensions even with the use of super-computers.

To overcome this problem, we propose to use a Lagrangian technique which exactly solves the transport stage on the entire domain and then to project the solution on a grid to compute the contribution of the collision operator. The resulting scheme shares many analogies with semi-Lagrangian methods [8, 9, 18] and with Monte Carlo schemes [24], as we will explain, but on the contrary to them, the method is as fast as a particle method while the numerical solution remains fully deterministic, which means that there is no source of statistical error. Thanks to this approach we are able to compute the solution of the full six dimensional kinetic equation on a laptop. This is, up to our knowledge, the

first time that the full kinetic equation has been solved with a deterministic scheme on a single processor machine for acceptable mesh sizes and in a reasonable amount of time (around ten hours for 100^3 space $\times$ 12^3 velocity space mesh points).

In this first work, we consider a simple collision operator, *i.e.* the BGK (Bhatnagar-Gross-Krook) relaxation operator [20]. The extension of the method to other operators like the Boltzmann one [1, 7] or to other kinetic equations like the Vlasov equation [2, 18] will be considered in future works. At the present moment, the method is designed to work on uniform grids, although extensions to other meshes are possible and will be also considered in the next future.

The article is organized as follows. In section 2, we introduce the Boltzmann-BGK equations and their properties. In section 3, we present the discrete velocity model (DVM). Then in section 4 we present the numerical scheme. Section 5 is devoted to the illustration of the analogies between such fast kinetic scheme (FKS) and particle methods. Several test problems up to three dimensional test cases which demonstrate the capabilities and the strong efficiency of the method are presented and discussed in section 6. Some final considerations and future developments are finally drawn in the last section.

2 Boltzmann-BGK Equation

We consider the following kinetic equation as a prototype model for developing our method:

$$\partial_t f + v \cdot \nabla_x f = \frac{1}{\tau}(M_f - f), \tag{1}$$

with the initial condition

$$f(x, v, t = 0) = f_0(x, v). \tag{2}$$

This is the Boltzmann-BGK equation where $f = f(x, v, t)$ is a non negative function describing the time evolution of the distribution of particles which move with velocity $v \in \mathbb{R}^d$ in the position $x \in \Omega \subset \mathbb{R}^d$ at time $t > 0$. For simplicity we consider the same dimension in space and in velocity space d, however it is possible to consider different dimensions in order to obtain different simplified models. In the BGK equation the collisions are modeled by a relaxation towards the local thermodynamical equilibrium defined by the Maxwellian distribution function M_f. The local Maxwellian function is defined by

$$M_f = M_f[\rho, u, T](v) = \frac{\rho}{(2\pi\theta)^{d/2}} \exp\left(\frac{-|u - v|^2}{2\theta}\right), \tag{3}$$

where $\rho \in \mathbb{R}^*$ and $u \in \mathbb{R}^d$ are the density and mean velocity while $\theta = RT$ with T the temperature of the gas and R the gas constant. The macroscopic values ρ,u and T are related to f by:

$$\rho = \int_{\mathbb{R}^d} f dv, \qquad u = \int_{\mathbb{R}^d} v f dv, \qquad \theta = \frac{1}{\rho d} \int_{\mathbb{R}^d} |v - u|^2 f dv. \tag{4}$$

The energy E is defined by

$$E = \frac{1}{2} \int_{\mathbb{R}^d} |v|^2 f dv = \frac{1}{2}\rho|u|^2 + \frac{d}{2}\rho\theta, \tag{5}$$

while the kinetic entropy of f by

$$H(f) = \int_{\mathbb{R}^d} f \log f dv. \tag{6}$$

The parameter $\tau > 0$ in (1) is the relaxation time. In this paper, τ is fixed at the beginning of each numerical test. Considering relaxation frequencies as functions of the macroscopic quantities does not change the numerical scheme we will propose and its behaviors. We refer to section 6 for the numerical values chosen.

If we consider the BGK equation (1) multiplied by 1, v, $\frac{1}{2}|v^2|$ (the so-called collision invariants), and then integrated with respect to v, we obtain the following balance laws:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \nabla_x \cdot (\rho u) = 0, \\ &\frac{\partial \rho u}{\partial t} + \nabla_x \cdot (\rho u \otimes u + P) = 0, \\ &\frac{\partial E}{\partial t} + \nabla_x \cdot (Eu + Pu + q) = 0, \end{aligned} \tag{7}$$

which express the conservation of mass, momentum and total energy, in which $P = \int_{\mathbb{R}^d} (v - u) \otimes (v - u) f dv$ is the pressure tensor while $q = \int_{\mathbb{R}^d} \frac{1}{2}(v - u)|v - u|^2 dv$ is the heat flux. Furthermore the following inequality expresses the dissipation of entropy:

$$\partial_t \left(\int_{\mathbb{R}^d} f \log f \, dv \right) + \nabla_x \cdot \left(\int_{\mathbb{R}^d} v f \log f \, dv \right) \leq 0. \tag{8}$$

System (7) is not closed, since it involves other moments of the distribution function than just ρ, ρu and E. Let us describe one way to close the system.

The Maxwellian M_f can be characterized as the unique solution of the following entropy minimization problem

$$H(M_f) = \min \left\{ H(f), \ f \geq 0 \ s.t. \int_{\mathbb{R}^d} m f \, dv = U \right\} \tag{9}$$

where m and U are the vectors of the collision invariants and of the first three moments of f respectively:

$$m(v) = \left(1, v, \frac{1}{2}|v|^2\right), \ U = (\rho, \rho u, E). \tag{10}$$

This is the well-known local Gibbs principle, and it expresses that the local thermodynamical equilibrium state minimizes the entropy, in the mathematical sense, of all the possible states subject to the constraint that moments U are prescribed.

Formally, when the number of collision goes to infinity, which means $\tau \to 0$, the function f converges towards the Maxwellian distribution. In this limit, it is possible to compute the moments P and q of f in terms of ρ, ρu and E. In this way, one can close the system of balance laws (7) and get the so-called Euler system of compressible gas

dynamics equations

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla_x \cdot (\rho u) = 0,\\
&\frac{\partial \rho u}{\partial t} + \nabla_x \cdot (\rho u \otimes u + pI) = 0,\\
&\frac{\partial E}{\partial t} + \nabla_x \cdot ((E+p)u) = 0,\\
&p = \rho\theta, \quad E = \frac{d}{2}\rho\theta + \frac{1}{2}\rho|u|^2.
\end{aligned} \tag{11}$$

3 The Discrete Velocity Model (DVM)

The principle of Discrete Velocity Model (DVM) is to set a grid in the velocity space and to transform the kinetic equation in a set of linear hyperbolic equations with source terms. We refer to the work of Mieussens [28] for the description of this model and we remind to it for the details.

Let $\mathcal{K}$ be a set of N multi-indices of $\mathbb{N}^d$, defined by $\mathcal{K} = \left\{k = (k^{(i)})_{i=1}^d,\ k^{(i)} \le K^{(i)}\right\}$, where $\{K^{(i)}\}$ are some given bounds. We introduce a Cartesian grid $\mathcal{V}$ of $\mathbb{R}^d$ by

$$\mathcal{V} = \{v_k = k\Delta v + a,\ k \in \mathcal{K}\}, \tag{12}$$

where a is an arbitrary vector of $\mathbb{R}^d$ and Δv is a scalar which represents the grid step in the velocity space. We denote the discrete collision invariants on $\mathcal{V}$ by $m_k = (1, v_k, \frac{1}{2}|v_k|^2)$.

Now, in this setting, the continuous distribution function f is replaced by a $N-$vector $f_{\mathcal{K}}(x,t)$, where each component is assumed to be an approximation of the distribution function f at location v_k:

$$f_{\mathcal{K}}(x,t) = (f_k(x,t))_k, \qquad f_k(x,t) \approx f(x, v_k, t). \tag{13}$$

The fluid quantities are then obtained from f_k thanks to discrete summations on $\mathcal{V}$:

$$U(x,t) = \sum_k m_k f_k(x,t)\,\Delta v. \tag{14}$$

The discrete velocity BGK model consists of a set of N evolution equations for f_k of the form

$$\partial_t f_k + v_k \cdot \nabla_x f_k = \frac{1}{\tau}(\mathcal{E}_k[U] - f_k), \tag{15}$$

where $\mathcal{E}_k[U]$ is a suitable approximation of M_f defined next. Two strongly connected and important questions arise when dealing with discrete velocity models. The first one is about the truncation and boundedness of the velocity space. The second one concerns the conservation of macroscopic quantities.

Truncation and boundedness of the velocity space. In DVM methods one needs to truncate the velocity space and to fix some bounds. This gives the number N of evolution equations (15). Of course, the number N is chosen as a compromise between the desired precision in the discretization of the velocity space and the computational cost, while the bounds are chosen to give a correct representation of the flow. Observe in fact that, the macroscopic velocity and temperature are bounded above by velocity bounds. This implies that the discrete velocity set must be large enough to take into account large variations of the macroscopic quantities which may appear as a result of the time evolution of the equations. Moreover, as a consequence of the velocity discretization, we have that the temperature is bounded from below. We summarize the above remarks by the following statement. Let f be a non negative distribution function, then the macroscopic velocity and temperature associated to f in $\mathcal{V}$ by

$$u = \frac{1}{\rho}\langle v f\rangle_{\mathcal{K}}, \qquad T = \frac{1}{dR\rho}\langle |v-u|^2 f\rangle_{\mathcal{K}}, \tag{16}$$

where $\langle . \rangle_{\mathcal{K}}$ denotes the summation over the set of multi-indices $\mathcal{K}$, satisfy the bounds [28]

$$\min_{\mathcal{K}} v_k^{(i)} \le \ u^{(i)} \ \le \max_{\mathcal{K}} v_k^{(i)}, \ \forall i = 1, \dots, d \tag{17}$$

$$\frac{1}{dR}\min_{\mathcal{K}} |v-u|^2 \le \quad T \quad \le \frac{1}{dR}\max_{\mathcal{K}} |v-u|^2. \tag{18}$$

Conservation of macroscopic quantities. Exact conservation of macroscopic quantities is impossible, because in general the support of the distribution function is non compact. Thus, in order to conserve macroscopic variables, different strategies can be adopted, two possibilities are described in [19, 28]. Moreover, the approximation of the equilibrium distribution M_f with $\mathcal{E}_k[U]$ must be carefully chosen in order to satisfy the conservation of mass, momentum and energy. In the following section we will discuss our choices in details. Such choices prevent the lack of conservation of physical quantities.

Remark 1 *Once DVM model is defined as above, the common choice which permits to solve the kinetic equation is to discretize the N evolution equations with the preferred finite volume or finite difference method [28, 31, 32, 35]. Alternatively, one can reconstruct the distribution function in space and then follows the characteristics backward in time to obtain the solution of the linear transport equation [8, 9, 17, 18]. Our choice, described in the next section, which enables to drastically decrease the computational cost, consists of an exact solution of the linear transport equation avoiding the reconstruction of the distribution function.*

4 Fast kinetic schemes (FKS)

The main features of the method proposed in this work can be summarized as follows:

- The BGK equation is discretized in velocity space by using the DVM method.

- A time splitting procedure is employed between the transport and the relaxation operators for each of the resulting N evolution equations (15). First- and second-order Strang time splitting [37] are considered.
- The transport part is exactly solved, which means without using a spatial mesh. The initial data of this step is given by the solution of the relaxation operator.
- The relaxation part is solved on the grid. The initial data for this step is given by the value of the distribution function in the center of the cells after the transport step.

Before describing the scheme, we explain how we overcome the drawback of the lack of conservation of macroscopic quantities in DVM methods.

4.1 Conservative methods

We introduce the conservative method for the initial data and then we extend it to the scheme. The initialisation is done in two steps. First we fix

$$\widetilde{f}_k(x, t=0) = f(x, v_k, t=0),\ k = 1, \dots, N. \tag{19}$$

Observe that, in order to do this operation we do not need to discretize the physical space, in others words, if the initial data is known continuously, this information can be kept. However, for simplicity, we already at this stage introduce a Cartesian uniform grid in the physical space. This is defined by the set $\mathcal{J}$ of M multi-indices of $\mathbb{N}^d$, which is $\mathcal{J} = \{j = (j^{(i)})_{i=1}^d,\ j^{(i)} \le J^{(i)}\}$, where $\{J^{(i)}\}$ are some given bounds which represent the boundary points in the physical space. Next, the grid $\mathcal{X}$ of $\mathbb{R}^d$ is given by

$$\mathcal{X} = \{x_j = j\Delta x + b,\ j \in \mathcal{J}\}, \tag{20}$$

where d represents at the same time the dimension of the physical space and the dimension of the velocity space which are taken equal for simplicity, even if this is not necessary for the setting of the numerical method. Finally, b is a vector of $\mathbb{R}^d$ which determines the form of the domain and Δx is a scalar which represents the grid step in the physical space. We consider a third discretization which is the time discretization $t^n = n\Delta t$. We will later in the paper introduce the time step limitations.

We denote with $f_{j,k}^n$ the approximation $f_{j,k}^n \simeq f(x_j, v_k, t_n)$ and with $\widetilde{f}_{j,k}^n$ the pointwise distribution value $\widetilde{f}_{j,k}^n = f(x_j, v_k, t_n)$ which are different, for conservation reasons, as explained next. In this notation, the discrete moments of the distribution f are

$$U_j^n = \langle m_k f_{j,k}^n\, \Delta v\rangle_{\mathcal{K}}. \tag{21}$$

The corresponding discrete equilibrium is denoted $\mathcal{E}_k[U_j^n]$, or equivalently by $\mathcal{E}_{j,k}^n[U]$, which is an approximation of $M_f[U_j^n]$ and it will be also defined later. When the distribution function is truncated in velocity space, conservation of the macroscopic quantities is no longer possible. Thus, in order to restore the correct conserved variables we make use

of a simple constrained Lagrange multiplier method [19], where the constraints are mass, momentum and energy of the solution. Let us recall the technique from [19]: Let N be the total number of discretization points of the velocity space of the distribution function. We consider one space cell, the same renormalization of f should be considered for all spatial cells. Let

$$\widetilde{f} = \left(\widetilde{f}_1, \widetilde{f}_2, \dots, \widetilde{f}_N\right)^T \tag{22}$$

be the pointwise distribution vector at $t = 0$ and

$$f = (f_1, f_2, \dots, f_N)^T \tag{23}$$

be the unknown corrected distribution vector which fulfills the conservation of moments. Let

$$C_{(d+2)\times N} = \begin{pmatrix} (\Delta v)^d \\ v_k(\Delta v)^d \\ |v_k|^2(\Delta v)^d \end{pmatrix} \tag{24}$$

and $U_{(d+2)\times 1} = (\rho\ \rho u\ E)^T$ be the vector of conserved quantities. Conservation can be imposed using a constrained optimization formulation:

$$\begin{gathered}
\text{Given } \widetilde{f} \in \mathbb{R}^N,\ C \in \mathbb{R}^{(d+2)\times N},\ \text{and } U \in \mathbb{R}^{(d+2)\times 1},\\
\text{find } f \in \mathbb{R}^N \text{ such that}\\
\|\widetilde{f} - f\|_2^2 \text{ is minimized subject to the constrain } Cf = U.
\end{gathered} \tag{25}$$

To solve this constrain minimization problem, one possibility is to employ the Lagrange multiplier method. Let $\lambda \in \mathbb{R}^{d+2}$ be the Lagrange multiplier vector. Then the corresponding scalar objective function to be optimized is given by

$$L(f, \lambda) = \sum_{k=1}^N |\widetilde{f}_k - f_k|^2 + \lambda^T(Cf - U). \tag{26}$$

The above equation can be solved explicitly. In fact, taking the derivative of $L(f,\lambda)$ with respect to f_k, for all $k = 1, ..., N$ and λ_i, for all $i = 1, ..., d+2$, that is to say the gradient of L, we obtain

$$\frac{\partial L}{\partial f_k} = 0,\ \ k = 1, ..., N \implies f = \widetilde{f} + \frac{1}{2}C^T\lambda, \tag{27}$$

and

$$\frac{\partial L}{\partial \lambda_i} = 0,\ \ i = 1, ..., d+2 \implies Cf = U. \tag{28}$$

Now, solving for λ we get

$$CC^T\lambda = 2(U - C\widetilde{f}), \tag{29}$$

and observing that the matrix CC^T is symmetric and positive definite, since C is the integration matrix, one deduces that the inverse of CC^T exists. In particular the value of λ is uniquely determined by

$$\lambda = 2(CC^T)^{-1}(U - C\widetilde{f}). \tag{30}$$

Back substituting λ into (27) provides

$$f = \widetilde{f} + C^T(CC^T)^{-1}(U - C\widetilde{f}). \tag{31}$$

Observe that, following the same principle, we can impose the conservation of other macroscopic quantities, in addition to mass, momentum and energy. A key point is that, in practice, we need to solve the above minimization problem only for the initial data $f(x_j, v_k, t = 0)$ because once the conservation is guaranteed for $t = 0$, this is also guaranteed for the entire computation because the exact solution is used for solving the transport step. The only possible source of loss of conservation for the entire scheme is the relaxation step. This means that, for this step, we will need to impose conservation of the macroscopic quantities but only for the equilibrium distribution.

The discretization of the Maxwellian distribution $M_f(x, v, t)$, should satisfy the same properties of conservation of the distribution f, *i.e.* $U_j^n = \langle m_k f_{j,k}^n \Delta v\rangle_{\mathcal{K}} = \langle m_k \mathcal{E}_k[U_j^n]\,\Delta v\rangle_{\mathcal{K}}$. To this aim, observe that the natural approximation

$$\mathcal{E}_k[U_j^n] = M_f(x_j, v_k, t_n),\ k \in \mathcal{K},\ n \geq 0,\ j \in \mathcal{J} \tag{32}$$

cannot satisfy these requirements, due to the truncation of the velocity space and to the piecewise constant approximation of the distribution function. Thus, the calculation carried out above for the definition of the initial distribution f, can be also performed for the equilibrium distribution M_f. This should be done each time we invoke the equilibrium distribution during the computation. The function $\mathcal{E}[U]$ is therefore given by the solution of the same minimization problem defined in (25), and its explicit value is given mimicking (31) by

$$\mathcal{E}[U] = M_f[U] + C^T(CC^T)^{-1}(U - CM_f[U]), \tag{33}$$

where $M_f[U]$ represents the pointwise values of the Maxwellian distribution $M_f[U] = M_f(x_j, v_k, t_n)$. Notice that the computation of the new distributions f and $\mathcal{E}$ only involves a matrix-vector multiplication. In fact, matrix C only depends on the parameter of the discretization and thus it is constant in time. In other words matrices C and $C^T(CC^T)^{-1}$ can be precomputed and stored in memory during the initialisation step. They are used during the simulation when the solution of system (25) is invoked.

Another possibility to approximate the Maxwellian distribution M_f is proposed in [28]. In that work, the authors define $\mathcal{E}_k[U]$ as the solution of a discrete entropy minimization problem

$$H_{\mathcal{K}}(\mathcal{E}[U]) = \min\left\{H_{\mathcal{K}}(g), g \geq 0 \in \mathbb{R}^N \text{ such that } \langle mg\rangle_{\mathcal{K}} = U\right\}. \tag{34}$$

This discretization (existence, uniqueness, convergence) has been mathematically studied in [28]. However, one drawback of this method, is the need for solving a non linear system of equations in each spatial cell for each time step. As we seek for efficiency, we only consider the first minimization strategy (25) to approximate the equilibrium distribution M_f.

4.2 Conservative and fast kinetic schemes FKS

We can now present the full scheme. Let us start with the first-order splitting scheme and then, define the second-order in time method based on the Strang splitting strategy [37].

Let $f_{j,k}^0$ be the initial data defined as a piecewise constant function in space and in velocity space, solution of equation (31) with $\widetilde{f}_{j,k}^0 = f(x_j, v_k, t = 0)$. We recall that the choice of a piecewise constant function in space is not mandatory for the method. Let also $\mathcal{E}_{j,k}^0[U]$ be the initial equilibrium distribution solution of equation (33) with $M_{j,k}^0 = M_f(x_j, v_k, t = 0)$. We start describing the first time step of the method $[t^0; t^1]$ starting at $t^0 = 0$, we further generalize the method to the generic time step $[t^n; t^{n+1}]$ starting from t^n.

First time step $[t^0; t^1]$. Let us describe the transport and relaxation stages.
Transport stage. We need to solve N linear transport equations of the form:

$$\partial_t f_k + v_k \cdot \nabla_x f_k = 0, \quad k = 1, \ldots, N, \tag{35}$$

where the initial data for each of the N equations is a piecewise constant function in the three dimensional space defined as

$$\overline{f}_k(x, t^0 = 0) = f_{j,k}^0 \quad \forall x \in [x_{j-1/2}, x_{j+1/2}], \quad k = 1, \ldots, N. \tag{36}$$

The exact solution of the N equations at time $t^1 = t^0 + \Delta t = \Delta t$ is given by

$$\overline{f}_k(x, t^1) = \overline{f}_k^*(x) = f(x - v_k\Delta t), \quad k = 1, \ldots, N. \tag{37}$$

Observe that, here, we do not need to reconstruct our function as for instance in the semi-Lagrangian schemes [17, 18], the shape of the function in space is in fact known and fixed at the beginning of the computation. Once the solution of the transport step is known, to complete one step in time, we need to compute the solution of the relaxation step. As in finite volume or finite difference methods, we solve the relaxation step only on the grid, thus only the value of the distribution function f in the centers of the cells are computed. From the exact solution of the function f_k we can immediately recover these values at the cost of one simple vector multiplication. On the other hand, one notices that for classical finite difference or finite volume methods nested loops for each dimension in space and in velocity space are mandatory to compute the solution of the transport part. This makes the computational cost of these methods extremely demanding in the multidimensional cases. On the contrary, the computational cost of the method we propose is only of the order of the number of points in which the velocity space is discretized ($O(N)$). In particular, for uniform meshes, we only need to compute the new value of f_k in the center of one single cell, to know the solution in the center of all others cells.

Relaxation stage. For this step we need to locally solve on the grid, *i.e.* in the center of each spatial cell, an ordinary differential equation. Thus, we have to solve:

$$\partial_t f_{j,k} = \frac{1}{\tau}(\mathcal{E}_{j,k}[U] - f_{j,k}), \quad k = 1, \ldots, N, \quad j = 1, \ldots, M, \tag{38}$$

where the initial data is the result of the transport step

$$\overline{f}_k(x_j, t^1) \equiv f_k^*(x_j), \quad k = 1, \ldots, N, \quad j = 1, \ldots, M. \tag{39}$$

Any discretization method in time for this term can be chosen, as for instance the preferred Runge Kutta method. However, being the above equation a first order linear ordinary differential equation, we choose to compute the exact solution. The last ingredient needed to perform the computation, is the value of the equilibrium distribution $\mathcal{E}$ at the center of the cell after the transport stage. To this aim, observe that, the Maxwellian distribution does not change during the relaxation step, which means that during this step the macroscopic quantities remain constants. This implies that only the transport stage possibly modifies the equilibrium distribution. In order to compute the Maxwellian, the macroscopic quantities in the center of the cells, *i.e.* the density, the mean velocity and the temperature, are given by summing the local value of the discrete distribution f over the velocity set $\langle m_k f_{j,k}^* \Delta v\rangle_{\mathcal{K}} = U_j^1,\ j = 1, \ldots, M$, where $f_{j,k}^* = f_k^*(x_j)$. Finally, the discrete equilibrium distribution at time $t^1 = t^0 + \Delta t$ is the solution of equation (33) with moments $U_j^1,\ j = 1, \ldots, M$. We can now compute the solution of the relaxation stage by

$$f_{j,k}^1 = \exp(-\Delta t/\varepsilon)\overline{f}_{j,k}^* + (1 - \exp(-\Delta t/\varepsilon))\mathcal{E}_{j,k}^1[U]. \tag{40}$$

Observe that the above equation furnishes only the new value of the distribution at time $t^1 = t^0 + \Delta t = \Delta t$ in the center of each spatial cell for each velocity v_k. However, what we need, in order to continue the computation, is the value of the distribution f in all points of the space. To overcome this problem, in classical discrete velocity methods several authors [28, 31] consider the distribution function constant in the cell as well as the Maxwellian distribution. The result is that they need to solve only an ordinary differential equation in the center of the cell taking the average value of the macroscopic quantities inside one cell. Here, we make a different approximation. We consider that the equilibrium distribution M_f has the same form as the distribution f in space. In other words $\mathcal{E}_k$ is a piecewise constant function in space for each velocity v_k. The values of this piecewise constant function are the values computed in the center of the spatial cells, *i.e.* one defines

$$\overline{\mathcal{E}}_k(x, t^1) = \mathcal{E}_{j,k}^1, \quad \forall x \text{ s.t. } \overline{f}_k(x, t^1) = \overline{f}_k(x_j, t^1),\ j = 1, \ldots, M. \tag{41}$$

This further implies that the relaxation term writes in term of spacial continuous function $\overline{f}_k(x, t^1)$ as

$$\overline{f}_k(x, \Delta t) = \exp(-\Delta t/\varepsilon)\overline{f}_k(x, t^1) + (1 - \exp(-\Delta t/\varepsilon))\overline{\mathcal{E}}_k(x, t^1)[U]. \tag{42}$$

For each velocity v_k this choice permits to keep the form of the distribution f_k constant in space throughout the computation, and, as a consequence it drastically reduces the computational cost. This ends the first time step.

We focus now on the time marching procedure for the first- and second-order splitting schemes which will allow to solve the Bolzmann-BGK equation.

Generic time step $[t^n; t^{n+1}]$. We present a first-order and second-order Strang splitting technique [37].

First-order splitting: Given the value of the distribution function $\overline{f}_k^n(x)$, for all $k = 1, \ldots, N$, and all $x \in \mathbb{R}^d$ at time t^n, the value of the distribution at time t^{n+1}, $\overline{f}_k^{n+1}(x)$, is given by

$$\overline{f}_k^*(x) = f_k^n(x - v_k\Delta t), \quad k = 1, \ldots, N \tag{43}$$

$$\overline{f}_k^{n+1}(x) = \exp(-\Delta t/\varepsilon)\overline{f}_k^*(x) + (1 - \exp(-\Delta t/\varepsilon))\overline{\mathcal{E}}_k^{n+1}(x)[U], \quad k = 1, \ldots, N, \tag{44}$$

where $\overline{\mathcal{E}}_k^{n+1}(x)[U]$ is a piecewise constant function, computed considering the solution of the minimization problem (33) relative to the moments value in the center of each spatial cell after the transport stage: $U_j^{n+1},\ j = 1, \ldots, M$. These moments are given by computing $\langle m_k f_{j,k}^* \Delta v\rangle_{\mathcal{K}}$ where $f_{j,k}^*$ is the value that the distribution function takes after the transport stage in the center of each spatial cell.

Second-order splitting: Given the value of the distribution function $\overline{f}_k^n(x),\ k = 1, \ldots, N$, $x \in \mathbb{R}^d$ at time t^n, the scheme reads

$$\overline{f}_k^*(x) = f_k^n(x - v_k\Delta t/2), \quad k = 1, \ldots, N \tag{45}$$

$$\overline{f}_k^{**}(x) = \exp(-\Delta t/\varepsilon)\overline{f}_k^*(x) + (1 - \exp(-\Delta t/\varepsilon))\overline{\mathcal{E}}_k^*(x)[U], \quad k = 1, \ldots, N, \tag{46}$$

where $\overline{\mathcal{E}}_k^*(x)[U]$ is a piecewise constant function, computed considering the solution of the minimization problem (33) relative to the moments values in the center of each spatial cell after the transport stage of size $\Delta t/2$. We call these moments $U_j^*,\ j = 1, \ldots, M$. They are given by the discrete summation $\langle m_k f_{j,k}^* \Delta v\rangle_{\mathcal{K}}$ where $f_{j,k}^*$ is the value that the distribution function takes after the transport stage in the center of each spatial cell. The last step consists of a second transport stage of half time step

$$\overline{f}_k^{n+1}(x) = f_k^{**}(x - v_k\Delta t/2), \quad k = 1, \ldots, N, \tag{47}$$

which ends the second-order splitting scheme.

Remark 2

- *As already mentioned the choices of uniform meshes and piecewise constant functions in space are not necessary for the construction of the method. These choices have been made because we wanted to analyze the method in its simplest form. We postpone to future works the study of non-uniform meshes and different shapes of the distribution function f in space. However, a key point is that, even if the method in its general form is already much more faster than finite volume, finite difference or semi Lagrangian methods for kinetic equations, it can be made extremely fast in the case of uniform meshes as we will explain in the next paragraph.*
- *For finite volume or finite difference methods applied to discrete velocity models of kinetic equations, the second-order time splitting implies the computation of the transport stage in two steps, from t^n to $t^{n+1/2}$ and from $t^{n+1/2}$ to t^{n+1}. Conversely, the same operation can be done with the relaxation step to get second order accuracy.*

- *In our method, extending the scheme from first- to second-order time splitting is almost as expensive as the first-order. In fact, except for the first time step in which we need to compute two times the transport operator with $\Delta t/2$, starting from the second time step we have to solve a sequence of two $\Delta t/2$ transport stages. However, being the transport computed exactly, solving the linear transport equations two times with $\Delta t/2$ or only one time with the entire Δt provides the same solution. This means that, in order to obtain second-order accuracy it is sufficient to solve the first time step with $\Delta t/2$ and then proceed as for the first-order method to obtain global second-order accuracy in time.*
- *However, any time splitting method does degenerate to first-order accuracy in the fluid limit, that is to say, when $\tau \to 0$.*
- *Due to the fact that the relaxation stage preserves the macroscopic quantities, the scheme is globally conservative. In fact, at each time step, the change of density, momentum and energy is only due to the transport step. This latter, being exact, does preserve the macroscopic quantities as well as the distribution function.*
- *For the same reason, the scheme is also unconditionally positive. In others words, we observe that $f_k^n(x) \geq 0$, for all $n > 0$, and $k = 1, \ldots, M$ if the initial datum is positive $f_k^0(x) \geq 0$ for all $k = 1, \ldots, M$. In fact, the transport maintains the shape of f unchanged in space while the relaxation towards the Maxwellian distribution is a convex combination of M_f and $f(x - v_k \Delta t)$ both being positive.*
- *We expect the scheme to perform very well in collisionless or almost collisionless regimes. In these cases in fact the relaxation stage is neglectible and only the exact transport does play a role. When moving from rarefied to dense regimes the projection over the equilibrium distribution becomes more important. Thus, the accuracy of the scheme is expected to diminish in fluid regimes, because the projection method is only first-order accurate. One possibility, for such regimes is to increase the order of the projection method towards the equilibrium. This possibility will also be analyzed in future works.*
- *The time step Δt is chosen as the classical CFL condition*

$$\Delta t \max_k \left(\frac{|v_k|}{\Delta x} \right) < 1. \tag{48}$$

Observe that this choice is not mandatory, in fact the scheme is always stable for every choice of the time step, but being based on a time splitting technique the error is of the order of Δt or $(\Delta t)^2$. This suggests to take the usual CFL condition in order to maintain the error small enough.

5 Analogies with particles methods

In this section we first introduce a Monte Carlo particle method which permits to solve the Boltzmann-BGK equation. Next, we introduce its deterministic counterpart, *i.e* a

deterministic particle method. Finally, we show that a slightly modified version of this latter method, in which the positions of the particles, instead of being randomly chosen, are taken initially at the same position in space for all the cells, is equivalent to a FKS method where some specific choices of the discretization parameters are done. This analogy permits to derive a very convenient form of the algorithm which for this choice of the discretization parameters.

The starting point of Monte Carlo methods is again given by a time splitting between free transport

$$\partial_t f + v \cdot \nabla_x f = 0, \tag{49}$$

and collision, which in the case of the BGK operator is substituted by a relaxation towards the equilibrium

$$\partial_t f = \frac{1}{\tau}(f - M_f[U]). \tag{50}$$

In Monte Carlo simulations the distribution function f is discretized by a finite set of particles

$$f = \sum_{i=1}^{N} \mathfrak{m}_i \, \delta(x - x_i(t)) \delta(v - v_i(t)), \tag{51}$$

where $x_i(t)$ represents the particle position, $v_i(t)$ the particle velocity and $\mathfrak{m}_i$ the particle mass which is usually taken constant. During the transport stage the particles move to their next positions according to

$$x_i(t + \Delta t) = x_i(t) + v_i(t) \Delta t, \tag{52}$$

where Δt is such that an appropriate CFL condition holds. This condition normally implies that one particle does not cross more than one cell in one time step.

The collision step acts only locally, changes the velocity distribution but preserves the macroscopic quantities. In this case, as already explained, the space homogeneous problem admits the following exact solution at time $t + \Delta t$

$$f(t + \Delta t) = e^{-\Delta t/\tau} f(t) + (1 - e^{-\Delta t/\tau}) M_f[U](t). \tag{53}$$

Thus, in a Monte Carlo method, the relaxation step consists in replacing randomly selected particles with Maxwellian particles with probability $(1 - e^{-\Delta t/\tau})$. This means

$$v_i(t + \Delta t) = \begin{cases} v_i(t), & \text{with probability } e^{-\Delta t/\tau} \\ M_f[U](v), & \text{with probability } 1 - e^{-\Delta t/\tau} \end{cases}, \tag{54}$$

where $M_f[U](v)$ in the above expression represents a particle sampled from the Maxwellian distribution with moments U. Observe that, second-order splitting can be used as well in the Monte Carlo methods. As in the case of the FKS, because the transport step is resolved exactly, the change with respect to the first-order method is only the first time step which has to be computed with a time step of $\Delta t/2$. This will assure second-order

accuracy in time except in the limit $\tau \to 0$ in which the method degenerates again to first-order accuracy.

We introduce now a modified particle method which shares many analogies with our method. Instead of the continuous kinetic equation, this modified particle approach solves the discrete velocity approximation of the kinetic equation. In this method, the distribution function is again represented by a piecewise constant function, defined on a compact support in the velocity space. The distribution function is approximated by a finite set of particles in each spatial cell as in the previous Monte Carlo method. The main difference with respect to the other particle method is that now the particles can attain only a discrete set of velocities and that the mass of each particle is no more a constant, instead it changes in time during the time evolution of the kinetic equation. These types of methods are known in literature as weighted particles methods [11, 26, 27]. Therefore we consider

$$f = \sum_{i=1}^{N} \mathfrak{m}_i(t) \, \delta(x - x_i(t)), \quad \delta(v - v_i(t)), \quad v_i(t) = v_k, \quad k \in \mathcal{K}, \tag{55}$$

where $\mathcal{K}$ is the same set of multi-indices than the DVM discretization (this means that the number of particle is fixed equal to the number of points N in which the velocity space is discretized). The BGK equation is again split into two stages: a transport and a relaxation stage. The transport part, as before, corresponds to the motion of the particles in space caused by their velocities (52). The main difference is in the solution of the relaxation part (50). In order to solve this equation from a particle point of view, we change the mass of each particle using the exact solution of the relaxation equation, *i.e.*

$$f(t + \Delta t) = e^{-\Delta t/\tau} f(t) + (1 - e^{-\Delta t/\tau}) M_f[U](t). \tag{56}$$

this corresponds to

$$\mathfrak{m}_i(t + \Delta t) = e^{-\Delta t/\tau} f(v_i) + (1 - e^{-\Delta t/\tau}) \mathcal{E}(v_i)[U], \quad i = 1, \ldots, N. \tag{57}$$

Again in practice to avoid the loss of conservation of macroscopic quantities, once the conserved quantities are computed in one cell, we solve the minimization problem (33) to get the function $\mathcal{E}[U]$. Thus, the above procedure requires the knowledge of U_j, $j = 1, \ldots, M$, which can only be estimated from the sample positions. The simplest method, which produces a piecewise constant reconstruction, is based on evaluating the histogram of the samples on the grid, considering all the samples inside one cell be of the same importance irrespectively of their positions. In practice, the density ρ_j, $j = 1, \ldots, M$ is given by the number of samples N_{I_j} belonging to the cell I_j

$$\rho_j = \frac{1}{\Delta x} \sum_{x_i \in I_j} \mathfrak{m}_i, \tag{58}$$

while the mean velocity in each spatial direction and the energy are given by

$$u_j = \frac{1}{\rho_j} \sum_{x_i \in I_j} \mathfrak{m}_i v_i, \qquad E_j = \frac{1}{2\Delta x} \sum_{x_i \in I_j} \mathfrak{m}_i |v_i|^2. \tag{59}$$

222

The method described above deserves some remarks. First, note that as $\tau \to 0$ the method becomes a particle scheme for the limiting fluid dynamic equations. This limit method is the analogous of a kinetic particle method for the compressible Euler equations. Second, the simple splitting method described is first-order in time. Second order Strang splitting can be implemented similarly to the case of the FKS scheme described in the previous section.

Now, we dispose of all the elements which permit to highlight the similarities with the FKS scheme. Observe that the relaxation step (57) is no more solved statistically as for the original Monte Carlo method (54). Thus, the scheme described is in fact a deterministic particle scheme, in which, however, the particle positions are still randomly initialized. Now, if we consider the piecewise reconstruction of the macroscopic quantities introduced before (58-59), we take one single particle for each velocity $v_k, k \in \mathcal{K}$ and we fix all particles positions at the beginning of the computation at the center of each cell we obtain the FKS described in the previous section. In fact, first the number of particles in each spatial cell remains constant in time and equal to the number of mesh point in velocity space N. This is because for each particle that goes out of one cell, there exists another particle with the same velocity which enters in the cell from another location. This is due to the fact that particles have initially the same position, they never change velocity and the mesh is uniform. Thus, during the time evolution the only quantity that is modified is the mass of the particle. This mass changes according to the solution of the relaxation equation (57). This is exactly what happens in the FKS method in equation (44). Finally, the transport is solved exactly for the particle scheme as well as for the FKS method. However, the weighted particle scheme, can be viewed as a particular case of the FKS method. In fact, to regain the weighted particle method, we have to fix the position of the particles, take only a single particle for a given velocity v_k, the mesh must be uniform and the shape of the distribution function in space must be piecewise constant for the FKS method. This analogy between the two schemes permits, from one side, to derive a very efficient algorithm for the FKS method. From the other side, it opens the way to in deep discussions from the theoretical point of view on the relation between the two methods , like the different convergence properties of the two approaches. We remind to a future work for an analysis of the convergence of the FKS method.

6 Numerical tests

6.1 General setting

In this section, we present several numerical tests to illustrate the main features of the method. First the performance of the scheme is tested in the one dimensional case for solving the Sod problem. In this case, we do comparisons of our method with different finite difference methods which can solve the same problem. In the one dimensional case, the computational speedup is not very relevant being all classical methods sufficiently fast. However, the FKS method is still faster than the other methods. In a second series of tests we solve a two dimensional- two dimensional kinetic equation. Finally we solve a full three-three dimensional problem. In this situation, it is a matter of fact that computing

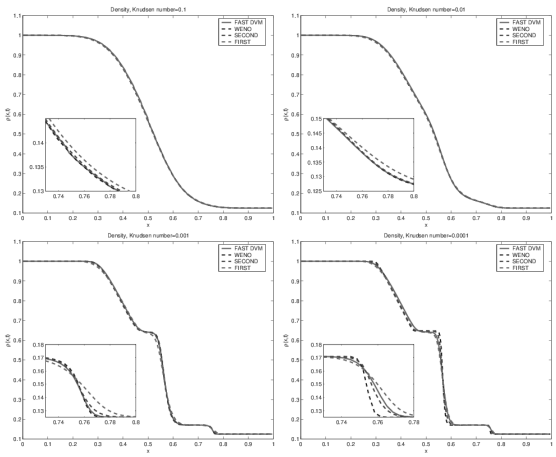


Figure 1: Sod test: solution at $t_{\text{final}} = 0.05$ for the density, with $\tau = 10^{-1}$ (top left), $\tau = 10^{-2}$ (top right), $\tau = 10^{-3}$ (bottom left) and $\tau = 10^{-4}$ (bottom right).

the solution of a kinetic equation with finite difference, finite volume or semi-Lagrangian methods is unreasonable. We will show results from our method running on a mono-processor laptop machine.

6.2 1D Sod shock tube problem

We consider the 1D/1D Sod test with 300 mesh points in physical and 100 points in velocity spaces. The boundaries in velocity space are set to -15 and 15. The left and right states are given by a density $\rho_L = 1$, mean velocity $u_L = 0$ and temperature $T_L = 5$ if $0 \leq x \leq 0.5$, while $\rho_R = 0.125$, $u_R = 0$, $T_R = 4$ if $0.5 \leq x \leq 1$. The gas is in thermodynamical equilibrium. We repeat the same test with 4 different values of the Knudsen number, ranging from $\tau = 10^{-1}$ to $\tau = 10^{-4}$. We plot the results for the final time $t_{\text{final}} = 0.05$ for the density (Figure 1), the mean velocity (Figure 2) and the

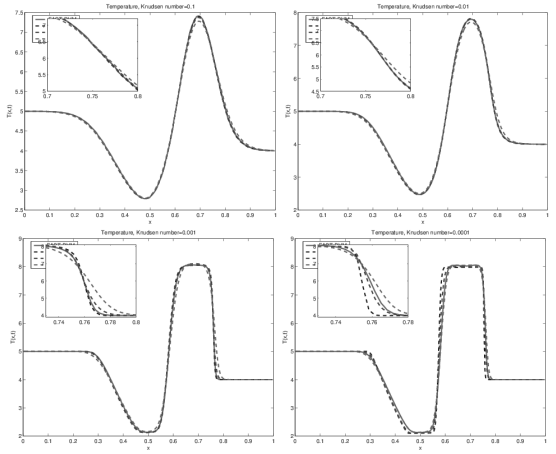


Figure 3: 1D Sod test: solution at $t_{\text{final}} = 0.05$ for the temperature, with $\tau = 10^{-1}$ (top left), $\tau = 10^{-2}$ (top right), $\tau = 10^{-3}$ (bottom left) and $\tau = 10^{-4}$ (bottom right).

step of the method (44)), does reduce the accuracy of the method. Notice that high order reconstruction of the equilibrium distribution could also be considered to increase the global accuracy in such case. However, a key point of the FKS is its low CPU time consumption in comparison to other existing methods. In the case $\tau = 10^{-4}$ for which the scheme exhibits diffusive behaviors, a comparison between the third order WENO method and our FKS method is carried out for a fixed CPU time. In other words, we consider for a given total computational time, which method gives better results. Thus, we solve the problem with 200 points in space and 100 in velocity space for the WENO method and we consider an FKS solver which employs 100 points in velocity space. In order to have the same computational time for the two methods, we can afford 1000 points for the FKS. The two results are compared in Figure 4. We observe that, in this situation, the FKS method gives more accurate solutions, in particular for the shock wave (see the zooms in the figures). Finally, observe that the gain in term of computational time is not

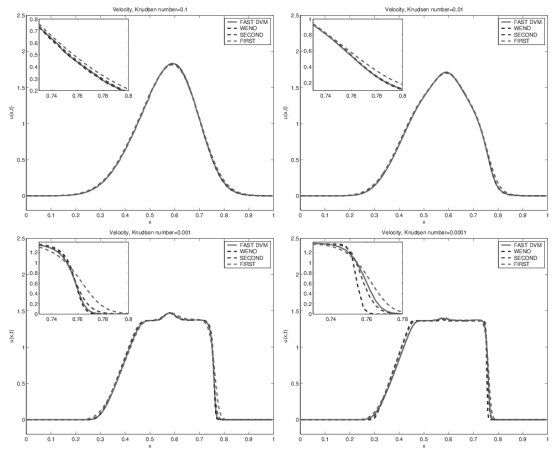


Figure 2: 1D Sod test: solution at $t_{\text{final}} = 0.05$ for the mean velocity, with $\tau = 10^{-1}$ (top left), $\tau = 10^{-2}$ (top right), $\tau = 10^{-3}$ (bottom left) and $\tau = 10^{-4}$ (bottom right).

temperature (figure 3). In each figure we compare the FKS method with a third order WENO method, a second-order MUSCL method and a first-order upwind method [23]. These numerical methods used as reference, employ the same discretization parameters, except for the time step which for stability reason is chosen equal to $\Delta t/2$ for the WENO and second-order MUSCL schemes, where Δt is the time step of the fast DVM method given by (48).

From Figures (1) to (3) we can observe that our method gives very similar results to the two high order schemes for $\tau = 10^{-1}$, $\tau = 10^{-2}$ and $\tau = 10^{-3}$ while for $\tau = 10^{-4}$, the scheme is more diffusive than the second and third order scheme but it still performs better than the first order method. The behaviors of the method for different regimes are due to the fact that for collisionless regimes the FKS gives almost the exact solution, this means that it is more precise than the third and second-order methods. When the gas becomes denser the projection towards the equilibrium, which is only first-order (second

so relevant for the one dimensional case, while it becomes very important for the two and the three dimensional case. In the later case, the difference is about being able to do or not to do the computation in a reasonable amount of time on a single processor machine.

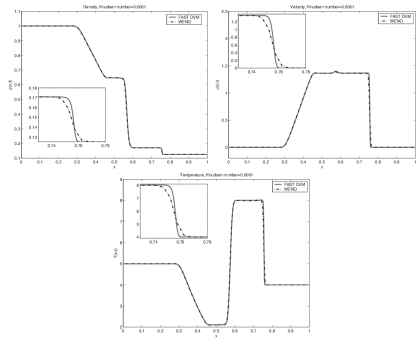


Figure 4: 1D Sod test: solution at $t_{\text{final}} = 0.05$ for the density, the mean velocity and the temperature with $\tau = 10^{-4}$. Comparison of solutions for the same computational time and different meshes. WENO 200 points (dashed line) and Fast DVM 1000 points (straight line).

6.3 2D Sod shock tube problem

We consider now the 2D/2D Sod test on a square $[0, 2] \times [0, 1]$. The velocity space is also a square with bounds -15 and 15, *i.e.* $[-15, 15]^2$, discretized with $N_v = 20$ points in each direction which gives 20^2 points. We repeat the same test using different $N_x \times N_y$ meshes ranging from $N_x = N_y = 25$ to $N_x = N_y = 200$. The domain is divided into two parts, a disk centered at point $(1, 1)$ of radius $R_d = 0.2$ is filled with a gas with density $\rho_L = 1$, mean velocity $u_L = 0$ and temperature $T_L = 5$, whereas the gas in the rest of the domain is initiated with $\rho_R = 0.125$, $u_R = 0$, $T_R = 4$. The final time is $t_{\text{final}} = 0.07$. The gas is in thermodynamical equilibrium during all the computation which means that we fix $\tau = 0$. In practice, we are using the kinetic scheme to compute the solution of the compressible Euler equation. We recall that, as seen in the previous section, this is the case in which the FKS scheme gives the worse results, this is due to the first order accurate projection

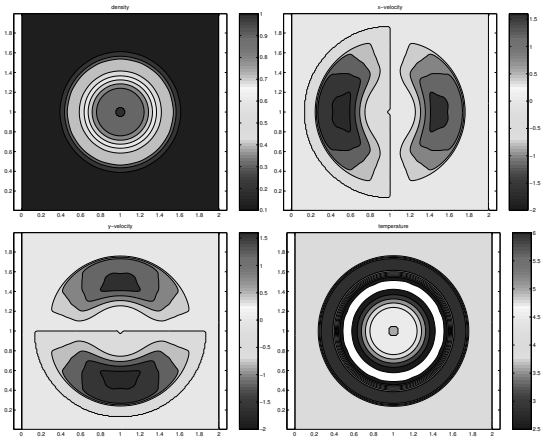


Figure 5: 2D Sod test: solution at $t_{\text{final}} = 0.07$ for the density (top left), the velocity in the x-direction (top right), the velocity in the y-direction (bottom left) and the temperature (bottom right).

towards the local Maxwellian distribution. However, this choice permits to compare our results with a numerical method for the compressible Euler equations, being as already stated, computationally very demanding to perform simulations of kinetic equations in the two dimensional case and considerably more demanding in the three dimensional case.

In Figure 5 we show the results for respectively the density, the mean velocity in the x-direction and in the y-direction and the temperature using a 200×200 mesh. In Figure 6 we report the profile for $x = 1$ of the same macroscopic quantities comparing the results to a first order and to a second order MUSCL scheme for the compressible Euler equations [23]. We clearly see that, as in the 1D case, the accuracy of the FKS method lies between the first and the second order accuracy in the limit $\tau \to 0$. We expect the accuracy to be highly improved when the gas is far from the thermodynamical equilibrium as in the one dimensional case.

In table 1 we report the CPU time T of these simulations, the CPU time per time cycle

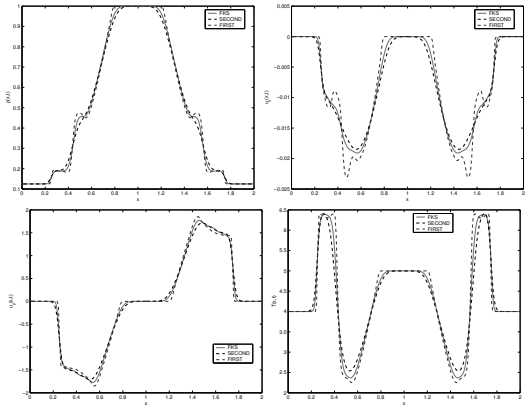


Figure 6: 2D Sod test: solution (continuous line) at $t_{\text{final}} = 0.07$ and $x = 1$ for the density (top left), the velocity in the x-direction (top right), the velocity in the y-direction (bottom left) and the temperature (bottom right). Comparisons with first order and second order MUSCL methods (dotted lines)

T_{cycle}, the CPU time per cycle per cell T_{cell} and the number of cycles needed to perform the computation for different meshes in space and a fixed mesh in velocity. As expected the number of time step linearly scales with the size of the spatial mesh at fixed velocity mesh (factor 2 when the cell number is multiplies by 4). The CPU time is very small compared to classical kinetic schemes, in less than 10 minutes the simulation of the Sod shock tube on a 200 mesh is computed. Finally we observe that the CPU time per cycle per cell is almost constant which allows to predict the end of the simulation and its cost beforehand.

6.4 Numerical validation of the 3D/3D fast FKS method

Here we report some simulations of the full 3D/3D problem. We consider only the case in which $\tau \equiv 0$, which means, we project towards equilibrium at each time step, this is the fluid limit. We recall that, in this regime, the numerical method gives the worst results in

Cell v # N_v^2 Bounds	**Cell x # N_c** $N_x \times N_y$	**Cell $x \times v$ # N_{tot}** $N_x \times N_y \times K^2$	**Cycle** N_{cycle}	**Time** T (s)	**Time/cycle** T_{cycle} (s)	**Time/cell** T_{cell} (s)
20^2 $[-15, 15]$	25×25 $= 625$	$25 \times 25 \times 20^2$ $= 250000$	13	2s	0.1538	2.46×10^{-4}
	50×50 $= 2500$	$50 \times 50 \times 20^2$ $= 10^6$	25	8s	0.32	1.28×10^{-4}
	100×100 $= 10000$	$100 \times 100 \times 20^2$ $= 4\ 10^6$	50	60s 1mn	1.2	1.20×10^{-4}
	200×200 $= 40000$	$200 \times 200 \times 20^2$ $= 16\ 10^6$	100	490s $\sim$ 8mn	535.75	1.22×10^{-4}

Table 1: 2D Sod shock tube. The time per cycle is obtained by $T_{\text{cycle}} = \text{T}/N_{\text{cycle}}$ and the time per cycle per cell by $T_{\text{cell}} = \text{T}/N_{\text{cycle}}/N_c$.

terms of precision, on the other hand, exact solution are known and this permits to make fair comparisons. For all the other regimes, the performances of the method are better as shown in the previous section.

The FKS method has been implemented in fortran on a sequential machine. The goal is to numerically show that such a kinetic scheme can reasonably perform on six dimensions on a mono-processor laptop. All simulations have been carried out on a HP EliteBook 8740W Intel(R) Core(TM) i7 Q840@1.87GHz running under a Ubuntu (oneiric) version 11.10. The code has been compiled with gfortran 4.6 compiler with -O3 optimization flags.

Otherwise noticed the velocity space is $[-15, 15]^3$ or $[-10, 10]^3$ and is discretized with $N_v = 13$ or $N_v = 12$ grid points in each velocity direction, leading to $N_v^3 = 2197$ or 1728 mesh points. The time step is fixed to 95% of the maximum time step allowed, as prescribed by the CFL condition (48), apart from the last time step which is chosen to exactly match the user-given final time. Symmetric boundary conditions are considered.

The Sod shock tube in 1D is run as a sanity checks in order to validate the implementation of the method and show its ability to reproduce 1D results with a 3D run. Then the Sod problem in 3D is simulated to show the performances of the FKS algorithm and further compared to a reference solution. For each simulation we report the memory consumption, the full CPU time and the CPU time cost per cell per time step. Some extrapolation of these results are also made to measure the efficiency of this method.

6.4.1 1D Sod shock tube problem: A sanity check

The first sanity check consists of running the 1D Sod shock tube in x direction on $N_x \times 2 \times 2$ cubes. The initial data are the same as for the 1D problem previously run. The final time is $t_{\text{final}} = 0.1$. In our numerical experiments the computational domain is of size 1 in x direction leading to $\Delta x = 1/N_x$. We set $\Delta y = \Delta z = \Delta x$. Four successively refined meshes in x direction are utilized, $N_x = 50, 100, 200$, and 400, in order to observe the convergence of the numerical method towards the exact solution.

In Figure 7 we display the density, the velocity and the temperature *vs* the exact solution with solid line (respectively panels **(a)**, **(c)** and **(d)**) and a 3D view on the mesh

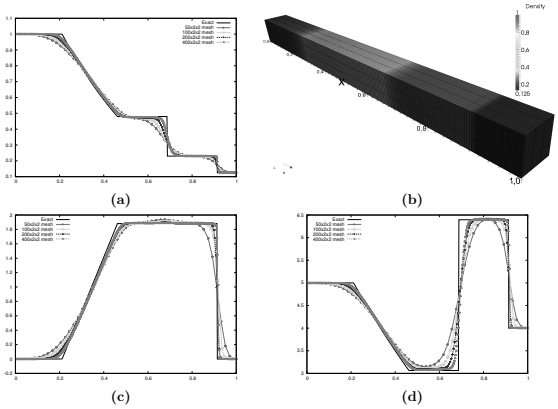


Figure 7: 3D-1D Sod problem at $t_{\text{final}} = 0.1$ for $50, 100$, 200, and 400 cells in x direction and 2 in y and z directions — Panels **(a)**, **(c)**, **(d)**: Density, velocity and temperature as a function of x *vs* exact solution (straight line) — Panel **(b)**: 3D view of colored density for a $200 \times 3 \times 3$ mesh.

cells colored by density (panel **(b)** where a $200 \times 3 \times 3$ mesh is used for figure scaling reasons). The first observation is the perfect symmetry in the ignorable directions y and z as all cells are plotted (notice that the results for a $N_x \times 5 \times 5$ cells mesh exactly match the $N_x \times 2 \times 2$ results). The second obvious observation is the convergence of the numerical solution towards the exact solution when the mesh is refined. These results assess the ability of the method and the code to reproduce 1D results without alteration.

In table 2 we gather the number of cycles N_{cycle}, the CPU time T of these simulations and display the CPU time per time cycle T_{cycle} and the CPU time per cycle per cell T_{cell}. As expected, the cycle number and the CPU time per time cycle scales with the cell number and, consequently, the CPU time per cycle per cell is almost constant. This allows to almost exactly predict the duration of a simulation knowing the cell number. Moreover we have provided the relative percentage of the cost of the transport and collision stages.

As expected the transport stage does not cost anything, in absolute value, especially when the number of cells increases. In fact, for computing the solution of this stage in

Cell # $N_x \times N_y \times N_z \times N_v^3$ $= N_c \times N_v^3$	**Cycle** N_{cycle}	**Time** T(s)	**Time/cycle** T_{cycle}(s) *Transp.* *Coll.*	**Time/cycle/cell** T_{cell}(s)	**Memory** Mem(MB)
$50 \times 2 \times 2$ $= 200 \times 13^3 = 439400$	81	18	0.22 0.07% 99.93%	1.11×10^{-3}	0.660
$100 \times 2 \times 2$ $= 400 \times 13^3 = 878800$	160	68	0.43 0.05% 99.95%	1.07×10^{-3}	0.704
$200 \times 2 \times 2$ $= 800 \times 13^3 = 1757600$	318	276	0.87 0.03% 99.97%	1.08×10^{-3}	0.812
$400 \times 2 \times 2$ $= 1600 \times 13^3 = 3515200$	634	1071	1.69 0.02% 99.98%	1.06×10^{-3}	1.000

Table 2: 1D Sod shock tube run with the 3D/3D FKS method. The time per cycle is obtained by $T_{\text{cycle}} = \text{T}/N_{\text{cycle}}$ and the time per cycle per cell by $T_{\text{cell}} = \text{T}/N_{\text{cycle}}/N_c$. The relative percentage of the cost of the transport and relaxation stages are provided. For our FKS method the transport stage costs almost nothing.

all domain, we consider the evolution of the distribution function f in one single cell, the same happens in the other cells. This means that the cost of this stage is proportional to the N_v^3 mesh points in the velocity space. On the other hand, in finite volume methods as well as Monte Carlo method the cost to solve this stage is proportional to $N_v^3 N_c$ with $N_c = N_x N_y N_z$ and, obviously this scales with N_c. Another satisfactory result is the memory storage Mem in MB (or Mo) of the method which is very low because we never have to store the distribution function values for more than N_v^3 points, leading to store $13^3 \times 7$ reals, say ~ 0.123MB independently of N_c. Conversely the storage of the Monte Carlo method scales with the cell number N_c. Finally as expected the time T scales with a factor 4 for twice the number of cells.

6.4.2 3D Sod shock tube problem

The 3D Sod shock tube has been run with the 3D/3D FKS method. The left state of the 1D Sod problem is set for any cell c with cell center radius $r_c \leq 1/2$, conversely the right state is set for cell radius $r_c > 1/2$. The final time is $t_{\text{final}} = 0.1$. The domain is the unit cube and the mesh is composed of $N_x \times N_x \times N_x$ cells with $\Delta x = 1/N_x$ and $\Delta x = \Delta y = \Delta z$. The problem is run with $N_x = 50$ (125000 cells), $N_x = 100$ (1 million cells) and $N_x = 200$ (8 millions cells). The velocity space is either $[-10; 10]$ discretized with 12^3 points, or $[-15; 15]$ discretized with 13^3 points. This leads to consider up to $200^3 \times 13^3 \simeq 17.7$ milliards cells. In Figure 8 the density is plotted as a function of the radius (left panel) and the colored density on a 3D view (right panel) for $N_x = 50$ (middle panels) and $N_x = 200$ (bottom panels). The two different choices for the bounds and the mesh points in velocity space do not significantly change the results hence only the solution with bounds $[-10; 10]$ and with 12^3 mesh points is reported. The reference solution is obtained with a 2D axisymmetric compatible staggered Arbitrary-Lagrangian-Eulerian code [25] with 1000 cells in radial and 20 cells in angular directions. Moreover in

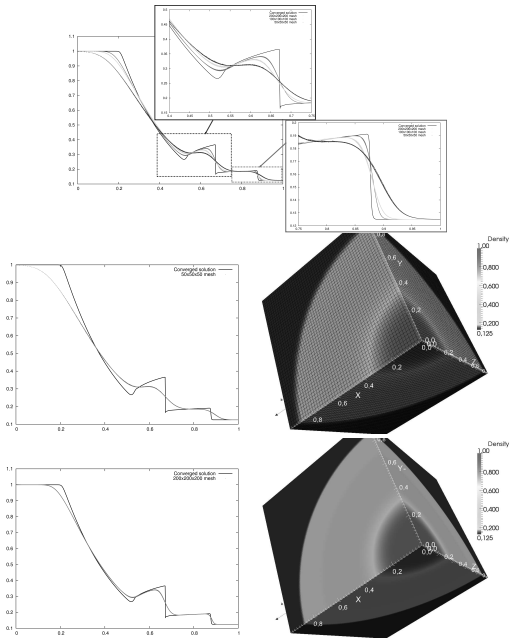


Figure 8: Sod problem at $t_{\text{final}} = 0.1$ for $N_x \times N_x \times N_x$ cells (for $N_x = 50, 100, 200$) for the velocity space $[-10; 10]$ discretized with 12^3 mesh points. — Top: Convergence of density as a function of cell center radius for all cells *vs* converged solution (straight thick line) for the three meshes with zooms on contact and shock waves. Left: Density as a function of cell center radius (middle: $N_x = 50$, bottom: $N_x = 200$) Right: 3D view of density on the unit cube $N_x = 50$ (middle) and $N_x = 200$ (bottom) (the mesh is only shown for $N_x = 50$).

N_v^3 Bounds	**Cell #** $N_c \times N_v^3$ $N_x \times N_y \times N_z \times N_v^3$	**Cycle** N_{cycle}	**Time** T (s)	**Time/cycle** T_{cycle} (s)	**Time/cell** T_{cell} (s)	**Mem** (MB)
13^3 $[-15, 15]$	$25^3 \times 13^3$ $= 3.4328125 \times 10^6$	32	346s (5.76mn)	10.81	6.92×10^{-4}	2.4
	$50^3 \times 13^3$ $= 274.625000 \times 10^6$	81	4900s (1.36h)	60.50	4.84×10^{-4}	15.5
	100×13^3 $= 2.1970 \times 10^9$	160	85720s (23.8h)	535.75	5.36×10^{-4}	115.5
extrapol.	200×13^3 $= 1.7576 \times 10^{10}$	320	$\sim 1.4 \times 10^6$s (16d)	~ 4400	5.5×10^{-4}	~ 900
12^3 $[-10, 10]$	$25^3 \times 12^3$ $= 27 \times 10^6$	27	218s (3.63mn)	8.07	5.17×10^{-4}	2.3
	$50^3 \times 12^3$ $= 125 \times 10^3$	54	2702s (45mn)	50.03	4.00×10^{-4}	15.4
	$100^3 \times 12^3$ $= 1.728 \times 10^9$	107	38069s (10.57h)	355.79	3.56×10^{-4}	115.4
extrapol.	$200^3 \times 12^3$ $= 1.3284 \times 10^{10}$	214	~ 633440s (7d)	~ 2960	3.7×10^{-4}	~ 900

Table 3: 3D Sod shock tube. The time per cycle is obtained by $T_{\text{cycle}} = \text{T}/N_{\text{cycle}}$ and the time per cycle per cell by $T_{\text{cell}} = \text{T}/N_{\text{cycle}}/N_c$. The lines marked with *extrapol.* have been extrapolated by fixing N_c, N_{cycle} and T_{cell}.

Figure 8 (top panel) we present the convergence of the density as a function of cell center radius for all cells for the $50 \times 50 \times 50$, $100 \times 100 \times 100$ and $200 \times 200 \times 200$ cells meshes. These curves are compared to the reference solution in straight thick line and they show that the results are converging towards the reference solution. In table 3 we gather the number of time steps and the total CPU time T for 50^3 and 100^3 cell meshes for the two different configurations: one with $N_v = 13$ and the velocity space $[-15, 15]$ and the second one with $N_v = 12$ and the velocity space $[-10, 10]$. For the 50^3 mesh the simulation takes 45 minutes or 1.36 hour depending on the configuration. For the finer 100^3 mesh the simulation takes either 11 hours or 24 hours The memory consumption ranges from 124Mb to 924Mb depending on the configurations and it scales with the number of cells N_c.

Then, we compute the cost per cycle T_{cycle} and per cycle per cell T_{cell}. One observe that the cost per cycle per cell is an almost constant equal to 4×10^{-4}s or 5.5×10^{-4}s. The extrapolation of the CPU time T for a 200^3 mesh at T_{cell} fixed leads to one or two weeks computation for the two configurations and a memory storage of about 900MB. In Figure 9 we plot the CPU time (red or blue symbols for each configuration and mesh points of the velocity space) and the extrapolation curves $CPU(N_x, N_c, T_{\text{cell}}) = \frac{N_{\text{cycle}}}{N_x} N_c T_{\text{cell}}$ for the 3D Sod problem up to time $t_{\text{final}} = 0.1$ for single processor laptop computation on a fixed mesh in velocity space of $N_v = 12^3$ points. We deduced that the FKS method can be used at most on a single processor machine up to a $200 \times 200 \times 200$ cells for roughly

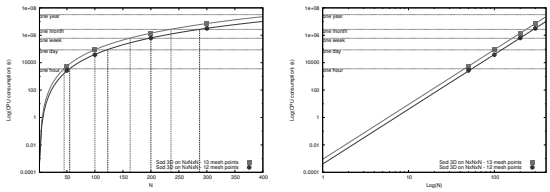


Figure 9: Left: Log of the CPU time consumption for the 3D Sod problem at $t_{\text{final}} = 0.1$ as a function of N (for $N \times N \times N$ cell meshes) on a single processor laptop The red/blue squares are taken from Table 3, the thick red/blue curves are the extrapolation curve from T_{cell}. The horizontal lines corresponding to one hour, one day, week, month and year are also plotted. $N = 100$ corresponds to the 'one million cells' in space — Right: Log/Log scale.

one week of computation. One also notices that the CPU time linearly scales on a log/log graph as expected (right panel of Figure 9)

7 Conclusions

In this work we have presented a new super efficient numerical method for solving kinetic equations. The method is based on a splitting between the collision and the transport terms. The collision part is solved on a grid while the transport linear part is solved exactly by following the characteristics backward in time. The key point is that, conversely to semi-Lagrangian methods, we do not need to reconstruct the distribution function at each time step. In this first paper, we have presented the basic formulation of this new method for the BGK equation: Uniform meshes, piecewise constant discretization of the velocity space and a simple projection towards the equilibrium distribution have been considered.

The numerical results show that the method is incredibly fast. We are now able to perform numerical simulations of the full six dimensional kinetic equation on a single processor machine in several hours. This important result opens the gate to extensive realistic numerical simulations of far from equilibrium physical models. Concerning the precision of the method, we observed, as expected, that the fast kinetic scheme (FKS) is more dissipative close to the fluid regime and very precise for gases far from the thermodynamical equilibrium.

In the future we would like to extend the method to non uniform meshes, more advanced boundary conditions and different discretization of the velocity space. One expects with this last point to increase the accuracy of the schemes without losing its attractive efficiency. To avoid the loss of accuracy close to the fluid limit, we want to couple the

FKS method to an high order solver for the system of equations which describes the fluid limit. Finally, we want to extend the method to other kinetic equations as the Boltzmann or the Vlasov equation.

References

[1] G.A.Bird, Molecular gas dynamics and direct simulation of gas flows, Clarendon Press, Oxford (1994).

[2] C.K. Birsdall, A.B. Langdon, Plasma Physics Via Computer Simulation, Institute of Physics (IOP), Series in Plasma Physics (2004).

[3] A.V. Bobylev, A. Palczewski, J. Schneider, *On approximation of the Boltzmann equation by discrete velocity models*. C. R. Acad. Sci. Paris Ser. I. Math. 320, (1995), pp. 639-644.

[4] J. Burt, I. Boyd, *A low diffusion particle method for simulating compressible inviscid flows*, J. Comput. Phys., Vol. 227, (2008), pp. 4653–4670

[5] R. E. Caflisch, *Monte Carlo and Quasi-Monte Carlo Methods*, Acta Numerica, (1998), pp. 1–49.

[6] R. E. Caflisch, L. Pareschi, *Towards an hybrid method for rarefied gas dynamics*, IMA Vol. App. Math., vol. 135 (2004), pp. 57–73.

[7] C. Cercignani, The Boltzmann Equation and Its Applications, Springer-Verlag, New York, (1988).

[8] N. Crouseilles, T. Respaud, E. Sonnendrcker, *A Forward semi-Lagrangian Method for the Numerical Solution of the Vlasov Equation*, Comp. Phys. Comm. 180, 10 (2009) 1730–1745.

[9] N. Crouseilles, M. Mehrenberger, E. Sonnendrcker, *Conservative semi-Lagrangian schemes for Vlasov equations*, J. Comp. Phys., (2010) pp. 1927–1953.

[10] L. Desvillettes, S. Mischler, *About the splitting algorithm for Boltzmann and BGK equations*. Math. Mod. & Meth. in App. Sci. bf 6, (1996), pp. 1079–1101.

[11] P. Degond, S. Mas-Gallic, *The weighted particle method for convection-diffusion equations. II. The anisotropic case*, Math. Comp. Vol. 53 (1989), pp. 509525.

[12] P. Degond, G. Dimarco, L. Pareschi, *The Moment Guided Monte Carlo Method*, Int. J. Num. Meth. Fluids, Vol.67, (2011), pp. 189–213.

[13] P. Degond, J.-G. Liu, L. Mieussens, *Macroscopic fluid models with localized kinetic upscaling effects*, SIAM MMS, vol. 5 (2006), pp. 940–979.

[14] G. Dimarco, L. Pareschi, *A Fluid Solver Independent Hybrid method for Multiscale Kinetic Equations*, SIAM J. Sci. Comput. Vol. 32, (2010), pp. 603–634.

[15] G. Dimarco, L. Pareschi, *Hybrid multiscale methods II. Kinetic equations*, SIAM Mult. Model. and Simul. Vol 6., (2007), pp. 1169–1197.

[16] W. E, B. Engquist, *The heterogeneous multiscale methods*, Comm. Math. Sci., vol. 1 (2003), pp. 87-133.

[17] F. Filbet, G. Russo, *High order numerical methods for the space non-homogeneous Boltzmann equation*. J. Comput. Phys., 186 (2003), 457–480.

[18] F. Filbet, E. Sonnendrücker, P. Bertrand, *Conservative Numerical schemes for the Vlasov equation*. J. Comput. Phys. 172, (2001) pp. 166–187.

[19] I.M. Gamba, S. H. Tharkabhushaman, *Spectral - Lagrangian based methods applied to computation of Non - Equilibrium Statistical States*. J. Comput. Phys. 228, (2009) pp. 2012–2036.

[20] E.P. Gross P.L. Bathnagar, M. Krook, *A model for collision processes in gases. I. small amplitude processes in charged and neutral one-component systems* Phys. Rev. 94 (1954), pp. 511–525.

[21] T. Homolle, N. Hadjiconstantinou, *A low-variance deviational simulation Monte Carlo for the Boltzmann equation*. J. Comput. Phys., Vol 226 (2007), pp 2341–2358.

[22] T. Homolle, N. Hadjiconstantinou, *Low-variance deviational simulation Monte Carlo*. Phys. Fluids, Vol 19 (2007), 041701.

[23] R. J. LeVeque, Numerical Methods for Conservation Laws, Lectures in Mathematics, Birkhauser Verlag, Basel (1992).

[24] S. Liu, Monte Carlo strategies in scientific computing, Springer, (2004).

[25] R. Loubère, *First steps into ALE INC(ubator). A 2D arbitrary-lagrangian-eulerian code on general polygonal mesh for compressible flows - Version 1.0.0*, Los Alamos National Laboratory Report, LA-UR-04-8840, (2004).

[26] S. Mas-Gallic, *A deterministic particle method for the linearized Boltzmann equation*, Transp. Theory Stat. Phys., Vol. 16, (1987), pp. 885-887.

[27] S. Mas-Gallic, F. Popaud, *Approximation of the transport equation by a weighted particle method*, Transp. Theory Stat. Phys., Vol. 17, (1988), pp. 311-345

[28] L. Mieussens, *Discrete Velocity Model and Implicit Scheme for the BGK Equation of Rarefied Gas Dynamic*, Math. Models Meth. App. Sci., Vol. 10, (2000), 1121–1149.

[29] K. Nanbu, *Direct simulation scheme derived from the Boltzmann equation*, J. Phys. Soc. Japan, vol. 49 (1980), pp. 2042–2049.

[30] W.F. Noh, *Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux.*, J. Comput. Phys. 72, (1987), pp 78-120

[31] A. Palczewski, J. Schneider, A.V. Bobylev, *A consistency result for a discrete-velocity model of the Boltzmann equation*. SIAM J. Numer. Anal. 34, (1997) pp. 18651883.

[32] A. Palczewski, J. Schneider, *Existence, stability, and convergence of solutions of discrete velocity models to the Boltzmann equation*. J. Statist. Phys. 91, (1998) pp. 307326.

[33] L. Pareschi, G. Russo, *Time Relazed Monte Carlo methods for the Boltzmann equation*, SIAM J. Sci. Comput. 23 (2001), pp. 1253–1273.

[34] L. Pareschi, *Hybrid multiscale methods for hyperbolic and kinetic problems*, Esaim Proceedings, Vol. 15, T. Goudon, E. Sonnendrucker & D. Talay Editors (2005), pp.87-120.

[35] S. Pieraccini, G. Puppo, *Implicit-explicit schemes for BGK kinetic equations*, J. Sci. Comp. (2007), pp. 1-28.

[36] D. I. Pullin, *Direct simulation methods for compressible inviscid ideal gas flow*, J. Comput. Phys., 34 (1980), pp. 231–244.

[37] G. Strang, *On the construction and the comparison of difference schemes*. SIAM J. Numer. Anal., (1968), pp. 506–517.

3.3 Interfaces in a finite volume scheme : Enhanced Natural Interface Positioning (ENIP)

Note that our goal in this section is not the description of the Finite Volume with Characteristic Flux scheme and its ultimate details and we refer the reader to the bibliography [210, 211, 212] for this matter. Instead we focus on the interface reconstruction technique that is a part of this scheme and for which we have proposed several improvements described in one article for the two-material case [18] *A totally Eulerian Finite Volume solver for multi-material fluid flows : Enhanced Natural Interface Positioning (ENIP)*, and one preprint for multi-material case [41] *Dealing with more than two materials in FVCF-ENIP method*.

The so-called ENIP technique is nowaday part of the FVCF scheme (Finite Volume with Characteristic Flux) introduced in [210] for simulating single phase compressible flows or multi-phase models without sharp interface capturing. This scheme has been supplemented with the so called NIP method (Natural Interface Positioning), see [212], to deal with multi-material fluid flows with sharp interface capturing. It is a cell centered totally Eulerian scheme, in which material interfaces are represented by a discontinuous piecewise linear curve. A treatment for interface evolution is proposed on Cartesian structured meshes which is locally conservative in mass, momentum and total energy and allow the materials to slide on each others. Discrete conservation laws are written on partial volumes as well as on pure cells, considering the interface in the cell as a moving boundary without any diffusion between materials. A specific data structure called *condensate* is introduced in order to write a finite volume scheme even when the considered volume is made of moving boundaries, i.e. interfaces. This treatment includes an explicit computation of pressure and velocity at interfaces. In [212], 2D results are shown illustrating the capability of the method to deal with perfect sliding, high pressure ratios and high density ratios. This former method however produces non satisfactory results in the context of advection of geometrical shapes especially when dealing with low Mach numbers. Generally speaking most of the advection and reconstruction methods have a tendency to destroy the shape of advected objects due to numerical approximations. This former method behaves similarly, but gives very poor results when advecting geometrical shapes especially when dealing with low Mach number flows. In this work, we have proposed a new method called ENIP (Enhanced NIP) that is an improvement of the NIP method by a more accurate treatment of condensates. In fact both NIP and ENIP are depicted in parallel in Fig.3.6. These pictures present the ideal example of a left-to-right advection of a cubic block of gray material. The condensate in x-direction made of the top part of the block is considered. On the left part of the figure is schematically presented the original method from J.-P. Braeunig *et al* [212] to deal with a condensate. On the right part of the figure we present our improvement of the method. The first four steps are identical to both approach.

(A) : the situation at t^n presents a block of material (gray) which uniquely defines the volume fractions of material in each cell. From the volume fractions an interface is reconstructed using Youngs' method [189], this is the blue segment associated with a unique blue normal.

(B) : a SLIC representation is built. In other words the material is represented by a vertical interface and located on either side of the cell according to the angle of the normal against a vertical line. There is an ambiguity when the interface is aligned with the condensate direction (cell 3 and 4 in Fig. 3.6). In such a case the location of the material on either side is based on roundoff.

(C) : the pieces of material in contact are glued together, as instance on both sides of cell interface j. The condensate is therefore a succession of material layers and interfaces labled with roman

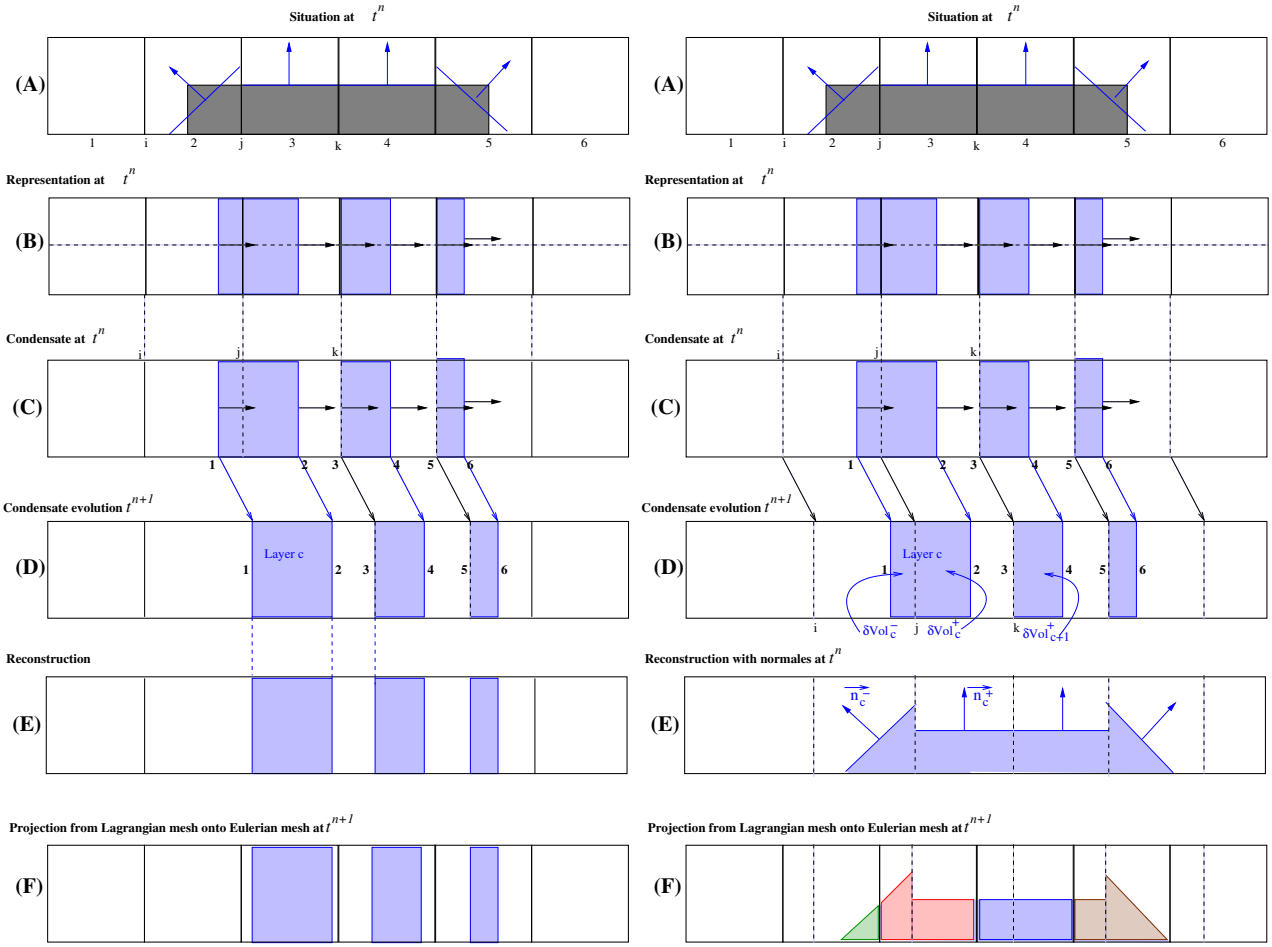


Figure 3.6 – *NIP vs ENIP method — **(A)** Situation at t^n with real materials geometry, interfaces and normals to them. **(B)** Representation of partial volumes at t^n. **(C)** Construction of a condensate at t^n by merging layers of contiguous partial volumes of the same material. **(D)** Evolution of condensate in a Lagrangian fashion during Δt. **(E)** Condensate reconstruction at t^{n+1}. **(F)** Condensate projection/remapping from Lagrangian mesh onto original mesh.*

numbers. Next the numerical scheme computes the material interface normal velocity (black arrows).

(D) : the interfaces/layers move with the previously computed normal velocities during the time step to reach t^{n+1}. For this step ENIP method also displaces the cell interface assuming a piece-wise linear velocity field along the condensate. Consequently the cells are pseudo-Lagrangian ones ; they may compress or expand.

(E) : this reconstruction step serves as retrieving a fair representation of the underlying material locations in the pseudo Lagrangian cells. This step did not exist in NIP. For ENIP we assume the normals to be unmodified during the time step, hence the t^n normals are considered.

(F) : materials are finally remapped back onto the Eulerian cells. While this step is obvious for NIP it is more demanding for ENIP because several polygonal shapes must be computed, see the same colored polygons which intersect to the same Eulerian cell.

Using a very simple example, the advection of a square, an inconsistency in the NIP interface

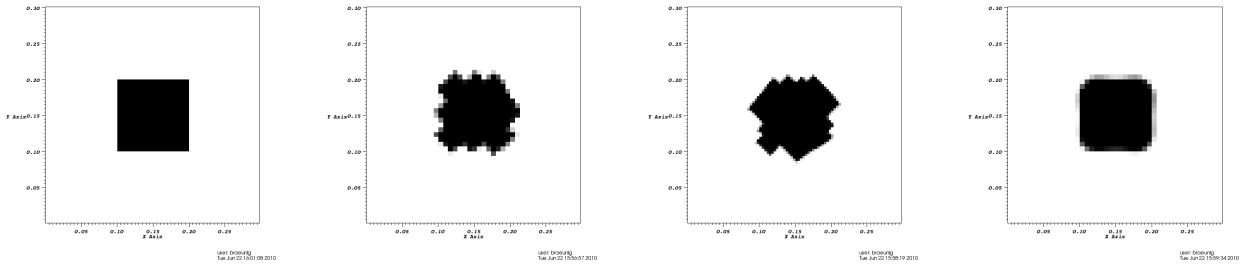


FIGURE 3.7 – *Inconsistency of NIP method and improvement gained by ENIP method — Advection of a square (zoom around the exact position of the initial and final square) — From left to right : exact solution, classical NIP with a* 60×60 *mesh, classical NIP with a* 120×120 *mesh and ENIP with a* 60×60 *mesh.*

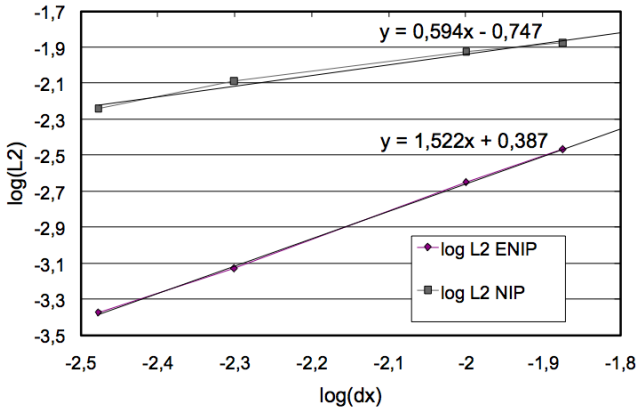


FIGURE 3.8 – *Convergence of ENIP vs NIP for a pure advection problem. The log of the* L_2 *error is displayed as a function of the log of* Δx*.*

reconstruction method is exhibited in Fig.3.7. An initial square $[0.1; 0.1] \times [0.2; 0.2]$ is located into the domain $\Omega = [0 : 0.4] \times [0; 0.6]$. The density into the square is set to $\rho_0(x) = 1$ whereas it is set to $\rho_0(x) = 0$ outside. In the pure advection context with a constant velocity (u, v) this square shape should be perfectly conserved . The exact solution at any point x and any time t is $\rho^{ex}(x, y, t) = \rho_0(x - u\ t, y - v\ t)$. The test consists in advecting the square with the constant velocity field $u = 1$, $v = 3$ up to the time $t = 0.1$ then reversing the advection field by setting $u = -1$, $v = -3$ up to final time $t = 0.2$ so that the final configuration exactly fits the initial one. In Figure 3.7 are shown the exact solution (top-left) and the results obtained with a 60×60 mesh for NIP (top-right) and ENIP (bottom-right). ENIP is visibly able to preserve the shape of the square whereas NIP is not. A mesh refinement of NIP computation (120×120 mesh for the bottom-left panel) does not improve the situation. If the numerical method provides an approximated solution called ρ_i^n in cell i at time t^n then the error in L_α norm is evaluated by ($\alpha = 1, 2$)

$$\varepsilon_\alpha = \frac{\sum_i |\rho_i^n - \rho^{ex}(x_i, t^n)|^\alpha}{\sum_i |\rho^{ex}(x_i, t^n)|^\alpha}. \tag{3.5}$$

The errors for the L_2 norm for successively refined meshes have been computed for both methods and, systematically ENIP over-tops NIP. Moreover in Figure 3.8 we display the log-log scale results for the error showing the improvement gained by ENIP ; indeed the slope which represents a measure of the numerical order of convergence is improved by a factor 2.5 (0.6 for NIP and 1.5 for ENIP).

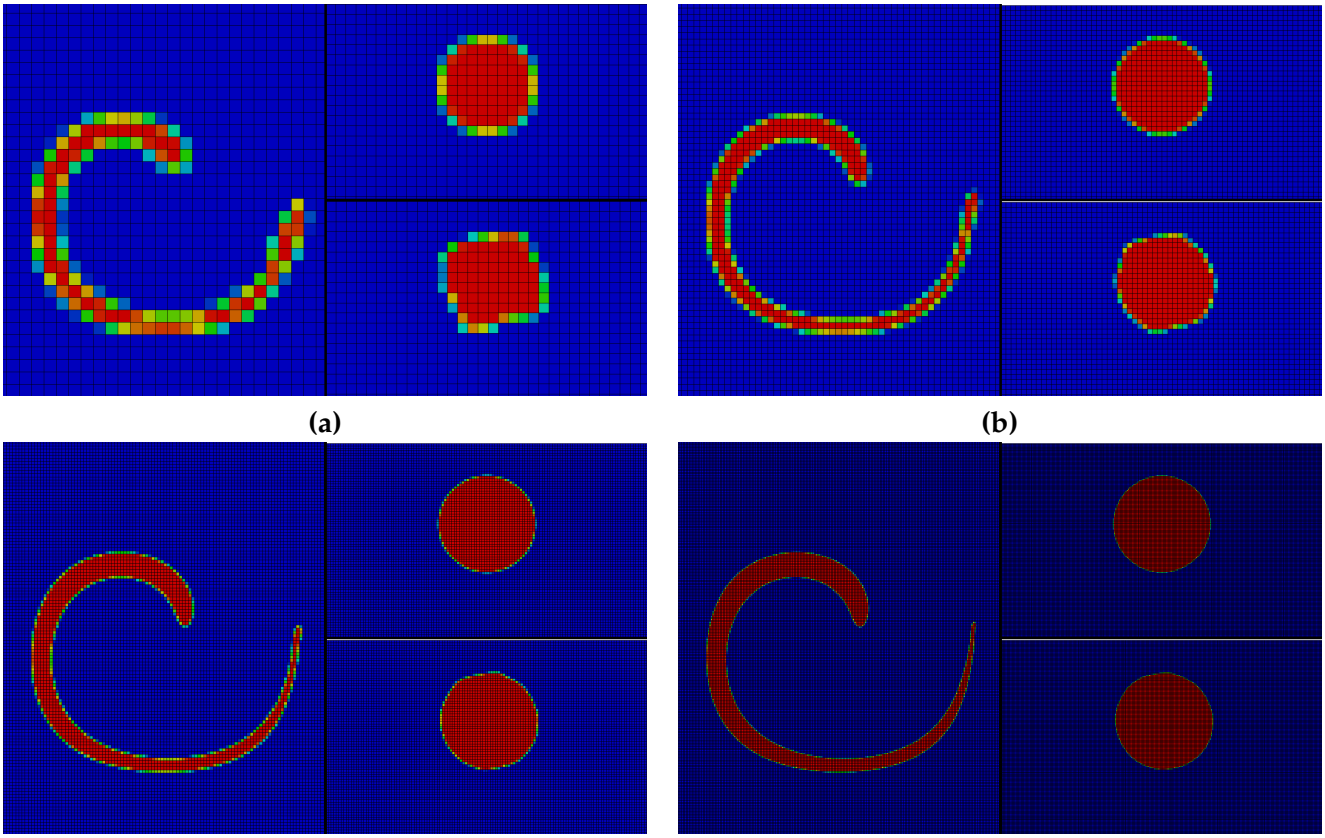


Figure 3.9 – *Disk embedded into a vortex problem — Results of ENIP method for the volume fractions (blue : material 1, red : material 2, any other color refers to a mixed cell).* ***(a)*** *:* 32×32 *mesh,* ***(b)*** *:* 64×64 *mesh,* ***(c)*** *:* 128×128 *mesh,* ***(d)*** *:* 256×256 *mesh. For comparison purposes for each mesh resolution we plot the initial* $t = 0$ *and final* $t = 4$ *times on the top of each other and on their left the time of maximal stretch* $t = 2$*.*

A standard volume tracking test case is the disk embedded into a vortex. It consists of a circle of radius 0.15 centered at $(0.5, 0.75)$. The computational domain is $\Omega = [0,1] \times [0,1]$. The mesh is a regular structured grid made of squares of size $\Delta x \times \Delta y$ with $\Delta x = \Delta y = 1/N$ with N the number of cells both in x and y directions. The incompressible velocity field is given by the streamfunction

$$\Psi = \cos\left(\frac{\pi}{4}t\right)\frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y) \tag{3.6}$$

with the velocity field defined to be $\boldsymbol{U} = (u, v) = (-\frac{\partial \Psi}{\partial y}, \frac{\partial \Psi}{\partial x})$. Due to the periodicity of Ψ, at time $t = 4$, the material configuration should be identical to the condition at time $t = 0$. The simulations were run to a final time of $t = 4.0$ with intermediate results at $t = 2.0$. Several succesively refined meshes are used : $N = 32, 64, 128$ and 256 and the results are plotted in Fig.3.9 for the initial $t = 0$, maximal stretch $t = 2$ and final $t = 4$ times. A two material disk is considered, one material is indexed by 2 in the disk (red color) and the second one is indexed by 1 in the surrounding (blue color). The results of colored volume fractions are shown, any color different from red and blue indicates a mixed cell. Numerical investigations on this test case have shown that the method is only first order accurate. However its extension to deal with more than two materials is almost trivial. In a forthcoming preprint [41] we have tested this approach with the simple onion-skin

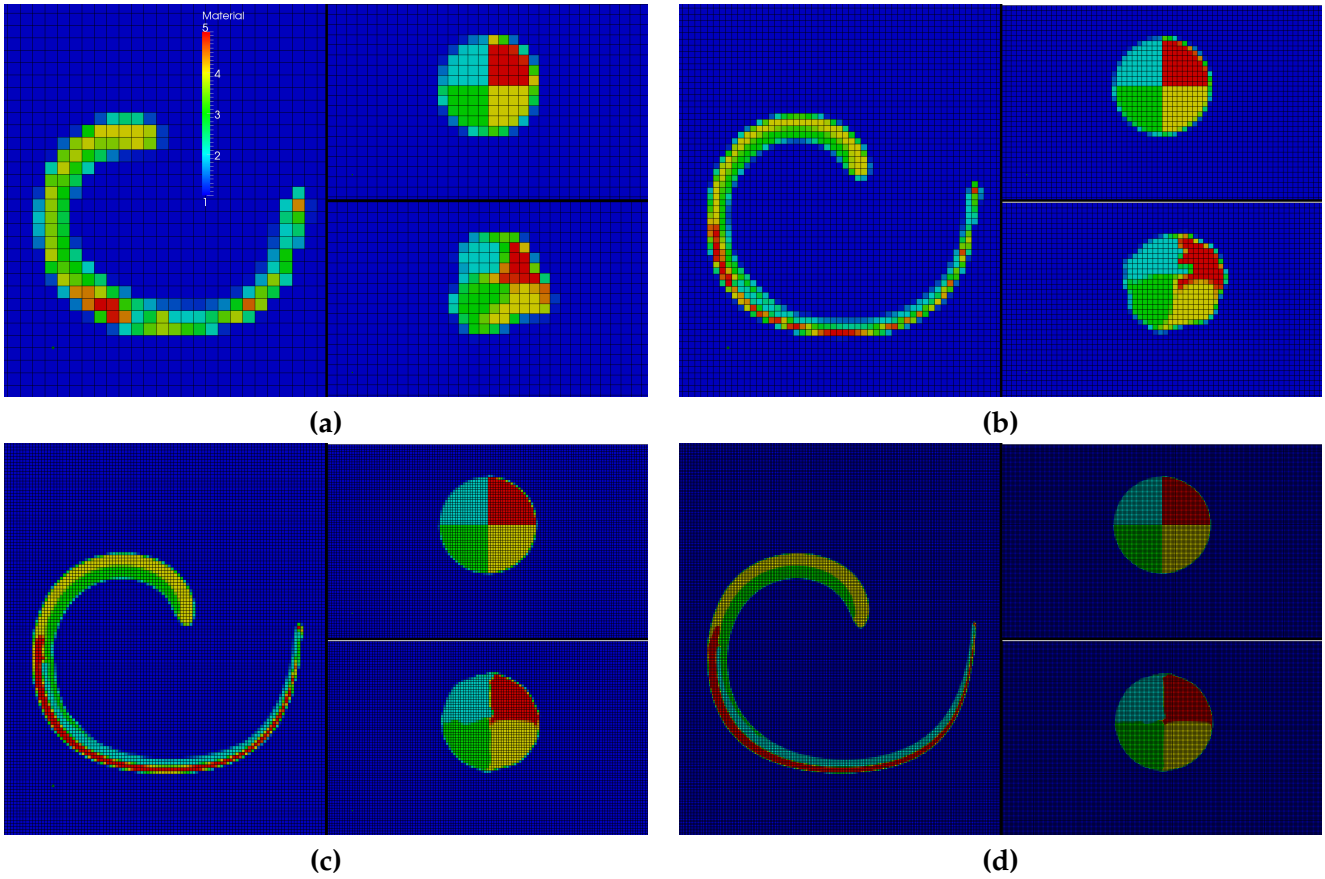


FIGURE 3.10 – *Four material disk embedded into a vortex problem — Results of ENIP method for the volume fractions.* ***(a)*** *:* 32×32 *mesh,* ***(b)*** *:* 64×64 *mesh,* ***(c)*** *:* 128×128 *mesh,* ***(d)*** *:* 256×256 *mesh. For comparison purposes for each mesh resolution we plot the initial* $t = 0$ *and final* $t = 4$ *times on the top of each other and on their left the time of maximal stretch* $t = 2$*.*

interface reconstruction method however more accurate method (MOF [195], Power diagram [17]) can also be considered within this framework. As an illustration we present the volume fractions in Fig. 3.10 for the vortex test case but with a four material disk (see the initial disk for each resolution). The disk material indexes are 2 (light blue), 3 (green), 4 (yellow) and 5 (red) and the surrounding material is labeled by 1 (navy blue). Each cell with a mixed color is indeed a mixed cell. The general shape of the materials is rather well preserved.

European Journal of Mechanics B/Fluids 31 (2012) 1–11

Contents lists available at SciVerse ScienceDirect

European Journal of Mechanics B/Fluids

journal homepage: www.elsevier.com/locate/ejmflu

A totally Eulerian finite volume solver for multi-material fluid flows: Enhanced Natural Interface Positioning (ENIP)

Raphaël Loubère [a], Jean-Philippe Braeunig [b], Jean-Michel Ghidaglia [c,*]

[a] CNRS et Université de Toulouse IMT (Institut de Mathématiques de Toulouse), 31062 Toulouse, France
[b] INRIA Nancy Grand-Est 615 rue du Jardin Botanique 54600 Villers-lès-Nancy and CEA DAM DIF, F-91297 Arpajon, France
[c] CMLA, CNRS et ENS de Cachan, 61 Av. du Président Wilson, Cachan Cedex 94235, France

ARTICLE INFO

Article history:
Received 19 October 2010
Received in revised form
18 July 2011
Accepted 19 July 2011
Available online 30 July 2011

Keywords:
Multi-material fluid flow
Finite volume
Natural Interface Positioning

ABSTRACT

This work concerns the simulation of compressible multi-material fluid flows and follows the method FVCF-NIP described in the former paper (Braeunig et al., 2009 [7]). This cell-centered finite volume method is totally Eulerian since the mesh is not moving and a sharp interface, separating two materials, evolves through the grid. A sliding boundary condition is enforced at the interface, and mass, momentum and total energy are conserved. Although this former method performs well on 1D test cases, the interface reconstruction suffers of poor accuracy in conserving shapes for instance in linear advection. This situation leads to spurious instabilities of the interface. The Enhanced-NIP method presented in the present paper provides a solution to an inconsistency in the former NIP method. This solution strikingly improves the numerical results. It takes advantage of a more consistent description of the interface in the numerical scheme. Results for linear advection and compressible Euler equations for inviscid fluids are presented to assess the benefits of this new method.

© 2011 Published by Elsevier Masson SAS.

1. Introduction

The two-material compressible hydrodynamics equations (Euler equations) are considered in this work. The flow regime is such that molecular viscosity within materials is neglected: materials are supposed to be immiscible and separated by sharp interfaces, with perfect sliding between materials. Each material is characterized by its own equation of state (EOS).

The formalism of finite volume methods is close to the mechanical viewpoint, and generic for different types of physical models. Thus, it is easier to add other physical phenomena like surface tension or turbulent diffusion for instance. The discretization order is limited, but this method is accurate to simulate hydrodynamic shock waves, because of the consistency between numerical treatment and mechanics.

The extension of Eulerian schemes to multi-material fluid flows can be obtained by various techniques. One is to introduce the cell mass fraction c_α of material α and let it evolve according to material velocity. The cell is called pure if a material α satisfies $c_\alpha = 1$ and is called mixed if $c_\alpha \in]0, 1[$. Pure cells filled by material α are calculated in the same manner as for the single material method. Mixed cell evolution is computed using a mixing equation of state that takes into account material mass fractions; see *e.g.* [1]. One drawback of this approach is the numerical diffusion of the interface. It turns out that for some applications, this drawback is not acceptable since the diffusion of one material into another one will correspond to a different physics. For example, the two materials could react when a molecular mixture is formed. Moreover, the mixing equation of state itself may lead to difficulties concerning physics and robustness.

In the case of sharp interface capturing methods, the interface is approximated in a mixed cell by a segment by most authors. However, more complex curves than straight line or more complex theory (see [2] for instance) might be used. A famous method using sharp interface reconstruction is the Eulerian as Lagrange plus Remap finite volume scheme, initiated in [3] and further improved in [4]. It belongs to the family of the so called Volume of Fluid (VOF) methods. The first step of this method is a Lagrangian scheme, resulting in a mesh displacement with material velocity. The second step is a multi-material remapping of Lagrangian mesh onto the original Eulerian mesh, by exchanging volume fluxes between cells related to the Lagrangian motion of cell edges. The new interface position in mixed cells is determined using the partial volumes of the materials and the interface normal vector. The latter is calculated using volume fractions from neighboring cells. Thus the ratio of each material in volume fluxes is deduced from the multi-material remapping. Some methods with the same kind of operator splitting are used for incompressible multi-material fluid flows as in [5]. These methods provide sharp interface between materials and discontinuous quantities in mixed cells, allowing large deformations and transient flows. In this context, the drawback of these Lagrange plus Remap methods is the limited accuracy of the underlying single phase scheme due to diffusion induced by the remapping step. Moreover, more complex physics at material interfaces such as sliding effects, is not possible.

The FVCF scheme (Finite Volume with Characteristic Flux) has been introduced in [6] for simulating single phase compressible flows or multi-phase models without sharp interface capturing. The method described in [7], the so called NIP method (Natural Interface Positioning), is combined to the FVCF method in order to deal with multi-material fluid flows with sharp interface capturing. It is a cell centered totally Eulerian scheme, in which material interfaces are represented by a discontinuous piecewise linear curve. A treatment for interface evolution is proposed on Cartesian structured meshes which is locally conservative in mass, momentum and total energy and allow the materials to slide on each others. Discrete conservation laws are written on partial volumes as well as on pure cells, considering the interface in the cell as a moving boundary without any diffusion between materials. A specific data structure called *condensate* is introduced in order to write a finite volume scheme even when the considered volume is made of moving boundaries, i.e. interfaces. This treatment includes an explicit computation of pressure and velocity at interfaces.

In [7], 2D results are shown illustrating the capability of the method to deal with perfect sliding, high pressure ratios and high density ratios. This former method however produces non satisfactory results in the context of advection of geometrical shapes especially when dealing with low Mach numbers. Generally speaking most of the advection and reconstruction methods have a tendency to destroy the shape of advected objects due to numerical approximations. This former method behaves similarly, but gives very poor results when advecting geometrical shapes especially when dealing with low Mach number flows. In this work, we propose a new method called ENIP (Enhanced NIP) that is an improvement of the NIP method by a more accurate treatment of condensates. Using a very simple example, the advection of a square, an inconsistency in the NIP interface reconstruction method will be exhibited. We will then introduce ENIP that cures this situation. Numerical examples are presented in Section 4 to assess the validity and efficiency of this new approach.

* Corresponding author.
E-mail addresses: raphael.loubere@math.univ-toulouse.fr (R. Loubère), braeunig@math.u-strasbg.fr (J.-P. Braeunig), jmg@cmla.ens-cachan.fr, ejmb@ghidaglia.net (J.-M. Ghidaglia).

0997-7546/$ – see front matter © 2011 Published by Elsevier Masson SAS.
doi:10.1016/j.euromechflu.2011.07.005

2. FVCF-ENIP: Finite Volume Characteristics Flux with Enhanced Natural Interface Positioning technique

2.1. Governing equations

The model addressed in this work is the compressible Euler equations in space dimension d that can be written in a conservative form as follows:

$$\frac{\partial}{\partial t}(\rho) + \mathrm{div}(\rho u) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u) + \mathrm{div}(\rho u \otimes u + pI) = 0, \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \mathrm{div}((\rho E + p)u) = 0, \tag{3}$$

where ρ denotes the density, $u \in \mathbb{R}^d$ the velocity field, p the pressure, $E = e + |u|^2/2$ the specific total energy and e the specific internal energy. An equation of state of the form $EOS(\rho, e, p) = 0$ or $p = p(\rho, e)$ is provided in order to close the system.

Let us consider a generic conservative form with $\boldsymbol{V} = (\rho, \rho u, \rho E)^t$ the unknown vector of conservative variables and flux $\boldsymbol{F}$ is a matrix valued function defined as

$$\begin{aligned} \boldsymbol{F}: \ & \mathbb{R}^{d+2} \longrightarrow \mathbb{R}^{d+2} \times \mathbb{R}^d \\ & \boldsymbol{V} \longmapsto \boldsymbol{F}(\boldsymbol{V}), \end{aligned} \tag{4}$$

for all direction $n \in \mathbb{R}^d$, $\boldsymbol{F}(\boldsymbol{V}) \cdot n$ is given in terms of $\boldsymbol{V}$ by

$$\boldsymbol{F}(\boldsymbol{V}) \cdot n = (\rho\,(u \cdot n)\,,\ \rho u\,(u \cdot n) + pn,\ (\rho E + p)\,(u \cdot n)) \,. \tag{5}$$

The compressible Euler Eqs. (1)–(3) can then be rewritten as

$$\partial_t \boldsymbol{V} + \mathrm{div}\boldsymbol{F}(\boldsymbol{V}) = 0. \tag{6}$$

2.2. FVCF: single material scheme

The FVCF method uses a directional splitting on Cartesian structured meshes. The method is detailed for only one generic direction denoted by x. In d dimensions of space, the algorithm described for direction x has to be replicated d times, one for each direction. However, this directional splitting does not modify the underlying single material scheme FVCF for pure cells. In 2D:

- variables at $t^{n,x}$ are calculated from those at t^n by the x direction step,
- variables at t^{n+1} are calculated from those at $t^{n,x}$ by the y direction step.

$$\mathrm{Vol}_i \frac{\boldsymbol{V}_i^{n,x} - \boldsymbol{V}_i^n}{\Delta t} + A_x\left(\phi_\ell^n + \phi_r^n\right) = 0, \tag{7}$$

$$\mathrm{Vol}_i \frac{\boldsymbol{V}_i^{n+1} - \boldsymbol{V}_i^{n,x}}{\Delta t} + A_y\left(\phi_d^n + \phi_u^n\right) = 0, \tag{8}$$

where the cell volume is Vol_i, the cell face areas are A_x and A_y normal to x and y directions, respectively, up, down, right and left direction fluxes ϕ_d^n, ϕ_u^n, ϕ_r^n, ϕ_ℓ^n are calculated with respect to the outgoing normal direction n_d of cell face Γ_d in direction d using variables at time t^n, i.e.

$$\phi_d^n = \frac{1}{A_d}\int_{\Gamma_d} \boldsymbol{F}(\boldsymbol{V}^n) \cdot n_d \mathrm{d}S. \tag{9}$$

This flux is further approximated using the finite volume scheme FVCF described in [6].

2.3. FVCF-NIP: multi-material scheme

One considers multi-material flows. The subcell model addressed here for the multi-material representation is a cell C of volume Vol_C containing n_m different materials, each of them filling a partial volume Vol_C^k such that

$$\sum_{k=1}^{n_m} \mathrm{Vol}_C^k = \mathrm{Vol}_C. \tag{10}$$

Cell C is referred as pure if $n_m = 1$, and as mixed if $n_m > 1$. The interfaces in mixed cells are approximated by segments separating materials into two partial volumes which are pure on both sides of the interface.

A partial volume cell-centered variable vector $\boldsymbol{V}_k = (\rho_k, \rho_k u_k, \rho_k E_k)^t$ and an equation of state $EOS_k(\rho_k, e_k, p_k) = 0$ are also associated with each material labeled by $k \le n_m$ in the mixed cell.

The FVCF-NIP method uses a directional splitting scheme for the interface evolution without loosing the accuracy of the Eulerian scheme in the bulk of materials. Consequently, this scheme is restricted to the structured Cartesian mesh.

The multi-material extension proposed in [7] considers the finite volume scheme (7)–(8) on each partial volume in a mixed cell. The obtained scheme is conservative by construction and the time step iscomputed considering only the *CFL* stability condition

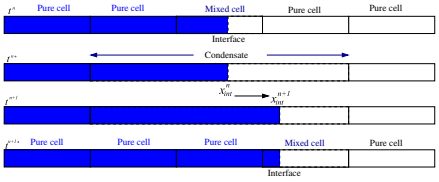


Fig. 1. Sketch of a condensate. Evolution of an interface through a cell edge during one time step. Mixed and pure neighbor cells are merged to obtain the so called condensate at fictitious time t^{n*}. Interface evolution is performed within this condensate from t^{n*} to t^{n+1*}. This condensate is then split back into Eulerian cells.

of the scheme in single material cells.[1] Stability of the mixed cell scheme is the purpose of the reference [8]. The NIP method consists in removing cell edges when this cell contains an interface. Therefore, each partial volume is merged with the neighbor pure cells filled with the same material; see Fig. 1. Variables in these enlarged partial volumes are obtained by writing the conservation laws on the merged volumes:

$$\overline{\mathrm{Vol}_1} = \mathrm{Vol}_1 + \mathrm{Vol}_{pure\ 1}, \tag{11}$$

$$\overline{\mathrm{Vol}_2} = \mathrm{Vol}_2 + \mathrm{Vol}_{pure\ 2}, \tag{12}$$

where Vol_1 and Vol_2 are the partial volumes of materials 1 and 2 in the mixed cell and $\mathrm{Vol}_{pure\ 1}$ and $\mathrm{Vol}_{pure\ 2}$ are the volume of neighbor pure cells of materials 1 and 2. Each volume or partial volume is associated with its state vector $\boldsymbol{V}$ of conserved variables. With this notations, we proceed the following conservative average of variables in merged volumes:

$$\overline{\boldsymbol{V}}_1 = \frac{\mathrm{Vol}_1\,\boldsymbol{V}_1 + \mathrm{Vol}_{pure\ 1}\,\boldsymbol{V}_{pure\ 1}}{\overline{\mathrm{Vol}_1}} \tag{13}$$

$$\overline{\boldsymbol{V}}_2 = \frac{\mathrm{Vol}_2\,\boldsymbol{V}_2 + \mathrm{Vol}_{pure\ 2}\,\boldsymbol{V}_{pure\ 2}}{\overline{\mathrm{Vol}_2}}. \tag{14}$$

This set of cells is associated with its left and right single material fluxes ϕ_ℓ and ϕ_r. Internal cell edges are forgotten, considering only enlarged volumes $\overline{\mathrm{Vol}_1}$ and $\overline{\mathrm{Vol}_2}$ and averaged variables $\overline{\boldsymbol{V}}_1$ and $\overline{\boldsymbol{V}}_2$, separated by an interface; this system is called a condensate.

This numerical strategy consists in condensing neighboring mixed cells in one direction of the Cartesian mesh, in which interfaces are considered as mono dimensional objects, namely, they are considered vertical during x direction step and horizontal during y direction step. A condensate then contains layers of successive different materials that are separated by straight interfaces. The thickness of these layers is calculated through volume conservation. The ordering of layers is given by the 2D description from the previous time step. It is determined thanks to the volume fractions of neighboring cells. The layer evolution is calculated in a Lagrangian fashion which implies that layers can be as thin as partial volumes are small. Once quantities and interface positions inside the condensate are known at time t^{n+1}, they are remapped [9] onto the original Eulerian mesh. Finally, a 2D normal in each mixed cell is computed as described in [4]: the method is based on an approximation of the gradient of the volume fraction function in mixed cells. It provides the normal to material's interface in each cell that is further used to locate materials within mixed cells. The numerical scheme used in a condensate and especially the sliding condition at the interface is presented in detail in [7], so consequently its description is omitted in this work. The emphasis is rather put on the interface reconstruction method.

As shown in [7] this numerical method has several attractive properties such as conservation and perfect sliding of materials for instance. Moreover, Δt is not restricted by small partial volume thanks to a tight control of density and pressure [8], the numerical experiments carried out in [6,7,9] have confirmed the efficiency of such a method for compressible multi-material computation. Indeed, the robustness, the accuracy and the perfect sliding property of the scheme have been verified by comparing solutions with academic test results as well as by testing on violent flows with stiff equations of state, high density ratios, high or low Mach numbers. The accuracy of the scheme to capture shock waves and contact discontinuities with the interface capturing method is very good. Although very promising, the method suffers from the way interfaces are dealt with.

The NIP method consists of the following steps assuming the condensate is in the x direction:

- *Situation* Fig. 2(A). It corresponds to the true situation at t^n with real material geometry, interfaces and normals to them.
- *Representation* Fig. 2(B). The representation step can be seen as the way of determining on which side (left or right) of the mixed cell the material is to be put. This is done by comparing the direction of the interface normal at time t^n with the vertical direction.
- *Condensate construction* Fig. 2(C). The construction of the condensate consists in discarding any cell edges in the mixed cells considered. Then the partial volumes of the same contiguous materials are glued together into the so called condensate layers. For instance, the dark materials of cells 2 and 3 are merged into one stand-alone layer.
- *Condensate evolution* Fig. 2(D). The condensate layer evolution is computed from t^n to t^{n+1} thanks to the numerical scheme developed in [7]. At each interface a velocity is assigned and, consequently, a new position of each layer within the condensate is determined in a Lagrangian manner. Any conserved variable is computed accordingly.
- *Reconstruction* Fig. 2(E). This phase consists in "guessing" the shape of each layer in the condensate before remapping.
- *Projection* Fig. 2(F). The projection step consists in remapping the shapes obtained from the reconstruction phase onto the Eulerian grid. This step produces updated partial volumes in mixed cells. Volume fractions are deduced.

The sliding at the interface is enforced by solving a Riemann problem, still in 1D accordingly with Godunov like methods, but in the direction of the normal to the interface, in such a way the tangential velocity and the tangential pressure stress are free at the interface. This 1D solution in the interface normal direction is then projected in 2D on the axis corresponding to the current directional splitting phase, i.e. formulas that give the 1D interface pressure and velocity in function of the 2D interface normal vector, the 2D velocities and the pressure. All details can be found in the former paper [7].

When all mixed cells in the domain are treated for direction x, the interface normals are computed using the updated volume fractions. This concludes the system evolution in direction x, as we are back to a similar situation as the one described in Fig. 2(A).

In the case where the normal is almost vertical, positioning the material on either side of the cell might be incorrect. Furthermore, the reconstruction phase is here clearly inconsistent: the interfaces are initially horizontal in cell 3 and 4 (Fig. 2(A)), while in the Reconstruction Fig. 2(E) and in the Projection Fig. 2(F) phase interfaces are set vertical for any initial geometry. This situation of a horizontal interface is the worst case, but it illustrates the lack of geometrical consistency of NIP. This inaccurate reconstruction step leads to a lack of accuracy of the volume fractions obtained after the remapping step. Ultimately, it impacts the whole numerical method in any advection process, as it can be seen in Fig. 5 with the diagonal advection of a square test.

[1] Without such a special treatment the time step would be constrained by the smallest partial volume, which could be arbitrarily small.

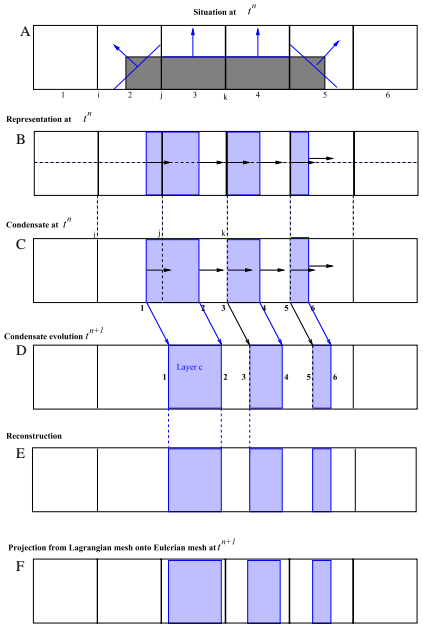


Fig. 2. NIP method — (A) Situation at t^n with real material geometry, interfaces and normals to them. (B) Representation of partial volumes at t^n. (C) Construction of a condensate at t^n by merging layers of contiguous partial volumes of the same material. (D) Evolution of condensate in a Lagrangian fashion during Δt. (E) Condensate reconstruction at t^{n+1}. (F) Condensate projection/remapping from the Lagrangian mesh onto original mesh.

2.4. FVCF-ENIP

The main idea of the new interface reconstruction method ENIP emanates from the following remarks:

1. At time t^n any interface normal in mixed cell i denoted by $\vec{n}_i$ is known. It is used to locate the partial volumes within cell i when the condensate is constructed (phase (B) and (C) of Fig. 2). However, $\vec{n}_i$ is never taken into account in the reconstruction and projection phases (E) and (F) from the same figure.

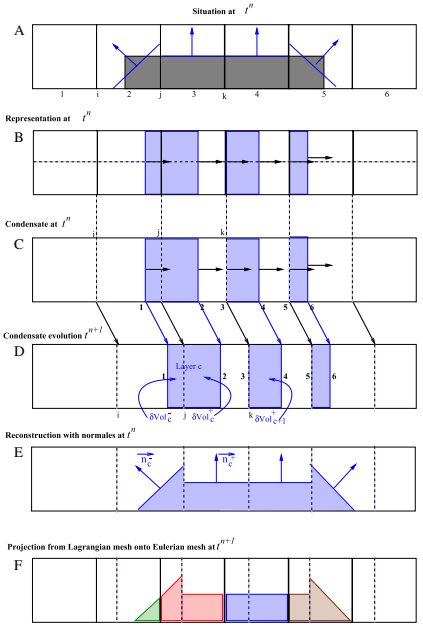


Fig. 3. The ENIP method – (A) Situation at t^n.(B) Representation at t^n. (C) Construction of a condensate at t^n by merging of mixed cells leading to layers of contiguous pieces of the same material. (D) Evolution of condensate in a Lagrangian fashion during Δt. Determine layer compression rates $\theta_c^{\pm}$ through the evolution of Lagrangian cells. (E) Condensate reconstruction at t^{n+1} using interface normals defined at t^n. (F) Condensate projection/remapping from the Lagrangian mesh onto the Eulerian mesh.

2. Any layer of the condensate evolves as a Lagrangian object in the original method. Consequently, the cell faces could evolve in an almost Lagrangian manner within this condensate. This makes it possible to conserve the initial geometry of partial volumes during this Lagrangian motion.

Therefore, ENIP modifies several steps of NIP as described in Fig. 3. Once a patch of neighboring mixed cells in x direction[2] is agglomerated. Then the same five steps as for the NIP method are performed. The first two steps are kept unmodified. The last three are modified as described in the following.

2.4.1. Lagrangian condensate evolution step

Cell interface Lagrangian velocity. After the condensate at t^n is constructed, each layer labeled c is located thanks to the left and right interface position, respectively, called x_c^-, x_c^+. The numerical scheme provides the layer evolution, and as a by-product, the velocities of these interface positions, u_c^-, u_c^+ are given by

$$x_c^{-,n+1} = x_c^- + \Delta t\, u_c^-, \qquad x_c^{+,n+1} = x_c^+ + \Delta t\, u_c^+. \tag{15}$$

We make the following fundamental linear displacement assumption: *The velocity linearly varies within any layer*; see Fig. 4 for a sketch. This assumption implies that any point $x_i \in [x_c^-; x_c^+]$ characterized by its 1D barycentric coordinates

$$\lambda_i^- = \frac{x_c^+ - x_i}{x_c^+ - x_c^-}, \qquad \lambda_i^+ = \frac{x_i - x_c^-}{x_c^+ - x_c^-}, \tag{16}$$

moves to location

$$x_i^{n+1} = \lambda_i^- x_c^{-,n+1} + \lambda_i^+ x_c^{+,n+1} = x_i + \Delta t\left(\lambda_i^- u_c^- + \lambda_i^+ u_c^+\right). \tag{17}$$

Then the point velocity is naturally set to $u_i = \lambda_i^- u_c^- + \lambda_i^+ u_c^+$. Using this previous formula, one can associate a "Lagrangian" velocity to

[2] The y direction is treated likewise.

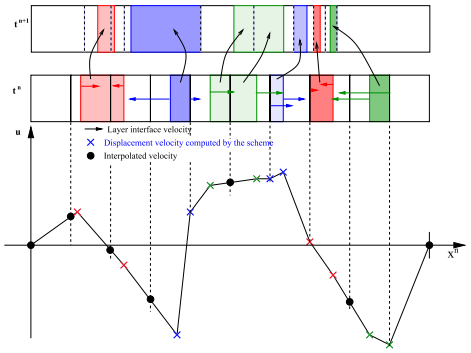


Fig. 4. Sketch of linear displacement assumption – displacement velocity varies linearly between the layer interface velocities ($\times$ in color) computed by the numerical scheme. The cell interface velocity ($\bullet$) is interpolated. The two top rows represent the evolution of a condensate in the x direction from t^n to t^{n+1}.

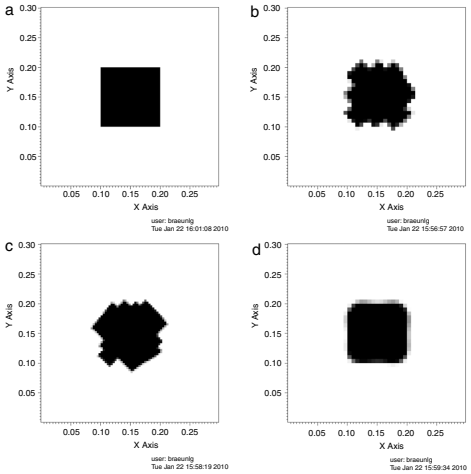


Fig. 5. Advection of a square (zoom around the exact position of the initial and final squares) – (a) exact solution – (b) classical NIP with a 60×60 mesh – (c) classical NIP with a 120×120 mesh – (d) ENIP with a 60×60 mesh.

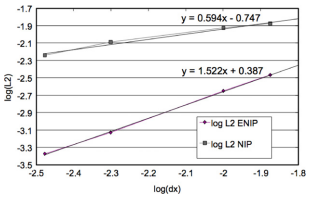


Fig. 6. Convergence of ENIP vs NIP for a pure advection problem. The log of the L_2 error is displayed as a function of the log of Δx.

any cell interface. For instance, in Fig. 3-(C) cell interface located at x_i^n moves to the position $x_i^{n+1} = x_i^n + \Delta t\, u_i$ with u_i being the linear combination between u_c^- and u_c^+ *via* the barycentric coordinates of point x_i in $[x_c^-; x_c^+]$. With the same formula one gets $x_{i+1}^{n+1} = x_{i+1}^n + \Delta t\, u_{c+1}^-$ in the next layer as $x_{i+1}^n \equiv x_{c+1}^-$.

2.4.1.1. Compression/expansion rates.

The global rate of compression/expansion in layer c during Δt is given by

$$\theta_c = \frac{x_c^{+,n+1} - x_c^{-,n+1}}{x_c^+ - x_c^-} = 1 + \Delta t \frac{u_c^+ - u_c^-}{x_c^+ - x_c^-}. \tag{18}$$

The linearity assumption provides a simple way to determine the rates of compression/expansion at left/right of a point $x_i \in [x_c^-; x_c^+]$

$$\theta_c^- = \frac{x_i^{n+1} - x_c^{-,n+1}}{x_c^+ - x_c^-}, \qquad \theta_c^+ = \frac{x_c^{+,n+1} - x_i^{n+1}}{x_c^+ - x_c^-}, \tag{19}$$

that fulfill $\theta_c^- + \theta_c^+ = \theta_c$. Moreover, the substitution of x_i^{n+1} in the previous equations yields

$$\theta_c^- = \frac{x_i - x_c^-}{x_c^+ - x_c -} + \Delta t \frac{u_i - u_c^-}{x_c^+ - x_c^-} = \lambda_i^+ + \Delta t \frac{u_i - u_c^-}{x_c^+ - x_c^-}, \tag{20}$$

where $u_i - u_c^- = (\lambda_i^- u_c^- + \lambda_i^+ u_c^+) - u_c^- = \lambda_i^+ (u_c^+ - u_c^-)$, therefore the compression rates simply write

$$\theta_c^- = \lambda_i^+ \left(1 + \Delta t \frac{u_c^+ - u_c^-}{x_c^+ - x_c^-}\right) = \lambda_i^+ \theta_c, \tag{21}$$

$$\theta_c^+ = \lambda_i^- \left(1 + \Delta t \frac{u_c^+ - u_c^-}{x_c^+ - x_c^-}\right) = \lambda_i^- \theta_c. \tag{22}$$

Each θ_c^+ or θ_c^- is associated to a unique Eulerian cell; as instance in Fig. 3, θ_c^- is associated to cell 2, θ_c^+ to cell 3, θ_{c+1}^+ to cell 4 and so on. Therefore, $\theta_c^{\pm}$ provides *de facto* the compression/expansion of the partial volume originating from its associated Eulerian cell motion. Furthermore, as any Eulerian mixed cell i possesses a unique normal denoted by $\vec{n}_i$, this last is associated to the corresponding partial volume $\theta_c^{\pm}$; this normal is consequently labeled $\vec{n}_c^{\pm}$. These rates are then used to reconstruct the material topology into the Lagrangian cell.

2.4.2. Reconstruction step

The Lagrangian cell i at t^{n+1} the interfaces of which moved as

$$x_i^{n+1} = x_i + \Delta t\, u_i, \qquad x_{i+1}^{n+1} = x_{i+1} + \Delta t\, u_{i+1}, \tag{23}$$

changed its volume as

$$\theta_i = \frac{\mathrm{Vol}_i^{n+1}}{\mathrm{Vol}_i} = \frac{x_{i+1}^{n+1} - x_i^{n+1}}{x_{i+1} - x_i} = 1 + \Delta t \frac{u_{i+1} - u_i}{x_{i+1} - x_i}. \tag{24}$$

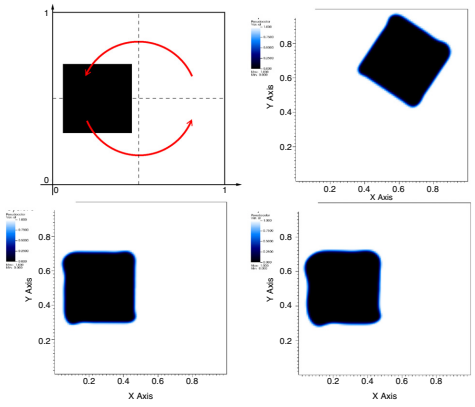


Fig. 7. Top-left: Sketch of the rigid rotation of a square – Top-right: After 5/8 of the full rotation – Bottom-left: After one full rotation – Bottom-right: After three full rotations.

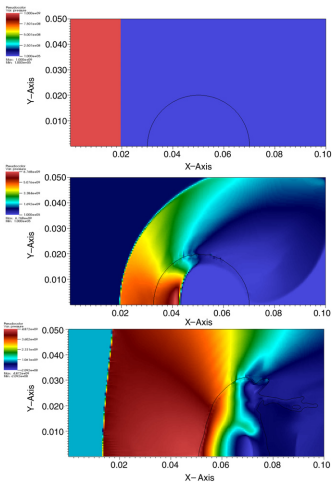


Fig. 8. The pictures represent the pressure profile of the air and of the water droplet, where the water/air interface is the black curve. Times of the computation from top to bottom are $t = 0$, $t = 3$ µs and final time $t = 14.8$ µs.

The velocity u_i depends on u_{c-1}^-, u_{c-1}^+ and u_{i+1} depends on u_c^-, u_c^+. Moreover $u_{c-1}^+ \equiv u_c^-$ by definition.

The second fundamental assumption states that the interface normals $\vec{n}_c^{\pm}$ do not change their direction during their Lagrangian evolution. The goal is to locate the partial volume into the Lagrangian cell at t^{n+1} and construct the linear interface, knowing its normal $\vec{n}_c^{\pm}$. Necessarily this partial volume is either in contact with cell interface x_i (superscript $+$) or x_{i+1} (superscript $-$). Its volume at t^{n+1} is given by

$$\mathrm{Vol}_c^{\pm,n+1} = \mathrm{Vol}_c^{\pm}\, \theta_c^{\mp} = \mathrm{Vol}_c^{\pm} + \Delta t\, \lambda_c^{\mp}(u_c^+ - u_c^-). \tag{25}$$

If $\mathrm{Vol}_c^{\pm,n+1} \leq \mathrm{Vol}_i^{n+1}$ then there exists a unique line oriented by the normal $\vec{n}_c^{\pm}$ and separating the cell volume into two sub-volumes $\mathrm{Vol}_{c-1}^{\pm,n+1}$ and $(\mathrm{Vol}_i^{n+1} - \mathrm{Vol}_c^{\pm,n+1})$, respectively, by the PLIC ("*Piecewise Linear Interface Construction*" [4]) method. As the displacement velocity $u(x)$ is supposed to be piecewise linear (by the first assumption, see Fig. 4), then, if $x_i < x_c^- < x_{i+1}$ one deduces $x_i^{n+1} < x_c^{-,n+1} < x_{i+1}^{n+1}$. Therefore, the sub-volume at t^{n+1} is strictly included into the Lagrangian cell volume Vol_i^{n+1}. This phase is depicted in Fig. 3(E).

2.4.3. Projection step

The projection step performs the exact intersection between the Lagrangian condensate obtained after the reconstruction step in Fig. 3(E) and the Eulerian mesh (bold line squares in Fig. 3-(A)). This step is depicted in Fig. 3-(F). The exact intersection consists in projecting each partial volume that is accurately located in the condensate, onto some Eulerian fixed cell(s). For instance in Fig. 3-(F), the first partial volume is projected onto Eulerian cells 2 (green cell) and 3 (red cell). Contrarily, the last partial volume is totally projected into Eulerian cell 5 (brown cell). This projection provides the quantity of material per Eulerian cell, or, equivalently its volume fraction. Once the volume fractions in the mixed cells are updated through the evolution of the condensates, interface normals are computed using the same technique as in the original NIP method.

3. Numerical results

In this section, we present a set of test cases to assess the efficiency of the approach described in the previous sections. First, one validates the technique on pure advection test cases that often present excessive smearing of interfaces due to the numerical inaccuracy embedded into the scheme. A square shaped object is advected with constant velocity in a diagonal direction in a first test, then into a rotating flow. Finally a hydrodynamics test case is presented.

3.1. Advection context

An initial square $[0.1; 0.1] \times [0.2; 0.2]$ is located in the domain $\Omega = [0; 0.4] \times [0; 0.6]$. The density in the square is set to $\rho_0(x) = 1$ whereas it is set to $\rho_0(x) = 0$ outside. In the pure advection context, this square shape should be perfectly conserved through the equation

$$\frac{\partial}{\partial t}\rho + u\frac{\partial}{\partial x}\rho + v\frac{\partial}{\partial x}\rho = 0, \tag{26}$$

where (u, v) is a constant velocity field. The exact solution at any point x and any time t is $\rho^{ex}(x, y, t) = \rho_0(x - u\,t, y - v\,t)$. If the numerical method provides an approximated solution called ρ_i^n in cell i at time t^n then the error in l_α norm is evaluated by ($\alpha = 1, 2$)

$$\varepsilon_\alpha = \frac{\sum_i |\rho_i^n - \rho^{ex}(x_i, t^n)|^\alpha}{\sum_i |\rho^{ex}(x_i, t^n)|^\alpha}. \tag{27}$$

The first test consists in advecting the square with the constant velocity field $u = 1$, $v = 3$ up to the time $t = 0.1$ then reversing the advection field by setting $u = -1$, $v = -3$ up to final time $t = 0.2$ so that the final configuration exactly fits the initial one. Any method (NIP and ENIP included) introduces some error that we intend to measure with this test. In Fig. 5, the exact solution (panel (a)), and the results obtained with a 60×60 mesh for NIP (panel (b)) and ENIP (panel (d)) are shown. ENIP is visibly able to preserve the shape of the square whereas NIP is not. A mesh refinement of NIP computation (120×120 mesh on the panel (c)) does not improve the situation. In Table 1, we gather the errors for the L_1, L_2 norms for successively refined meshes for the NIP and the proposed ENIP method on this advection problem. Systematically ENIP over-tops NIP. The convergence of the ENIP method as the grid refined is shown in Fig. 6; the log–log scale results for the error in L_2 norm for both methods show the improvement obtained by ENIP; indeed the slope represents a measure of the numerical order of convergence. ENIP has improved it by a factor 2.5 (0.6 for NIP and 1.5 for ENIP).

The next test consists in the rigid rotation of a square $[0.06; 0.46] \times [0.3; 0.7]$ (density 1) into the unit square domain; see the top-left panel of Fig. 7. A 100×100 uniform mesh is considered and the rotation is given by the velocity field

$$u = -100(y - 0.5), \qquad v = 100(x - 0.5).$$

In Fig. 7, we display the density after 5/8 of the full rotation, after one and three rotations. The square shape is almost preserved. Contrarily, the classical NIP method would totally lose the shape after one rotation.

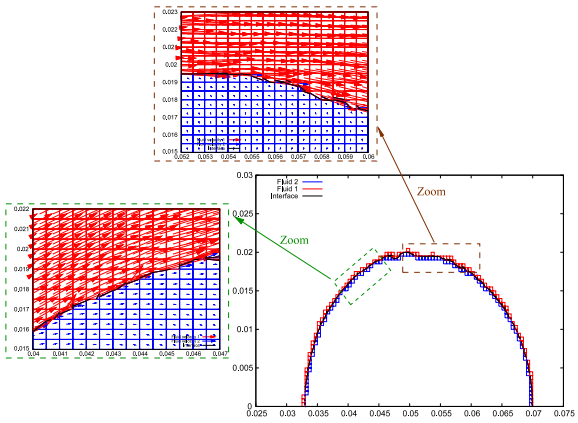


Fig. 9. The pictures above represent the velocity field with the same scale for the arrows length for water (blue) and air (red). The interface is reconstructed with the position of the materials in the mixed cells as it is computed by the method. Two zooms are presented where two velocity vectors are drawn in a cell (only one in single material cells), centered on partial volumes of materials, to show the sliding at the interface at time 3 μs.

Table 1
Error in L_1, L_2 norms for the advection problem — NIP versus ENIP methods.

$\Delta x = \Delta y$	L_1 NIP	L_1 ENIP	L_2NIP	L_2ENIP
0.02	3.652	0.196	2.575	0.079
0.0133	0.389	0.165	0.318	0.081
0.01	0.339	0.111	0.284	0.053
0.005	0.221	0.042	0.195	0.017
0.0033	0.155	0.025	0.138	0.010

Table 2
Initial state for the shock-droplet interaction with equations of state of the form $p = (\gamma - 1)\rho e - \Pi$, with e the internal energy.

Quantity	Post-shock air	Air at rest	Water droplet at rest
Density ρ (kg/m^3)	7.73549	1.29	1000
Pressure p (Pa)	10^9	10^5	10^5
Velocity x (m/s)	25415.657	0	0
Velocity y (m/s)	0	0	0
γ	1.4	1.4	7
Π (Pa)	0	0	2.1×10^9

3.2. Sliding context

3.2.1. Shock-droplet interaction

In this test is considered a 2D water droplet surrounded by air at atmospheric conditions. A very fast shock wave is traveling in the air (Mach $\approx$ 77) from the left to the right hand side and will interact with the droplet. The shock wave goes even faster in the air than in the water at initial state. The initial planar shock wave then curves around the water droplet. In this flow will appear a very strong shear between the air and the water droplet, since the density ratio between the gas and the liquid is here 1.29/1000. This test is presented to emphasize the need of the sliding condition at an interface between fluids of such different densities to properly model the problem in the context of compressible Euler equations, meaning in the context of inviscid flows. The domain is $[0; 0] \times [0.1; 0.05]$ m^2 is corresponding to half the geometry of the domain and of the water droplet of radius 0.02 m. The mesh is 200×100 cells. The initial conditions are given in Table 2, the initial geometry is given in Fig. 8 and results in Fig. 9.

This test shows the robustness of the method with a stiff equation of state for the water and a high density ratio between the materials. The shear at the interface is pictured in Fig. 9.

3.2.2. Free fall of a block of water

We run an idealized 2D test case that corresponds to the free drop of a liquid rectangle within a 2D rectangular tank filled with gas [9]. This context is inspired by the problem of sloshing that may appear in the tanks of Liquid-Natural-Gas (LNG) carriers. The study focuses on the ability for the numerical simulations to take a proper account of physics that is of major importance during the liquid impact such as the escape of the gas underneath and its compression. A strong sliding process occurs between the compressed gas and the falling liquid. The ability of the method to properly deal with sliding conditions at the interface has a major effect on the final numerical compression and the shape of the trapped air. This has ultimately a strong influence on the impact pressure.

The test case consists in a domain $\Omega = [0.0; 0.0] \times [10\text{ m}; 15\text{ m}]$ filled with air. The liquid is initially at rest in the rectangle $[0; 2] \times [5; 10]$ and is falling under the gravity that is pointing downward with magnitude $g = 9.81$ m s^{-2}. A free fall of the liquid into vacuum would impact at $t_{\text{impact}} = 0.64$ s; however due to the presence of the gas this theoretical value is not correct for our simulation however some critical phenomena still occur in the vicinity of this time. As found at around t_{impact}, a pocket of gas is trapped under the falling liquid and this strongly influences the numerical impact pressure by decelerating and damping the free fall of the liquid. Therefore, a sliding condition at the interface should qualitatively improve the numerical results. One considers a mesh made of 100×150 uniform cells on the domain. One shows the results for NIP and ENIP at time $t = 0.66$ s Fig. 10(a)–(b) and $t = 0.69$ s in Fig. 10(c)–(d). The classical NIP method was already able to deal with such sliding effects. However, the interface reconstruction method employed is not sufficiently stable to be free of oscillation that one suspects to be only a numerical artifact (see panels (a–c)). Contrarily, the new reconstruction method ENIP on this very same test case is able to produce a smooth interface that permits to obtain a more realistic simulation. Indeed, this simulation prominently displays the fact that the "bubbling" effects of NIP is of pure numerical origin and that ENIP cures this drawback. The velocity field is shown in Fig. 10 for the original method at times $t = 0.66$ s and $t = 0.69$ s, respectively, in Fig. 10(a) and (c). The proposed method results are displayed in Fig. 10(b) and (d). Original NIP method results at $t = 0.66$ s in (a) and the ENIP method in (b) are relatively close although differences can be seen at the corner interface. NIP results are less smooth. Moreover, below the corner the vector field of air with natural tendency to escape the falling water block is not as continuous as for the ENIP results. This leads for the NIP method to an earlier trapping of the gas pocket under the falling block of water, because of the NIP numerical interface instabilities that are erased with the ENIP method. This effect is emphasized at $t = 0.69$ s in (c). At this time the original NIP method traps some air almost at rest underneath the block of falling water. ENIP results in (d) show that the air should continue to escape.

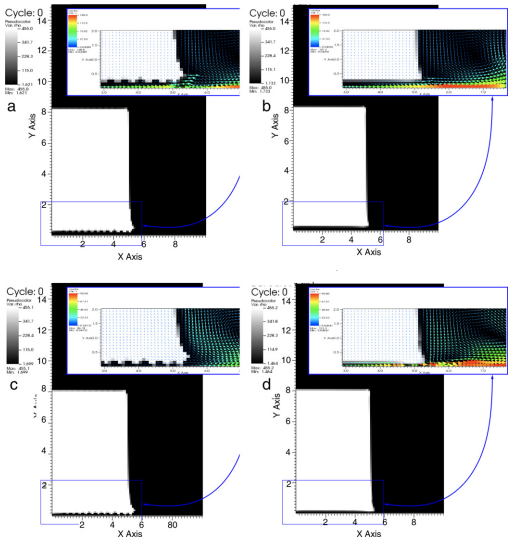


Fig. 10. Free drop of a liquid rectangle within a 2D rectangular tank filled with gas. Zoom on the velocity field (vectors). The color is scaled according to the magnitude of the velocity field. Original NIP results at $t = 0.66$ s in (a) and the ENIP method at the same time (b) are relatively close although differences can be seen at the corner interface. Contrarily, at $t = 0.69$ s in (c) the original NIP method traps some air underneath the block of falling water. At this time the air is still escaping under the falling block as seen for the ENIP results in (d).

4. Conclusion and perspectives

This paper deals with the improvement of the so-called NIP (Natural Interface Positioning) method. The NIP method, described in [7] is an add-on to the FVCF method in order to treat multi-material fluid flows, uses the concept of condensate. A condensate is the association of contiguous mixed cells in either x or y direction. They are further treated as an entity to make possible the treatment of each mixed cell taken individually. NIP is the method based on the following steps: *Representation*, *Condensate construction*, *Condensate evolution*, *Reconstruction*, and *Projection*. The present paper points the weakness of the NIP method in pure advection context and, consequently, in a full multi-material hydrodynamics one. An Enhanced NIP method is proposed (ENIP). It modifies several of the previous listed steps. More precisely the condensate is assumed to evolve in an almost-Lagrangian fashion. The reconstruction step assumes that the condensate keeps the same form modulo some expansion/compression that the numerical scheme already provides. So the displacement of the condensate is performed either with the true computed velocity or with an interpolation of it. *In fine*, the condensate preserves its topology contrarily to the original NIP method for which the condensate has no recollection of its shape from the beginning of the time step.

The capability of the full numerical method is now dramatically improved as seen in advection test cases (advection and rigid rotation of a square). Moreover, we ran ENIP on difficult multi-material hydrodynamics tests. First, we have illustrated the sliding at the interface capability of the ENIP method and its robustness on a shock-droplet interaction test, with a high Mach number, a high density ratio and with very different equations of state between water and air. Second, by simulating the free drop of a liquid rectangle within a 2D rectangular tank filled with gas in the context of sloshing that may appear in the tanks of Liquid-Natural-Gas carrier (see [9]). The accuracy, stability and robustness of the ENIP method is clearly seen especially at the time some air is trapped under the water. In the near future we plan to investigate the evolution of this method to the case of mixed cells with more than two materials. In this case, the only difficulty lies in the positioning of the different materials in the cell, but their evolution within the condensate follows exactly the same algorithm ENIP with no modification of the numerical scheme. The MPI parallelization of the code is already achieved; see [10]. The method has been extended in 3D by Chauveheid [11].

References

[1] D.M. Anderson, G.B. McFadden, A.A. Wheeler, A diffuse-interface methods in fluid mechanics, Ann. Rev. Fluid Mech. 30 (1998) 139–165.
[2] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (2008) 5361–5384.
[3] W.F. Noh, P. Woodward, SLIC Simple Line Interface Caculation, in: Lectures notes in Physics, 59, Editions Springer, Berlin, 1976.
[4] D.L. Youngs, Time-Dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, 1982, pp. 273–285.
[5] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multifluid flows with SURFER, J. Comput. Phys. 113 (1) (1994) 134–147.
[6] J.-M. Ghidaglia, A. Kumbaro, G. Le Coq, On the numerical solution to two fluid models via a cell centered finite volume method, Eur. J. Mech. B/Fluids 20 (6) (2001) 841–867.
[7] J.P. Braeunig, B. Desjardins, J.-M Ghidaglia, A totally Eulerian finite volume solver for multi-material fluid flows, Eur. J. Mech. B/Fluids 28 (4) (2009) 475–485.
[8] J.P. Braeunig, An algorithm to control the pressure evolution for the FVCF-NIP method for compressible multi-material fluid flows, Int. J. Finite 7 (2010) 101–132.
[9] J.P. Braeunig, L. Brosset, F. Dias, J.-M Ghidaglia, Phenomenological study of liquid impacts through 2D compressible two-fluid numerical simulations, in: 19th Int. Offshore and Polar Eng. Conf. Osaka, Japan, ISOPE, 2009.
[10] M. Peybernes, J.-P. Braeunig, O. Roussel, J.-M. Ghidaglia, Parallelization of a pure Eulerian finite volume solver for multi-material fluid flows, in: Conference MULTIMAT 2011, Arcachon, September 2011.
[11] D. Chauveheid, Simulation d'instabilités hydrodynamiques dans les écoulements compressibles: étude numérique et application d'une méthode volumes finis avec reconstruction d'interfaces, Ph.D. Thesis (in French), Ecole normale supérieure de Cachan, to be defended in 2012.

Dealing with more than two materials in the FVCF-ENIP method

Bruno Blais[a], Jean-Philippe Braeunig[b], Daniel Chauveheid[c], Jean-Michel Ghidaglia[d], Raphaël Loubère[e,]

[a] *URPEI, Chemical Engineerring Departement, Ecole Polytechnique de Montréal, P. O. Box 6079, Station CV, Montréal, QC, H3C 3A7, Canada*
[b] *CEA-DAM DIF, Arpajon Cedex, 91297, France*
[c] *CMLA, ENS de Cachan 61 Av. du Président Wilson, Cachan Cedex 94235, France CEA-DAM DIF, Arpajon Cedex, 91297, France*
[d] *CMLA, ENS de Cachan 61 Av. du Président Wilson, Cachan Cedex 94235, France*
[e] *CNRS et Université de Toulouse, IMT (Institut de Mathématiques de Toulouse), Toulouse cedex, 31062, France*

Abstract

This work concerns the simulation of compressible multimaterial fluid flows and follows up the method Finite Volume with Characteristics Flux for two materials described in paper "A totally Eulerian finite volume solver for multi-material fluid flows : Enhanced Natural Interface Positioning (ENIP)", European Journal of Mechanics B/Fluids, vol.31 , No4, pp. 1-11 (2012). The interface reconstruction method was designed to deal with only two materials. In this paper we present the generalisation of the method to more than two materials. The design principles remain the same as for the two material method. Nevertheless some specifics treatments have been added, like an automatic order of treatment of materials using material centers of mass in the so-called condensate. Interestingly the method can accept any interface reconstruction method. For simplicity purposes, the crude onion skin approach with unique material normal has been chosen. The entire scheme has been tested on the four material disk embedded into a reversible incompressible velocity field test case for which numerical error can be computed. Then several more demanding hydrodynamics impacts of droplet on thin film have been simulated to show the ability of the method to qualitatively capture three material complex behaviours (sliding, impact).

Keywords: Multimaterial fluid flow, Finite Volume, Natural Interface Positioning
2000 MSC:

Email addresses: bruno.blais@polymtl.ca (Bruno Blais), jean-philippe.braeunig@cea.fr (Jean-Philippe Braeunig), daniel.chauveheid@cmla.ens-cachan.fr (Daniel Chauveheid), jmg@cmla.ens-cachan.fr (Jean-Michel Ghidaglia), raphael.loubere@math.univ-toulouse.fr (Raphaël Loubère)

Preprint submitted to The European Journal of Mechanics B Fluids 7 novembre 2012

Table des matières

1. Introduction

Dealing with multiple materials in computational fluid dynamics (CFD) is nowadays a pressing necessity. As a matter of fact, the increase of complexity in physics modeling associated to even bigger computer ressources have led to a tremendous need for multimaterial friendly numerical methods. Almost any modern Initial Fusion Confinement (ICF) computation, astrophysical simulation, biology or material models involve two or more interacting materials, with possibly different physics modeling or mechanical behavior. However the extension of a single material numerical method into a multimaterial one is usually not a trivial task, but, unavoidably, each will have to endure this evolution to deal with realistic simulations in these fields.

In [7] the authors have developed a Finite Volume with Characteristics Flux (FVCF) scheme for single material flows simulation. A two material extension has been proposed and validated in [4, 9, 2]. Since 2008 two materials flows simulations have been run, but for physics related applications this restriction on the number of materials had to be removed.
Consequently we develop in this paper an extension of FVCF that deals efficiently with an arbitrary number of materials to simulate genuine physical situations.
In section 2 we briefly present the former FVCF numerical scheme in its two material version. Then in section 3 we revamp the scheme to deal with more than two materials. The original scheme treats mixed cells (where two materials coexist) using the so-called "condensate" data structure. This notion is therefore extended to accept the occurence of an arbitrary number of materials. The original interface reconstruction method was based on Youngs' algorithm [12, 13] which has not been initially designed for more than two materials. Consequently we adopt the simplest multimaterial interface reconstruction method, called by Youngs the "onion skin" approach [12], which is a good compromise between ease of implementation and quality of results. More sophisticated interface reconstruction methods would produce even better results at the price of increasing the overall complexity of the method. This new algorithm is implemented into *FluxIC*, the FVCF simulation code. Several test cases are simulated and presented in section 4. For verification and convergence testing purposes we have run a multimaterial academical hydrodynamics test cases : a five material vortex stretching. Then genuine demanding physical three materials tests are presented, namely droplet impacts at high velocity on a thin liquid film in vacuum. These tests are intended to measure the behaviours of our approach both from an academical and engineering points of view.
Conclusions and perspectives are finally drawn.

2. FVCF-ENIP : Finite Volume Characteristics Flux with Enhanced Natural Interface Positioning technique

2.1. Governing equations

The model addressed in this work is the compressible Euler equations in d dimensions of space which can be written in a conservative form using the specific total energy as follows :

$$\frac{\partial}{\partial t}(\rho) + \operatorname{div}(\rho \boldsymbol{u}) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \operatorname{div}(\rho \boldsymbol{u} \otimes \boldsymbol{u} + pI) = 0, \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \operatorname{div}((\rho E + p)\boldsymbol{u}) = 0, \tag{3}$$

where ρ denotes the density, $\boldsymbol{u} \in \mathbb{R}^d$ the velocity field, p the pressure, $E = e + |\boldsymbol{u}|^2/2$ the specific total energy and e the specific internal energy. An equation of state (EOS) of the form $EOS(\rho, e, p) = 0$ or $p = p(\rho, e)$ is provided in order to close the system.
Let us consider the generic conservative system with $\boldsymbol{V} = (\rho, \rho\boldsymbol{u}, \rho E)^t$ the unknown vector of conservative variables and the flux $\boldsymbol{F}$ being a matrix valued function defined as :

$$\begin{array}{rccc} \boldsymbol{F}: & \mathbb{R}^{d+2} & \longrightarrow & \mathbb{R}^{d+2} \times \mathbb{R}^d \\ & \boldsymbol{V} & \longmapsto & \boldsymbol{F}(\boldsymbol{V}). \end{array} \tag{4}$$

For any direction $\boldsymbol{\nu} \in \mathbb{R}^d$, $\boldsymbol{F}(\boldsymbol{V}) \cdot \boldsymbol{\nu}$ is given in terms of $\boldsymbol{V}$ by :

$$\boldsymbol{F}(\boldsymbol{V}) \cdot \boldsymbol{\nu} = \left(\rho\,(\boldsymbol{u} \cdot \boldsymbol{\nu}), \rho\boldsymbol{u}\,(\boldsymbol{u} \cdot \boldsymbol{\nu}) + p\boldsymbol{\nu}, (\rho E + p)\,(\boldsymbol{u} \cdot \boldsymbol{\nu})\right). \tag{5}$$

The compressible Euler equations (1-3) can then be rewritten in a compact form as :

$$\partial_t \boldsymbol{V} + \operatorname{div} \boldsymbol{F}(\boldsymbol{V}) = 0, \tag{6}$$

supplemented by an EOS.

2.2. FVCF : Single material scheme

The Finite Volume Characteristics Flux for a single material has been developed and described in [7]. The FVCF scheme is written here using a directional splitting on a Cartesian structured meshes. In d dimensions of space, the algorithm described for a generic direction x has to be replicated d times, one time for each direction. However, this directional splitting does not modify the underlying single-material scheme FVCF for cells, that are both pure and not adjacent to mixed cells. Indeed, in *2D* :

- variables at $t^{n,x}$ are calculated from those at t^n by the x direction step,
- variables at t^{n+1} are calculated from those at $t^{n,x}$ by the y direction step.

$$\mathrm{Vol}_i \frac{\boldsymbol{V}_i^{n,x} - \boldsymbol{V}_i^n}{\Delta t} + A_x \left(\boldsymbol{\phi}_\ell^n + \boldsymbol{\phi}_r^n\right) = 0, \quad (7)$$

$$\mathrm{Vol}_i \frac{\boldsymbol{V}_i^{n+1} - \boldsymbol{V}_i^{n,x}}{\Delta t} + A_y \left(\boldsymbol{\phi}_d^n + \boldsymbol{\phi}_u^n\right) = 0, \quad (8)$$

where the cell volume is Vol_i, the cell face area are A_x and A_y respectively normal to x and y directions. Δt is the time step between t^n and t^{n+1}. Up, down, right and left direction fluxes $\boldsymbol{\phi}_u^n$, $\boldsymbol{\phi}_d^n$, $\boldsymbol{\phi}_r^n$, $\boldsymbol{\phi}_\ell^n$ are calculated with respect to the outgoing normal direction $\boldsymbol{\nu}_d$ of cell face Γ_d in direction d all using variables V^n at time t^n, i.e.

$$\boldsymbol{\phi}_d^n = \frac{1}{A_d} \int_{\Gamma_d} \boldsymbol{F}(\boldsymbol{V}^n) \cdot \boldsymbol{\nu}_d dS. \quad (9)$$

This flux is further approximated using the finite volume scheme FVCF described in [7]. Of course, results obtained for $\boldsymbol{V}^{n+1}$ by using two steps (7) and by adding fluxes in all directions in one step are thus strictly identical.

2.3. FVCF-ENIP : Two material scheme

ENIP stands for Enhanced Natural Interface Positioning technique. This extension of FVCF scheme to deal with two material fluid flows has been proposed in [4, 9] and improved in [2]. For the sake of clarity we only recall the main steps of this method.

The subcell model addressed here for the multimaterial representation is a cell C of volume Vol_C containing n_m different materials, each of them filling a partial volume Vol_C^k such that

$$\sum_{k=1}^{n_m} \mathrm{Vol}_C^k = \mathrm{Vol}_C. \quad (10)$$

Cell C is referred as to pure if $n_m = 1$, and as mixed if $n_m > 1$. The interfaces in mixed cells are represented by segments separating materials into two partial volumes which are pure on both sides of the interface.

A partial volume cell-centred variable vector $\boldsymbol{V}_k = (\rho_k, \rho_k \boldsymbol{u}_k, \rho_k E_k)^t$ and an equation of state $EOS_k(\rho_k, e_k, p_k) = 0$ are also associated with each material labelled by $1 \leq k \leq n_m$ in the mixed cell.

FVCF-ENIP method uses a directional splitting scheme for the interface evolution, but without modifying the single-phase Eulerian scheme by using the time integration scheme (7) in the bulk of materials. Consequently this scheme is restricted to structured Cartesian meshes.

The two material extension proposed in [9, 2] relies on conservation laws written for each partial volumes in a mixed cell, taking into account the interface as a moving boundary. The obtained finite volume scheme is conservative by construction. The time step is constrained by the same *CFL* condition on partial volumes as the single material scheme. This is one key point because the time step would be constrained by the smallest partial volume in the domain, which can be arbitrarily small. This restriction has been removed by developping a fine density and pressure control algorithm whose guidelines can be found in [10].

The ENIP numerical strategy consists in condensating neighbouring mixed cells in one cell line of the cartesian mesh, in which interfaces are considered as mono dimensional objects, but taking into account the local multi-dimensional geometry by the interface normal vector. This object is referred to as a *condensate*, which then contains layers of successive different materials that are separated by moving interfaces. The ordering of layers is given by the *2D* interface reconstruction from the previous time step which uses the volume fractions of neighbouring cells. The layer evolution is calculated in a Lagrangian fashion. Once the quantities and the new position of interfaces are calculated after a directional step, they are remapped back onto the original Eulerian mesh. When all condensates are remapped, a $2D$ normal is computed in each mixed cell for each interface as described in [8] : the method is based on an approximation of the volume fraction gradient. It is further used to locate materials within mixed cells during the remapping and to compute the fluxes through the interfaces with a sliding condition. The numerical scheme used in a condensate is presented in great details in [9] [2]. We omit its complete description in this work to focus on the interface reconstruction method, but we will give an outline now.

At time t^n, volume fractions and interfaces normal vectors $\boldsymbol{\nu}_i$ in each mixed cell i are known, then the method performs the following steps (assuming the condensate is in the x direction) :

- *Representation.* The representation step is the way of determining at which side (left or right) of an interface the material is to be put. This is done by comparing the direction of the interface normal at time t^n with the vertical direction.
- *Condensate construction.* The construction of the condensate consists of discarding any cell edges in the mixed cells considered. Then the partial volumes of the same contiguous materials are merged together

into so called condensate layers. Layers centred quantites are obtained by a conservative volume average of partial volumes quantities.
- *Condensate evolution.* The condensate layer evolution is computed from t^n to t^{n+1} thanks to the numerical scheme developed in [9].
- *Reconstruction.* We assume that the interface normals $\boldsymbol{\nu}_i$ do not change their direction during the interfaces Lagrangian evolution. The goal is to locate the materials in the Lagrangian mixed cell at t^{n+1} and to construct the piecewise linear interfaces, knowing their normal vector and the partial volume of each material.
- *Remapping/Projection.* The projection step consists of remapping the shapes obtained from the reconstruction phase onto the Eulerian grid. An exact intersection between the Lagrangian condensate obtained after the reconstruction step and the Eulerian mesh is employed.

When all mixed cells in the domain are treated for direction x, the 2D interface normals are computed using the updated volume fractions. This concludes the system evolution in direction x, the y direction can be performed.

3. Extension of FVCF-ENIP to multimaterial fluid flows

The purpose of this work is to further extend the scheme to treat more than two components in multimaterial flows. First one reviews some of the multimaterial interface reconstruction methods. Then the extension of ENIP is proposed.

3.1. Quick review of multimaterial interface reconstruction methods

Although VOF-PLIC techniques have been successfully used to accurately simulate two-material flows and free-surface flows in two and three dimensions, their application to flows involving three or more materials has been mostly *ad hoc*. The most common extensions of PLIC to cells with more than two materials is to process materials one-by-one leading to a reconstruction that is dependent on the order in which the materials are processed. In the so-called onion skin approach, each material interface is assumed to separate two materials and consists of a single line segment with both endpoints on the computational cell boundary. This form of reconstruction works only for layer structures [12, 1] and even there it may create overlapping layers if different normals are used during the reconstruction process, see Fig. 1. A more general approach is the so-called nested dissection method [5, 6], where each material is separated from the others in a (usually user-)specified order.

236

A pure polygon representing the first material is marked out from the cell, leaving a mixed polygon for the remaining materials. Then, a polygon representing the second material is marked out from the mixed polygon from previous step. The process continues until the last material is processed, see Fig. 1 middle panel for an illustration.

The interface reconstructed by one of the above methods is close to the correct configuration if the correct material ordering is provided. An incorrect ordering sometimes results in dramatic degradation of the interface. Worse, there may not be an ordering which will give the correct configuration by nested dissection like for a quadruple point as instance. Also the correct ordering in one part of the domain may be inappropriate in a different part. Finally two-material cells next to multimaterial cells may be affected by the order in which materials are processed as discussed in [16]. Finally, the presence of multiple materials in simulations creates special considerations for second-order accurate methods like LVIRA [14], see [17] for a discussion on this point. Such incorrect reconstructions adversely impact the material advection in flow simulations. In most cases, wrong material ordering results in materials being advected prematurely or belatedly into neighbouring cells. This can also lead to fragments of the material separated and drifting away.

To address this problem, there has been some work on deriving the material order automatically by Mosso and Clancy [18] and by Benson [19]. Both were designed assuming a layer shape of materials. This *a priori* assumption is sometimes inappropriate for multimaterial junctions like quadruple points. Some other techniques, restricted to three materials, have somehow addressed the material ordering problem as Choi and Bussman [15] and Caboussat *et al.* [20]. Bonnell *et al.* [21] have also described an interface reconstruction method for multiple interfaces but the method is not guaranteed to match volume fractions exactly.

Recently some authors have proposed a solution to the problems of order-dependency and a *a priori* choice of interface shapes like Schofield *et al.* [16] using a weighted Voronoi diagram (or power diagram). This method is capable of generating a material-order independent partitioning of multimaterial cells and has been shown to be second-order accurate in [17], see Fig. 1 right panel for an illustration. Yet another promising method is the Moment-Of-Fluid (MOF) method [5, 6] which demands the knowledge of first moments of materials (also called reference material centroids) in order to select the most accurate interface reconstruction method that is to say, the method minimising the difference between the reference material centroids and the

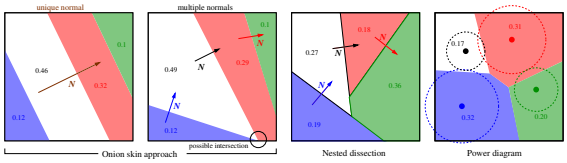


FIGURE 1: Illustration of multimaterial interface reconstruction techniques. Left panels : onion skin approach with unique normal and with multiple normals (may lead to intersection and wrong reconstruction) — Middle panel : Nested dissection, the blue polygon is reconstructed with its normal and further removed from the mixed remaining cell, then the white polygon is constructed with its normal in the mixed cell. The white polygon is then removed and the red material is constructed with its normal. The remaining of the cell corresponds to the green material — Right panel : order independent power diagram reconstruction method. The material order is blue, white, red and green for the first three panels.

generated centroids from the partitioning of the cell.
In this work we adopt the simplest interface reconstruction method namely the onion skin approach with unique normal per mixed cell, see Fig. 1. The normal is chosen to be the one from a user-given priority material. The order in which the materials are treated is chosen automaticaly by ordering them along the direction of the condensate by means of their approximate material centroids, see Fig. 3.
In fact the extension of our scheme to n_m materials using the simplest interface reconstruction method is a first required step to demonstrate the feasibility of the approach. Later we may consider the improvement brought by more advanced interface reconstruction techniques such the ones presented in this brief review.

3.2. FVCF-ENIP : multiple ($n_m > 2$) materials case

In Fig. 2 is sketched the condensate evolution when multiple materials are present in mixed cells. In this example the order in which the materials are treated is : first the brown material, second the white, then the green and the red, hence the colours of the normal in the initial representation of the condensate Fig. 2-**(A)**. Let us call $Vol_C^{k,n}$ the volume occupied by material

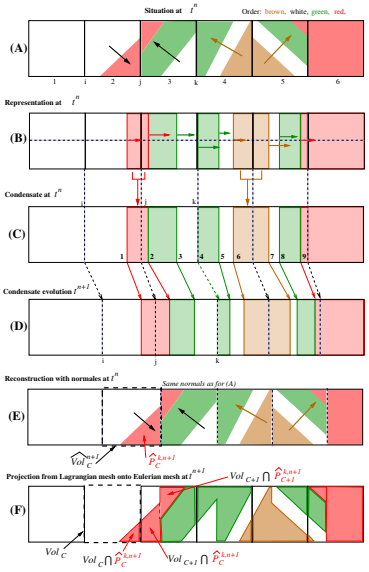


FIGURE 2: ENIP method to treat $n_m > 2$ materials — Material order : brown, white, green, red. **(A)** Situation at t^n with materials positions, interfaces and their normals to them. **(B)** Representation of materials at t^n. **(C)** Construction of a condensate at t^n by merging of mixed cells leading to layers of contiguous pieces of the same material. **(D)** Evolution of the condensate in a Lagrangian fashion during Δt. Determination of layer's compression rates through the evolution of Lagrangian cells. The cell interfaces are also displaced during Δt (black dashed lines). **(E)** Condensate reconstruction at t^{n+1} using the interface normals given at t^n. **(F)** Condensate projection/remapping from Lagrangian mesh (dashed lines) onto Eulerian mesh (straight black lines).

k in cell indexed C, C ranges from 1 to 6 in the figure. The following steps, namely **(B)**, **(C)**, **(D)** steps do not change. The result of these steps is the compression rates δVol_C^k for each piece of fluid, in other word material k occupies volume $\widetilde{Vol}_C^{k,n+1} = Vol_C^{k,n} + \delta Vol_C^k$ in the Lagrangian cell C. The reconstruction step **(E)**, even if it has not changed compared to the two-material case, requires some particular care when the compression and expansion rates are transferred to the materials. This is especially true if a more complex interface reconstruction techniques than the onion skin is to be used in **(A)**. In our case one reconstructs the interfaces using the initial normals in the new Lagrangian cell the volume of which is $\widetilde{Vol}_C^{n+1}$. Volume $\widetilde{Vol}_C^{n+1}$ is split into polygons of pure materials, say $\widehat{P}_C^{k,n+1}$, such that $\widetilde{Vol}_C^{n+1} = \bigcup_k \widehat{P}_C^{k,n+1}$. Finally step **(F)** demands the exact intersection of possible non-convex polygons of material $\widehat{P}_C^{k,n+1}$ with the Eulerian cell Vol_C (see for instance the red material in the third Eulerian cell, or green material in the fourth). In other words, for all material k, one computes

$$Vol_C^{k,n+1} = \bigcup_{C'} Vol_C \cap \widehat{P}_{C'}^{k,n+1}. \tag{11}$$

The volume fraction of material k in Eulerian cell C is given by : $f_C^k = Vol_C^{k,n+1}/Vol_C$ because $Vol_C^{n+1} \equiv Vol_C$, c being an Eulerian cell. Once all condensates in x-direction are treated, all volume fractions in the domain are updated. These volume fractions in the entire neighbourhood are finally provided to the interface reconstruction method which provides the new interfaces after the x step. Then the condensate treatment in the second direction, say y, is performed using the very same technique. Consequently we omit this description. The time step is completed once the pure cells are updated and the condensates in x and y directions have been treated.

Notice that this multiple material treatment of condensates can deal with an arbitrary number of materials within the mixed cells, is independent of the initial materials arrangement and of the interface reconstruction technique. As already mentioned we will only test the simple onion skin approach. The unique normal in mixed cell is computed with respect to the same material whose index is provided by the user. Originally the onion skin approach requires an ordered list of materials in each mixed cell. In our approach we refined this point : when the condensate in x direction is computed one orders the materials thanks to the x component of their centroids at time t^n, see

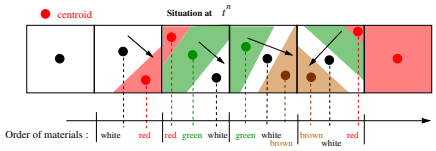


FIGURE 3: Automatic ordered list of material for the onion skin method. The materials are ordered according to their centroids relative location in the condensate direction.

Fig. 3. This order is further redefined for the y direction of the condensate. More advanced interface reconstruction techniques such as MOF [5, 6] may lead to pure polygons $\widehat{P}_C^{k,n+1}$ with simpler or more complex forms. However the exact intersection algorithm is equally able to deal with such situations.

4. Numerical results

In this section, we present some dynamic multimaterial interface reconstruction examples using the technique described in previous sections implemented into a 2D compressible Eulerian code. Many advection simulations have been run such as advection of rigid multimaterial shapes (square, disc) aligned or not-aligned with the mesh, and severe or light deformations of materials. In this paper we only present a severe distorted multimaterial test case : a four material disk is distorted due to a vortex-like motion (incompressible velocity field) and must, at final time, retrieve its original position and shape. Accordingly, a lot of hydrodynamical multimaterial verification test cases have been simulated, however we limit ourselves to the genuine demanding three material tests from the field of droplet impact on a thin liquid film, because these show the good behaviour of the numerical scheme coupled with a multimaterial interface reconstruction technique.

4.1. Four material disk test cases

A standard volume tracking test case is the disk embedded into a vortex [11]. The computational domain is $\Omega = [0, 1] \times [0, 1]$. The mesh is a regular structured grid made of squares of size $\Delta x \times \Delta y$ with $\Delta x = \Delta y = 1/N$ with

N the number of cells both in x and y directions. A four material disk of radius $R_d = 0.15$ is placed such that the center is positioned at $(0.5, 0.75)$ in Ω filled with a fifth material indexed by 0. The normal vector of material 0 is always chosen for the "onion skin" mixed cell reconstruction.

The incompressible velocity field is given by the stream-function

$$\Psi = \cos\left(\frac{\pi}{4}t\right)\frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y) \tag{12}$$

with the velocity field defined to be $\boldsymbol{U} = (u, v) = (-\frac{\partial\Psi}{\partial y}, \frac{\partial\Psi}{\partial x})$. due to the periodicity in time of Ψ, at time $t = 4$, the material configuration should be identical to the condition at time $t = 0$. Here the time step used was $\Delta t = 0.2\ \Delta x$ where Δx is the mesh spacing and the simulations were run to a final time of $t = 4.0$ with intermediate results at $t = 2.0$. Several successively refined meshes are used : $N = 32, 64, 128$ and 256.

First, a two material disk is considered, one material is indexed by 1 in the disk and the second one is indexed by 0 in the surrounding. Then a four material disk is considered. The disk is split by one horizontal and one vertical line passing through its centre. Each obtained quadrant is assigned a different material index. The results are shown for the initial $t = 0$, maximal stretch $t = 2$ and final $t = 4$ times. The two material results are displayed in Fig. 4 top panel and the five material ones in Fig. 4 bottom panel. We first remark that the shape of the disk is almost the same for the two- or five-material situations, which proves that the multimaterial treatment does not degrade the numerical scheme. Moreover we observe that the cross shape of the centre of the disk is relatively well reproduced even if the onion skin interface reconstruction method cannot take into account cross-shaped interface. The most interesting feature is the good location of each fluid. The materials are numbered as 1 (white), 2 (beige), 3 (orange), 4 (red), 5 (dark red) and we plot on the figure the average value in mixed cell between these numbers, the purpose of this figure is to show that the multimaterial treatment does not spoil a two material results and, moreover that the bulk of materials are well located. In Fig. 5 we propose a finer view for the 64×64 mesh. All mixed cells are plotted for the initial (left panel) and final (right panel) times. In each mixed cell we draw a square for each material present omitting the white surrounding material. The coloured squares can have three sizes depending if the volume fraction is less than 10^{-4} for very-small quantity, less than 10^{-2} for small quantity, or greater than 10^{-2}. Using this convention we

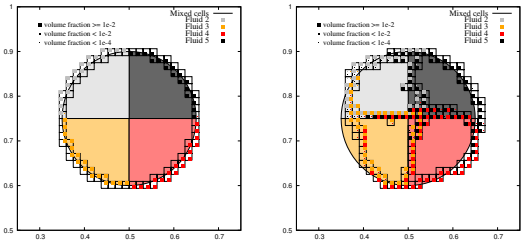


FIGURE 5: Five material vortex test at initial time (left) and at complete reversal time $t = 4.0$ (right). Run on a 64×64 grid. The squares are the mixed cell and the volume fractions are represented by coloured squares. Three sizes are shown, "macroscopic" volume fractions (greater than 10^{-2}), "microscopic" volume fractions (greater than 10^{-4}) and "mesoscopic" ones in between.

can genuinely observe how the materials have spread during the simulation. One observes that the orange and red materials have somewhat spread into the beige and orange materials although, in reality, they should not have. We can also observe that the materials form a connex structure and that no non-contiguous mixed cells are created.

For a given mesh $N \times N$, the total error in L_1 norm for material volumes is defined by

$$e^N_{vol} = \frac{\sum_{i\in Cells}\sum_{k=0}^{n_m=4}\|\mathrm{Vol}^{final}_{i,k} - \mathrm{Vol}^0_{i,k}\|}{\sum_{i\in Cells}\sum_{k=0}^{n_m=4}\mathrm{Vol}^0_{i,k}} \tag{13}$$

where $\mathrm{Vol}^{final}_{i,k}$ is the volume of material k in cell i at final time, and $\mathrm{Vol}^0_{i,k}$ is its initial volume. For each mesh we compute the error and the rate of convergence produced by the method which are displayed in Table 1 for the two materials and five materials test cases. The five materials errors are

238

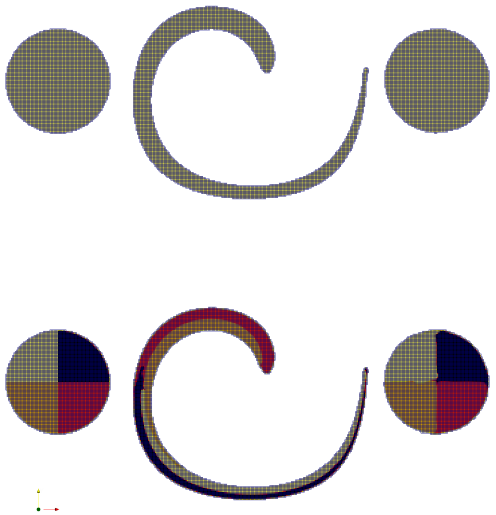


FIGURE 4: Material configuration for the two (top) and five (bottom) material vortex test at initial time $t = 0$, maximum stretch time $t = 2.0$ and at complete reversal time $t = 4.0$ run on a 256×256 grid.

N	2 materials		5 materials	
	Error	Rate	Error	Rate
32	2.69×10^{-2}	—	7.69×10^{-2}	—
64	1.24×10^{-2}	1.11	3.31×10^{-2}	1.01
128	5.93×10^{-3}	1.07	1.36×10^{-2}	1.19
256	2.82×10^{-3}	1.07	5.97×10^{-3}	1.08

TABLE 1: Errors for the disk in the vortex test case. The rate of convergence are computed as $log(e^N_{vol}/e^{N/2}_{vol})/log(2)$.

approximately two times bigger than the two materials errors. The rate of convergence is one for the original method (two materials) and the present method (five materials). This shows that dealing with more than two materials does not destroy or reduce the accuracy of the method even if the interface reconstruction technique, namely the onion skin model, is rather crude.

4.2. Hydrodynamic test cases

In this section we present multimaterial hydrodynamical test cases. The test case lies in the context of a gas droplet impacting onto a thin liquid film inside vacuum, see Fig. 6. We derive three test cases with the same initial geometry by varying the density ratio between the droplet and the liquid film. During the impact, a jet will form between the droplet and the film dragging a proportion of both materials. This phenomenum is highly dependent on the fluids inertia, because each material posseses its own velocity and a sliding condition is set at the interfaces, then each fluid can evolve independently from another, our method seems appropriate to deal with such tests. These conditions are dictated by Euler equations, where no dissipative processes exist (except in shock waves). These tests have been chosen to qualitatively illustrate the FVCF-NIP method capability.

Nevertheless, authors are aware that physical processes such as friction, molecular or turbulent diffusion between two liquids render the slip condition at the interface a bad choice in term of modelling. However, the FVCF-NIP method could take these physical mechanisms into account if they were appropriately implemented in the model. The vacuum is naturally taken into account by the FVCF-NIP method. Indeed, it is considered as the limit of a perfect gas equation of state, the interface capturing scheme in *condensates* naturally degenerates to the following vacuum model : null density,

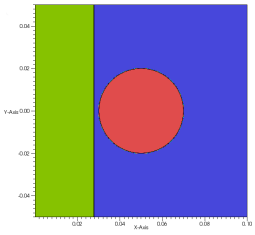


FIGURE 6: Initial condition for the droplet impact (materials are shown).

null pressure and no acoustic waves are coming from or entering the vacuum. Inside the vacuum "material", no single-phase FVCF fluxes and no evolution equations are computed since all quantities remain null.

Depending on the relative densities of the droplet and of the liquid film, one observes different behaviours in physical experiments [22]. The computational domain is $\Omega = [0; 0.1] \times [-0.05; 0.05]$ (only half of the domain is computed but for the figures the computational domain is mirrored). Wall boundary conditions are considered. The initial droplet is centred at $(0.05, 0)$ at radius $R = 0.02$ with stiffened gas equation of state $p = (\gamma - 1)\ \rho e - \Pi$, with parameters given in table 2. The droplet is in motion with horizontal velocity component $u = -10^3$ and at rest in y direction $v = 0$. The thin film is modelled with the same γ parameter as the droplet, it is located at $x \leq 0.028$. The surrounding gas is considered as a perfect gas so light compared to the fluid density, that we model it as vacuum ($\rho = 0, p = 0$). Three test cases, which depend on the realtive densities between the thin film and the droplet, are simulated (see Table 2 for the initial states) The mesh is made of 300×150 cells.

Test #1. In this test case we consider the thin film density to be two times bigger that of the droplet. In Fig. 7 are displayed the material interface and material locations at time $t = 17.63$.

Tests	Thin film				Droplet			
	ρ	p	γ	$\Pi(\times 10^9)$	ρ	p	γ	$\Pi(\times 10^9)$
#1	$2\ 10^3$	10^5	7	10	10^3	10^5	7	2.1
#2	10^3	10^5	7	2.1	10^3	10^5	7	2.1
#3	10^3	10^5	7	2.1	$2\ 10^3$	10^5	7	10

TABLE 2: Droplet impact initial conditions for the three test cases.

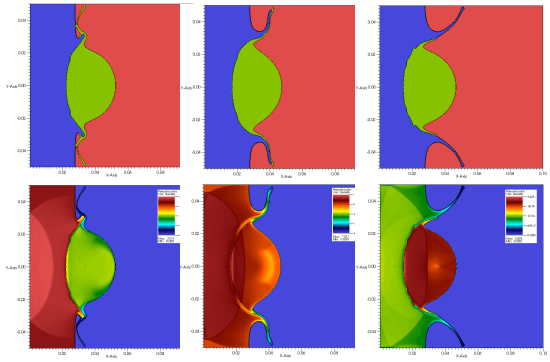


FIGURE 7: Droplet impact test cases. Left : Test #1 low density droplet at $t = 17.63$, Middle : Test #2 equal density droplet-film at $t = 25.15$, Right : Test #3 high density droplet at $t = 24.57$. Top line : materials geometry. Bottom line : Density. The black lines represent the iso-contour of each material.

Test #2. In this test case we consider the same density for the droplet and the thin film. In Fig. 7 are displayed the material interface and material locations at time $t = 25.15$.

Test #3. In this test case we consider the droplet density two times heavier than the thin film. In Fig. 7 are displayed the material interface and material locations at time $t = 24.57$.

The fact that the numerical method considers one velocity per material is illustrated on these tests by observing the jet of lighter material, should it be the droplet (Test #1) or the thin fluid (Test #3). Indeed it is expelled faster from the impact zone due to the fact that its inertia is lower. With a numerical scheme which can not differentiate the velocity of each fluid this 'sliding behaviour' would not be possible. Consequently the material with lighter inertia will drag the bigger one. This is also seen on the middle panels of Fig. 7 for which the droplet and thin film materials have the same inertia, in this case the expelled filaments are of the same size and move with each other. Qualitatively the results obtained by the method are close to experimental results which can be found in [22]. Note that these droplet impact test cases are intended to validate the interface reconstruction technique in our multimaterial hydrodynamics code. More advanced physical test cases will be simulated and quantitative results will be extracted for future communications.

5. Conclusion and perspectives

The purpose of this paper was to construct a multimaterial extension of the Finite Volume Characteristics Flux with Enhanced Natural Interface Positioning technique. A two material scheme was already available and this paper shows the modification needed to build an efficient method to deal with an arbitrary number of materials. More precisely, minor modifications have been made when a condensate is treated. We have carefully designed the method in such a way that virtually any interface reconstruction method can be used. The crude but robust onion skin interface reconstruction method has been considered in this work.
Test cases on advection or hydrodynamics have shown the efficiency of this extension. We have presented in this paper the "four material disk in a vortex" test case and three "impacts of droplet onto a thin film". These are

representatives of classical academical interface reconstruction test cases and genuine demanding hydrodynamics test cases for which multimaterial treatment is mandatory. The results obtained by the new method are very acceptable and the onion skin interface reconstruction method is mostly to be blamed for the observed weaknesses ; a unique normal per mixed cell is often a poor approximation of reality and an improper material ordering may result in materials being advected prematurely or belatedly into neighbouring cells. The reasoning behind this choice of onion skin method is the fact that the scheme has been tested with the poorest method and the results are already satisfying. Moreover more advanced interface reconstruction methods can be later coupled to the code without requiring additional modifications to the scheme.
In the near future we plan to implement more advanced order independent interface reconstruction techniques. From the physical point of view we plan to test furthermore the code on demanding physical simulation and estimate quantitatively and qualitatively our results against experimental and/or other simulations. Finally the extension to 3D and axisymetric geometry are two future points that we have to develop and validate.

Références

[1] B.Y. Choi, M. Bussmann. A piecewise linear approach to volume tracking a triple point. *Int. J. for Numer. Methods in Fluids*, 53 (6), (2007) pp. 1005-1018.

[2] R. Loubère, J.P. Braeunig, J.-M Ghidaglia A totally Eulerian finite volume solver for multi-material fluid flows : Enhanced Natural Interface Positioning (ENIP), European Journal of Mechanics B/Fluids, vol.31 , No4, pp. 1-11 (2012). doi :10.1016/j.euromechflu.2011.07.005

[3] J.P. Braeunig, L. Brosset, F. Dias, J.-M Ghidaglia, Phenomenological study of liquid impacts through 2D compressible two-fluid numerical simulations, *19th Int. Offshore and Polar Eng. Conf. Osaka, Japan, ISOPE* (2009)

[4] J.-P. Braeunig, Sur la simulation d'écoulements multi-matériaux par une méthode eulérienne directe avec capture d'interfaces en dimensions 1, 2 et 3 (technical part in english), Thèse de Doctorat N^o 2007/85, Ecole Normale Supérieure de Cachan, (2007).

[5] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data. *J. Comput. Phys.*, 227, pp. 5361-5384, (2008).

[6] H. T. Ahn and M. Shashkov. Multi-material interface reconstruction on generalized polyhedral meshes. *J. of Comput. Phys.*, 226, pp. 2096-2132 (2007).

[7] J.-M. Ghidaglia, A. Kumbaro, G. Le Coq, On the numerical solution to two fluid models via a cell centered finite volume method, *Eur. J. Mech. B–Fluids*, Vol. 20, No 6, p.841–867, (2001).

[8] D.L. Youngs, Time-Dependent multi-material flow with large fluid distortion, *Numerical Methods for Fluid Dynamics*, edited by K. W. Morton and M. J. Baines, p. 273–285, (1982).

[9] J.P. Braeunig, B. Desjardins, J.-M Ghidaglia, A totally Eulerian finite volume solver for multi-material fluid flows, European Journal of Mechanics B/Fluids, vol. 28, No4, pp. 475-485 (2009).

[10] J.P. Braeunig, An algorithm to control the pressure evolution for the FVCF-NIP method for compressible multi-material fluid flows Int. J. Finite Vol. in press (2010).

[11] W.J. Rider, D.B. Kothe. Reconstructing volume tracking. *J. of Comput. Phys.* 141 (1998) 112.

[12] D.L. Youngs. Time dependent multi-material flow with large fluid distortion. In K. W. Morton and M. J. Baines (Eds), *Numerical Methods for Fluid Dynamics*, (1982) Academic Press, New York, 273.

[13] D.L. Youngs. An interface tracking method for a 3d Eulerian hydrodynamics code. Technical Report AWE/44/92/35, AWRE Design and Math Division, 1984.

[14] J.E. Pilliod Jr., E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. of Comput. Phys.* 199 (2004) 465.

[15] B.Y. Choi, M. Bussmann. A piecewise linear approach to volume tracking a triple point. *Int. J. for Numer. Methods in Fluids* 53, (2007) 1005.

[16] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubère. Material order independent interface reconstruction using power diagrams *Int. J. for Numer. Methods in Fluids* 56(6), pp. 643-659, (2008).

[17] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubère. A Second-order Accurate Material-order-independent Interface Reconstruction Technique for Multi-Material Flow Simulations, *J. of Comput. Phys.* Volume 228, Issue 3, 20 February 2009, Pages 731-745

[18] S.Mosso and S.Clancy. A geometrically derived priority system for young's interface reconstruction. Technical Report LA-CP-95-0081, Los Alamos National Laboratory, Los Alamos, NM, 1995.

[19] D.J. Benson. Eulerian finite element methods for the micromechanics of heterogeneous materials : Dynamic prioritization of material interfaces. *Computer Methods in Applied Mechanics and Engineering* 151 (1998) 343–360.

[20] A. Caboussat, M. M. Francois, R. Glowinski, D. B. Kothe, and J. M. Sicilian. A numerical method for interface reconstruction of triple points within a volume tracking algorithm. Mathematical and Computer Modelling (2008), *in press.*

[21] K.S. Bonnell, M.A. Duchaineau, D.R. Schikore, B. Hamann, K.I. Joy. Material interface reconstruction *IEEE Transactions on Visualization and Computer Graphics* 9(4) (2003) 500.

[22] S.T. Thoroddsen, T.G. Etoh, and K. Takehara High-Speed Imaging of Drops and Bubbles Annual Review of Fluid Mechanics Vol. 40 : 257-285

Conclusions and perspectives

This thesis has presented some of the works my collaborators and I were able to pursue. Most of these investigations revolve around a joined effort to improve ALE simulation codes; Lagrangian scheme analysis and test, rezone and reconnect strategies, remapping algorithms, multi-material treatment and interface reconstruction techniques. My goal in this thesis was not only to provide some details of these works but also to provide 'the big picture behind the scene' and to try when possible to give some historical context. Moreover I have introduced several other topics which are not genuinely related to ALE such as the so-called very high-order MOOD method, some ultra fast kinetic scheme and multi-material finite volume VFFC scheme.

This is some of the current research pursued and all of these subjects are still under investigation. This helps a lot when perspectives are to be drawn ! Everything in what has been presented in this thesis is far from begin satisfactory, and improvements will be searched in the near future.
One of my goals is to construct a 3D version of a reconnection ALE code that could run on a single workstation with "light" parallelization technique. By using the most appropriate and efficient numerical methods we should be able to build such a tool for the community. As a consequence the Lagrangian schemes, the rezoner and the remapper must be somewhat revamped to adapt to the inherent difficulties of a 3D simulation code. In short, more efficient remappers (less memory-consuming), more stable Lagrangian schemes and trully poyhedral rezoner must be designed.

I pay a lot of attention to the work on bridging cell-centered and staggered Lagrangian schemes, in fact I do believe that this is a good way to improve the schemes by feeding them with success from the others. After all these formulations are probably not so far away from each others. Moreover comparing numerical methods is a thankless activity but absolutely needed to deeply understand their intrinsic behaviors which, at the end of the day, only reveal when difficult tests cases are simulated.

Unavoidable the ALE formulation is fruitful and applying this machinery to other physical contexts is kicking : elasto-plasticity, interaction of fluid/structure as instance. Before being able to properly treat these subjects we surely will have to define a good remapper for tensors. Being able to reconstruct a high-order accurate representation of a tensor, being able to limit this and begin able to remap onto a new grid are some tools that we need to construct. All of these require some careful design and study which are on my "to do" list.

Concerning the context of very high-order finite volume schemes, we are currently extruding the

essence of the original MOOD method ; it seems to be a more general design principle to build numerical schemes than a numerical method devoted to hydrodynamics. The MOOD paradigm can virtually be extended to treat steady-state solutions, parabolic/elliptic system of equations, source terms. Moreover developing a MOOD-like method mixing different schemes such as unlimited Discontinuous Galerkin schemes may be interesting. We plan to pair up our MOOD scheme with ADER like techniques. An implicit version of the MOOD scheme is also currently being worked out. Finally the validation of the MOOD concept on genuine physical contexts is under investigations. Concentrating our experiences on MOOD into a dedicated library for the community is also a wishful thinking of mine. Undoubtedly this topic will occupy some of my research time.

Because for the next years I do not plan to change my natural tendency to take advantages of good research topics, it is irrefutable that my work will also cover some other eclectic subjects.

Appendix

Abbreviations and acronyms

The abbreviations and acronyms used in this document are listed in the follwoing table.

AWE	:	Atomic Weapon Establisment, Aldermaston, U.K
CEA	:	"Commisariat à l'énergie atomique" in France
CEA-DIF	:	CEA center of Bruyères-le-Châtel, France
CEA-CESTA	:	CEA center of Le Barp, France
CELIA	:	"Centre d'Etude Laser et Interaction et Applications", University of Bordeaux, France
CMLA	:	"Centre de Mathématiques et de Leurs Applications", ENS-Cachan, Paris, France
CNRS	:	"Centre National de Recherche Scientifique" in France
CVUT	:	Czech Technical University in Prague, Czech Republic
DGA	:	"Direction Générale de L'Armement" in France
IMB	:	"Institut de Mathématiques de Bordeaux" in France
IMT	:	"Institut de Mathématiques de Toulouse" in France
INRIA	:	"Institut National de Recherche en Informatique et Automatique" in France
LANL	:	Los Alamos National Laboratory, Los Alamos, New Mexico, U.S.A
LLNL	:	Lawrence Livermore National Laboratory, Livermore, California, U.S.A
LMJ	:	Mega-joule laser located at the CEA-CESTA in France
MIP	:	"Mathématiques pour l'Industrie et la Physique" group within IMT
NIF	:	National Ignition Facility laser located at the LLNL

Bibliography

[1] R. Loubère. *Une méthode particulaire Lagrangienne de type Galerkin discontinu. Application à la mécanique des fluides et à l'interaction laser plasma*. PhD thesis, Université Bordeaux I, 2002. (Cited page 1.)

[2] R. Abgrall, R. Loubère, and J. Ovadia. A Lagrangian Discontinuous Galerkin-type method on unstructured meshes to solve hydrodynamics problems. *Int. J. Numer. Meth. Fluids*, 44 :645–663, 2004. (Cited page 1.)

[3] Weber S., G. Riazuelo, P. Michel, R. Loubère, F. Walraet, V. T. Tikhonchuk, V. Malka, J. Ovadia, and G. Bonnaud. Modeling of laser-plasma interaction on hydrodynamics scales : Physics developments and comparison with experiments. *Laser and particle beams*, 22 :189–195, 2004. (Cited page 1.)

[4] S. Weber, P.-H. Maire, R. Loubère, G. Riazuello, P. Michel, J. Ovadia, and V. Tikhonchuk. A transport simulation code for inertial confinement fusion (ICF) relevant laser/plasma interaction. *Comp. Phys. Commun.*, 168 :141–158, 2005. (Cited page 1.)

[5] R. Loubère. First steps into ALE INC(ubator) - version 2.0.0. Technical Report LAUR-04-8840, Los Alamos National Laboratory Report, 2005. see also R. Loubère, M. Shashkov, 2D Arbitrary-Lagrangian-EUlerian (ALE) code on polygonal grids for shock wave simulation, Khimicheskaya Fizika, 2005, 24 (2), pp 19-31. (Cited page 1, 101, 104 et 215.)

[6] R. Loubère and M. Shashkov. 2D arbitrary-Lagrangian-Eulerian (ALE) code on polygonal grids for shock wave simulation. *Khimicheskaya Fizika*, 24(2) :19–31, 2005. (Cited page 1.)

[7] R. Loubère and M. Shashkov. A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods. *J. Comput. Phys.*, 209(1) :105–138, 2006. (Cited page 1, 5, 98, 102, 104 et 105.)

[8] R. Loubère, M. Staley, and B.Wendroff. The repair paradigm : New algorithms and applications to compressible flow. *J. Comput. Phys.*, 1(2) :385–404, 2006. (Cited page 1, 6, 103, 115, 117 et 187.)

[9] B. Després and R. Loubère. Convergence and sensitivity analysis of repair algorithms in 1D. *Int. J. Finit. Vol.*, 3(1), 19 January 2006. (Cited page 1, 6, 103, 123 et 187.)

[10] R. Loubère and E. J. Caramana. The force/work differencing of exceptional points in the discrete, compatible formulation of Lagrangian hydrodynamics. *J. Comput. Phys.*, 216(1) :1–18, 2006. (Cited page 1, 5, 7, 49, 54, 55 et 56.)

[11] E. J. Caramana and R. Loubère. "Curl-q" : A vorticity damping artificial viscosity for Lagrangian hydrodynamics calculations. *J. Comput. Phys.*, 215(2) :385–391, 2006. (Cited page 1, 5, 7, 49, 50 et 51.)

[12] A. L. Bauer, D. E. Burton, E. J. Caramana, R. Loubère, M. J. Shashkov, and P. P. Whalen. The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics. *J. Comput. Phys.*, 218(2) :572–593, 2006. (Cited page 2, 5, 7, 8, 9, 12, 15, 16, 31, 32, 39, 45 et 101.)

[13] S.P. Schofield, R.V. Garimella, M.M. Francois, and R. Loubère. Material order independent interface reconstruction using power diagrams. *International Journal for Numerical Methods in Fluids*, 56 :643–659, 2008. (Cited page 2, 6, 165, 168, 170, 171 et 173.)

[14] M. Kuchařík, R. Liska, R. Loubère, and M. Shashkov. Arbitrary-Lagrangian-Eulerian (ALE) method in cylindrical coordinates for laser/plasma simulation. In Sylvie Benzoni-Gavage and Denis Serre, editors, *Hyperbolic problems : Theory, Numerics, Application, Proceedings of the 11th Int. Conf on Hyperbolic problems, ENS Lyon July 17-21*. Springer-Verlag Berlin Heidelberg, 2006. (Cited page 2.)

[15] A.L. Bauer, R. Loubère, and B.B. Wendroff. On stabiliy analysis of staggered schemes. *SIAM Journal on Numerical Analysis*, 46 :996–1011, 2008. (Cited page 2, 5, 31, 39 et 40.)

[16] R. Loubère, M. Shashkov, and B. Wendroff. Volume consistency in a staggered grid Lagrangian hydrodynamics scheme. *J. Comput. Phys.*, 227 :3731–3737, 2008. (Cited page 2, 5, 31, 45 et 46.)

[17] Samuel P. Schofield, Rao V. Garimella, Marianne M. Francois, and Raphaël Loubère. A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations. *J. Comput. Phys.*, 228 :731–745, 2009. (Cited page 2, 6, 165, 170, 171, 172, 173, 179 et 231.)

[18] Jean-Philippe Braeunig, Jean-Michel Ghidaglia, and Raphaël Loubère. A totally Eulerian finite volume solver for multi-material fluid flows : Enhanced natural interface positioning (ENIP). *European Journal of Mechanics - B/Fluids*, 31 :1–11, 2012. (Cited page 2, 6, 186 et 227.)

[19] C. Fochesatto, R. Loubère, R. Motte, and J. Ovadia. An interface reconstruction method devoted to filament. *Computer and Fluids*, 2012. to be submitted. (Cited page 2.)

[20] Richard Liska, Raphaël Loubère, Pierre-Henri Maire, Jerome Breil, Stephane Galera, and Pavel Váchal. Comparison of staggered and cell-centered Lagrangian and ALE hydrodynamical methods. In Tadmor, E and Liu, J and Tzavaras, A, editor, *HYPERBOLIC PROBLEMS : THEORY, NUMERICS AND APPLICATIONS, PART 2*, volume 67, Part 2 of *Proceedings of Symposia in Applied Mathematics*, pages 755–764, 2009. 12th International Conference on Hyperbolic Problems, College Park, MD, JUN 09-13, 2008. (Cited page 2 et 159.)

[21] R. Loubère, P.-H. Maire, and P. Váchal. Formulation of a staggered two-dimensional Lagrangian scheme by means of cell-centered approximate riemann solver. In Maya Neytcheva Editors Gunilla Kreiss Per Lötstedt, Axel Malqvist, editor, *Proceedings of ENUMATH 2009, the 8th European Conference on Numerical Mathematics and Advanced Applications, Uppsala, Sweden*. Springer, 2009. (Cited page 2 et 70.)

[22] Raphaël Loubère, Pierre-Henri Maire, and Pavel Váchal. A second-order compatible staggered Lagrangian hydrodynamics scheme using a cell-centered multidimensional approximate Riemann solver. In *ICCS 2010 - INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE, PROCEEDINGS*, volume 1 of *Procedia Computer Science*, pages 1925–1933, 2010. International Conference on Computational Science(ICCS), Amsterdam, NETHERLANDS, MAY 31-JUN 02, 2010. (Cited page 2 et 70.)

[23] R. Loubère, P.-H. Maire, and P. Váchal. Staggered Lagrangian discretization based on cell-centered riemann solver and associated hydro-dynamics scheme. *Communication in Computational Physics*, 10(4) :940–978, 2011. doi :10.4208/cicp.170310.251110a. (Cited page 2, 5, 49, 70, 73 et 98.)

[24] P. Váchal R. Loubère, P.-H. Maire. 3D staggered Lagrangian hydrodynamics with cell-centered riemann solver based artificial viscosity. *Int. J. Numer. Meth. Fluid*, 72(13) :22–42, 2013. Article first published online : 27 SEP 2012, DOI : 10.1002/fld.3730. (Cited page 3, 5, 49, 70, 73, 74, 75 et 76.)

[25] M. Kuchařík, R. Loubère, L. Bednárik, and R. Liska. Enhancement of Lagrangian slide lines as a combined force and velocity boundary condition. *Computer and Fluids*, in press :–, 2012. available on line 15 juin 2012. (Cited page 3, 5, 15, 49 et 62.)

[26] R. Loubère, P.-H Maire, M. Shashkov, J. Breil, and S. Galera. ReALE : A Reconnection-based Arbitrary-Lagrangian-Eulerian method. *J. Comput. Phys.*, 229 :4724–4761, 2010. (Cited page 3, 6, 129, 131, 135, 138, 139, 140, 141, 142 et 143.)

[27] R. Loubère, P.-H. Maire, and M. Shashkov. ReALE : A reconnection arbitrary-Lagrangian-Eulerian method in cylindrical geometry. *Computer and Fluids*, 46 :59–69, 2011. doi :10.1016/j.compfluid.2010.08.024. (Cited page 3, 6, 129, 140, 141, 144 et 145.)

[28] L. Carballal-Perdiz, P. Degond, F. Deluzet, R. Loubère, and J.-M. Rovarch. PRACT-X : PRediction of Air Contaminant Transport on multiple (X) scales. Technical report, Rapport DGA - Convention 2007 - 25 - 011, 2009. in preparation with A. Lozinski "Multiscale finite element method for perforated domains". (Cited page 3.)

[29] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for hyperbolic systems : Multi-dimensional Optimal Order Detection (MOOD). *J. Comput. Phys.*, 230(10) :4028–4050, 10 May 2011. (Cited page 3, 6, 185 et 188.)

[30] S. Clain, S. Diot, and R. Loubère. Multi-dimensional Optimal Order Detection (MOOD) - A very high-order finite volume scheme for conservation laws on unstructured meshes. In J. Fort, J. Fürst, J. Halama, R. Herbin, and Hubert, editors, *FVCA 6, International Symposium, Prague, June 6-10, 2011*, volume 4 of *Proceedings in Mathematics, 1st Edition*. Springer, 2011. 1065 p. 106 illus. in color. (Cited page 3 et 185.)

[31] S. Clain, S. Diot, R. Loubère, G. Machado, R. Ralha, and R.M.S. Pereira. Very high-order finite volume method for one-dimensional convection diffusion problems. In *MATHEMATICAL MODELS FOR ENGINEERING SCIENCE (MMES'11), International conference of the Institute for Environment, Engineering, Economics and Applied Mathematics (IEEEAM)*, 2012. (Cited page 3 et 185.)

[32] S. Diot, S. Clain, and R. Loubère. Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials. *Computer and Fluids*, 64 :43–63, 2012. http ://dx.doi.org/10.1016/j.compfluid.2012.05.004. (Cited page 3, 6, 185 et 188.)

[33] R. Loubère S. Clain, S. Diot. The MOOD method in the three-dimensional case : Very- high-order finite volume method for hyperbolic systems. *Int. J. Numer. Meth. Fluid*, 2013. accepted in January 2013. (Cited page 3, 6, 185 et 188.)

[34] R. Loubère. Etude des schémas numériques ALE multi matériaux 1. Technical report, Rapport final du contrat d'étude liant de CEA-DAM et l'IMT, 2007. 101 pages. (Cited page 3.)

[35] R. Loubère. Etude des schémas numériques ALE multi matériaux 2. Technical report, Rapport final du contrat d'étude liant de CEA-DAM et l'IMT, 2008. 78 pages. (Cited page 3.)

[36] R. Loubère. Etude des schémas numériques ALE multi matériaux 3. Technical report, Rapport final du contrat d'étude liant de CEA-DAM et l'IMT, 2009. 144 pages. (Cited page 3.)

[37] R. Loubère. Etude des schémas numériques ALE multi matériaux 4. Technical report, Rapport final du contrat d'étude liant de CEA-DAM et l'IMT, 2010. 97 pages. (Cited page 3.)

[38] R. Loubère. Etude des schémas numériques ALE multi matériaux 5. Technical report, Rapport final du contrat d'étude liant de CEA-DAM et l'IMT, 2011. 67 pages. (Cited page 3.)

[39] G. Dimarco and R. Loubère. Towards an ultra efficient kinetic scheme part I : basics on the bgk equation. *J. Comput. Phys.*, 2012. accepted. (Cited page 3, 6, 185, 212 et 213.)

[40] G. Dimarco and R. Loubère. Towards an ultra efficient kinetic scheme part I : the high-order case. *J. Comput. Phys.*, 2013. submitted. (Cited page 3.)

[41] Bruno Blais, Jean-Philippe Braeunig, Jean-Michel Ghidaglia, and Raphaël Loubère. Dealing with more than two materials in FVCF-ENIP method. *submitted to European Journal of Mechanics - B/Fluids*, 2012. (Cited page 6, 186, 227 et 230.)

[42] P.-H. Maire. Contribution à la modélisation numérique de la fusion par confinement inertiel. *Habilitation à Diriger des Recherches, Université de Bordeaux*, 2011. (Cited page 7, 12, 28 et 70.)

[43] B. Després and C. Mazeran. Lagrangian gas dynamics in two dimensions and Lagrangian systems. *Arch. Rational Mech. Anal.*, 178 :327–372, 2005. (Cited page 7 et 17.)

[44] G. Carré, S. Delpino, B. Després, and E. Labourasse. A cell-centered Lagrangian hydrodynamics scheme in arbitrary dimension. *J. Comput. Phys.*, 228(14) :5160–5183, 2009. (Cited page 7.)

[45] P.-H. Maire and B. Nkonga. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *J. Comput. Phys.*, 228(3) :799–821, 2009. (Cited page 7 et 135.)

[46] A.J. Barlow. A compatible finite element multi-material ALE hydrodynamics algorithm. *International Journal for Numerical Methods in Fluids*, 56 :953–964, 2008. (Cited page 7.)

[47] B. Rebourcet. Comments on the filtering of numerical instabilities in Lagrangian hydrocodes. Conference on Numerical methods for multi-material fluid flows ; Czech Technical University in Prague on September 10 - 14, 2007 ; http://www-troja.fjfi.cvut.cz/ multimat07/presentations/tuesday/Rebourcet_filtering.pdf. (Cited page 7.)

[48] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *SIAM J. Sci. Comput.*, 29(4) :1781–1824, 2007. (Cited page 7 et 70.)

[49] P.-H. Maire and J. Breil. A second-order cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *Int. J. Numer. Meth. Fluids*, 56 :1417–1423, 2008. (Cited page 7, 70 et 135.)

[50] P.-H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *J. Comput. Phys.*, 228 :2391–2425, 2009. (Cited page 7, 17, 70 et 135.)

[51] A. Barlow and P. Roe. A cell centred lagrangian godunov scheme for shock hydrodynamics. *Comput. Fluids*, 46(1) :133–136, 2011. (Cited page 7, 17 et 70.)

[52] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comp. Meth. Appl. Mech. Engrg.*, 99 :235–394, 1992. (Cited page 7, 17, 95, 98, 99, 104 et 164.)

[53] J. C. Campbell and M. J. Shaskhov. A tensor artificial viscosity using a mimetic finite difference algorithm. *J. Comput. Phys.*, 172(4) :739–765, 2001. (Cited page 7, 17, 20, 21, 50, 54 et 135.)

[54] J. C. Campbell and M. J. Shashkov. A compatible Lagrangian hydrodynamics algorithm for unstructured grids. *SelÃğuk J. Appl. Math.*, 4(2) :53–70, 2003. (Cited page 7 et 135.)

[55] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146(1) :227–262, 1998. (Cited page 7, 11, 12, 13, 15, 21, 28, 29, 30, 101 et 104.)

[56] E. J. Caramana and M. J. Shashkov. Elimination of artificial grid distorsion and hourglass–type motions by means of Lagrangian subzonal masses and pressures. *J. Comput. Phys.*, 142 :521–561, 1998. (Cited page 7, 15, 22 et 102.)

[57] E. J. Caramana, M. J. Shashkov, and P. P. Whalen. Formulations of artificial viscosity for multidimensional shock wave computations. *J. Comput. Phys.*, 144 :70–97, 1998. (Cited page 7, 17, 18, 19, 45, 50, 54 et 74.)

[58] G. Scovazzi, M. A. Christon, T. J.R. Hughes, and J. N. Shadid. Stabilized shock hydrodynamics : I. A Lagrangian method. *Comp. Meth. Appl. Mech. Engrg.*, 196 :923–966, 2007. (Cited page 7.)

[59] G. Scovazzi. Stabilized shock hydrodynamics : II. Design and physical interpretation of the SUPG operator for Lagrangian computations. *Comp. Meth. Appl. Mech. Engrg.*, 196 :966–978, 2007. (Cited page 7.)

[60] G. Scovazzi, E. Love, and M. J. Shashkov. Multi-scale Lagrangian shock hydrodynamics on Q1/Po finite elements : Theoretical framework and two-dimensional computations. *Comp. Meth. Appl. Mech. Engrg.*, 197 :1056–1079, 2008. (Cited page 7.)

[61] V.A. Dobrev, T.E. Ellis, Tz. V. Kolev, and R. N. Rieben. Curvilinear finite elements for Lagrangian hydrodynamics . *Int. J. Numer. Meth. Fluid*, 2010. doi : 10.1002/fld.2366. (Cited page 7.)

[62] Tz.V. Kolev and R.N. Rieben. A tensor artificial viscosity using a finite element approach. *J. Comput. Phys.*, 228(22) :8336–8366, 2010. (Cited page 7 et 17.)

[63] E. J. Caramana and P. P.Whalen. Numerical preservation of symmetry properties of continuum problems. *J. Comput. Phys.*, 141 :174–198, 1998. (Cited page 7, 29 et 101.)

[64] E. J. Caramana, C. L. Rousculp, and D. E. Burton. A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry. *J. Comput. Phys.*, 157 :89–119, 2000. (Cited page 7 et 101.)

[65] R. D. Richtmyer and K. W Morton. *Difference Methods for Initial Value Problems 2nd ed.* Interscience Publishers, 1967. (Cited page 8 et 30.)

[66] H.A. Bethe, R. Christy, R.R. Davis, R.P. Feynman, S. Frankel, E.J. Konopinski, N. Livesay, N. Metropolis, and E. Nelson. IBM calculations of implosion hydrodynamics. Technical report, Los Alamos National Laboratory report, LA-94, June 20 1944. unclassified, cover only. (Cited page 8.)

[67] J. von Neumann and R. D. Richtmyer. A method for the numerical calculations of hydrodynamical shocks. *J. Appl. Phys.*, 21 :232–238, 1950. (Cited page 8 et 17.)

[68] R. D. Richtmyer and K. W. Morton. *Difference methods for initial-value problems*. John Wiley, Interscience Publishers, 1967. (Cited page 8.)

[69] M. Rotenberg (Eds.) B. Alder, S. Fernbach, editor. *Methods in Computational Physics, Advances in Research and Applications,Fundamental Method in Hydrodynamics*, volume 3. Academic Press, 1964. (Cited page 8, 9, 29 et 31.)

[70] W.B. Goad. WAT : a numerical method for two-dimensional unsteady fluid flow. Technical Report LAMS-2365, Los Alamos National Laboratory Report, 1960. (Cited page 8.)

[71] A.P. Favorskii. Variational-discrete models of hydrodynamics equations. *Diff. Equations*, 16 :1308–1321, 1980. (Cited page 9 et 31.)

[72] T.F. Adams L.G. Margolin. Spatial differencing for finite difference codes. Technical Report LA-10249, Los Alamos National Laboratory Repor, 1985. (Cited page 9 et 31.)

[73] A.A. Samarskii, V.F. Tishkin, A.P. Favorskii, and M.J. Shashkov. Operational finite difference schemes. *Diff. Equations*, 17 :854–862, 1981. (Cited page 9.)

[74] D. E. Burton. Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity. *Advances in the Free Lagrange Method*, Springer-Verlag, New-York, 1990. (Cited page 9, 12, 15 et 135.)

[75] D. E. Burton. Multidimensional Discretization of Conservation Laws for Unstructured Polyhedral Grids. Technical Report UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994. (Cited page 9 et 95.)

[76] L.G. Margolin, M.J. Shashkov, and M.A. Taylor. Symmetry-preserving discretizations for Lagrangian gas dynamics. In P. Neittaanmäaki, T. Tiihonen, and P. Tarvainen, editors, *Proceedings of the 3rd European Conference, Numerical Mathematics and Advanced Applications*, pages 725–732. World Scientific, 2000. (Cited page 12 et 13.)

[77] M. Shashkov. *Conservative finite difference methods on general grids*. CRC Press, 1996. (Cited page 13.)

[78] D. E. Burton. Multidimensional discretization of conservation laws for unstructured polyhedral grids. *Report UCRL-JC-118306, Lawrence Livermore National Laboratory*, 1994. Proceedings of Second International Workshop on Analytical Methods and Process Optimization in Fluid and Gas Mechanics (SAMGOP), Arzamas-16, Russia. (Cited page 15 et 135.)

[79] P.-H. Maire, R. Abgrall, J. Breil, R. Loubère, and B. Rebourcet. A nominally second-order cell-centered lagrangian scheme for simulating elastic-plastic flows on two-dimensional unstructured grids. *J. Comput. Phys.*, 235 :626–665, 2013. also available at http://hal.inria.fr/hal-00701802. (Cited page 15.)

[80] E. J. Caramana. The implementation of slide lines as a combined force and velocity boundary condition. *J. Comput. Physics*, 228(11) :3911–3916, 2009. (Cited page 15, 49, 62, 63 et 64.)

[81] R. Landshoff. A numerical method for treating fluid flow in the presence of shocks. Technical Report LA-1930, Los Alamos National Laboratory, 1955. (Cited page 17.)

[82] W.D. Schulz. *Methods in Computational Physics*, volume 3, chapter Two-dimensional Lagrangian hydrodynamic difference equations, pages 1–45. Academic Press, 1964. (Cited page 17, 18 et 29.)

[83] W.D. Schulz. Tensor artificial viscosity for numerical hydrodynamics. *J. Math. Phys.*, 1964. (Cited page 17 et 18.)

[84] G.T. Richards. Derivation of a generalized von neumann pseudo-viscosity with directional properties. Technical Report UCRL-14244, Lawrence Livermore Radiation Laboratory, 1965. (Cited page 17.)

[85] M. L. Wilkins. Use of artificial viscosity in multidimensionnal fluid dynamic calculations. *J. Comput. Phys.*, 36 :381–403, 1980. (Cited page 17.)

[86] V.F. Kurapatenko. *Difference Methods for Solutions of Problems of Mathematical Physics*, chapter -. Janenko, N. N., American Mathematical Society, Providence, RI,, 1967. (Cited page 17.)

[87] R. B. Christensen. Godunov methods on a staggered mesh - an improved artificial viscosity. Technical Report UCRL-JC-105269, Lawrence Livermore National Laboratory, 1991. (Cited page 17 et 72.)

[88] J. K. Dukowicz. A general, non-iterative Riemann solver for Godunov's method. *J. Comput. Phys.*, 61 :119–137, 1984. (Cited page 17 et 71.)

[89] J. K. Dukowicz and B. Meltz. Vorticity errors in multidimensional Lagrangian codes. *J. Comput. Phys.*, 99 :115–134, 1992. (Cited page 17.)

[90] W. F. Noh. Errors for calculations of strong shocks using artificial viscosity and an artificial heat flux. *J. Comput. Phys.*, 72 :78–120, 1987. (Cited page 17.)

[91] D. J. Benson. A new two-dimensional flux-limited shock viscosity for impact calculations. *Computer Methods in Applied Mechanics and Engineering*, 93(1) :39 – 95, 1991. (Cited page 17.)

[92] K.Lipnikov. A framework for developing a mimetic tensor artificial viscosity for lagrangian hydrocodes on arbitrary polygonal meshes. *J. Comput. Phys.*, 229 :7911–7941, 2010. (Cited page 17.)

[93] G.Maenchen and S. Sack. *Methods in Computational Physics*, volume 3, chapter The Tensor code, pages 181–210. Academic Press, 1964. (Cited page 17.)

[94] M.Owen. A tensor artificial viscosity for SPH. *J. Comput. Phys.*, 201(2) :601 – 629, 2004. (Cited page 17.)

[95] R. Loubère. On the effect of the different limiters for the tensor artificial visosity for the compatible langrangian. Technical report, Los Alamos report, LAUR-05-9301, 2005. (Cited page 21.)

[96] G. Scovazzi and E. Love. A generalized view on galilean invariance in stabilized compressible flow computations. *Int. J. Numer. Meth. Fluid*, 64(10-12) :1065–1083, 2010. (Cited page 22.)

[97] M. Kuchařík. *Arbitrary-Lagrangian-Eulerian (ALE) methods in plasma physics*. PhD thesis, Czech technical University in Prague, 2006. (Cited page 29.)

[98] M.L. Wilkins. *Methods in Computational Physics*, volume 3, chapter Calculation of elastic-plastic flows, pages 211–263. Academic Press, 1964. (Cited page 29.)

[99] V. F. Tishkin, N. N. Tiurina, and A. P. Favorskii. Finite-difference schemes for calculating hydrodynamic flows in cylindrical coordinates. *preprint, No. 23, Keldysh Institute of Applied Mathematics, Moscow, Russia*, 1978. (Cited page 29.)

[100] N. V. Mikhailova, V. F. Tishkin, N. N. Tyurina, A. P. Favorskii, and M. Y. Shashkov. Numerical modelling of two-dimensional gas-dynamic flows on a variable-structure mesh. *USSR Comput. Math. Math. Phys.*, 26(5) :74–84, 1988. (Cited page 29.)

[101] P. Lax and B. Wendroff. Difference schemes for hyperbolic equations with high order of accuracy. *Comm. Pure Appl. Math.*, XVII :381–398, 1964. (Cited page 39.)

[102] P.-H. Maire, J. Breil, and S. Galera. A cell-centered arbitrary Lagrangian Eulerian (ALE) method. *Int. J. Numer. Meth. Fluid*, 56 :1161–1166, 2008. (Cited page 70.)

[103] P.-H. Maire. A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry. *J. Comput. Phys.*, 228 :6882–6915, 2009. (Cited page 70.)

[104] P.-H. Maire. A unified sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids. *Int. J. Numer. Meth. Fluid*, 65(11) :1281–1294, 2010. (Cited page 70.)

[105] P.-H. Maire. A high-order one-step sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids. *Computer and Fluids*, 46(1), 2011. (Cited page 70.)

[106] A. Burbeau-Augoula. A Node-Centered Artificial Viscosity Method for Two-Dimensional Lagrangian Hydrodynamics Calculations on a Staggered Grid. *Commun. Comput. Phys.*, 8 :877–900, 2009. (Cited page 70.)

[107] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys.*, 14 :227–253, 1974. Reprinted at *J. Comp. Phys.*, 135 :203–216, 1997. (Cited page 95.)

[108] D. J. Benson. An efficient, accurate, simple ALE method for nonlinear finite element programs. *Comp. Meth. Appl. Mech. Engrg.*, 72 :305–350, 1989. (Cited page 95 et 130.)

[109] L.G. Margolin. Introduction to "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds". *J. Comput. Phys.*, 135 :198–202, 1997. (Cited page 95.)

[110] D.S. Kershaw, M.K. Prasad, M.J. Shaw, and J.L. Milovich. 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method. *Comp. Meth. Appl. Mech. Engrg.*, 158 :81–116, 1998. (Cited page 95.)

[111] P. Kjellgren and J.Hyvarien. An arbitrary Lagrangian-Eulerian finite element method. *Computational Mechanics*, 21 :81–90, 1998. (Cited page 95.)

[112] J. S. Peery and D. E. Carroll. Multi-material ALE methods in unstructured grids. *Comp. Meth. Appl. Mech. Engrg.*, 187 :591–619, 2000. (Cited page 95 et 130.)

[113] B. R. Pember R. W. Anderson, N. S. Elliott. An arbitrary lagrangian-eulerian method with adaptive mesh refinement for the solution of the euler equations. *J. Comput. Phys.*, 199(2) :598–617, 2004. (Cited page 95.)

[114] D. E. Burton. Consistent finite-volume discretization of hydrodynamics conservation laws for unstructured grids. *Report UCRL-JC-118788, Lawrence Livermore National Laboratory*, 1994. (Cited page 95 et 135.)

[115] A. M. Winslow. Equipotential zoning of two-dimensional meshes. Technical Report UCRL-7312, Lawrence Livermore National Laboratory, 1963. (Cited page 96 et 130.)

[116] A. Winslow. Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh. *J. Comput. Phys.*, 1 :149–172, 1966. (Cited page 96 et 130.)

[117] A. Barlow. ALE and AMR mesh refinement techniques for multi-material hydrodynamics problems. ICFD Workshop on Mesh Refinement Techniques, Oxford, U.K, 2005. (Cited page 96.)

[118] Markus Berndt, Milan Kuchařík, and Mikhail J. Shashkov. Using the feasible set method for rezoning in ALE. *Procedia CS*, 1(1) :1885–1892, 2010. (Cited page 97.)

[119] P. Knupp. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I – a framework for surface mesh optimization. *Int. J. Numer. Meth. Eng.*, 48 :401–420, 2000. (Cited page 97.)

[120] P. Knupp and N. Robidoux. A framework for variational grid generation : Conditioning the jacobian matrix with matrix norms. *SIAM J. on Scientific Computing*, 21(6) :2029–2047, 2000. (Cited page 97.)

[121] P. Knupp. Algebraic mesh quality metrics. *SIAM J. Sci. Comput.*, 23(1) :193–218, 2001. (Cited page 97.)

[122] P. Knupp, L. G. Margolin, and M. Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *J. Comput. Phys.*, 176 :93–128, 2002. (Cited page 97.)

[123] P. Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39(3) :217 – 241, 2003. (Cited page 97.)

[124] P. Knupp, L. G. Margolin, and M. Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *J. Comput. Phys.*, 176 :93–128, 2002. (Cited page 97, 104 et 130.)

[125] Pavel Vachal, Rao V. Garimella, and Mikhail J. Shashkov. Untangling of 2D meshes in ALE simulations. *Journal of Computational Physics*, 196(2) :627 – 644, 2004. (Cited page 97 et 104.)

[126] R. Loubère. On the automatic choice of the rezone strategy in arbitrary-Lagrangian-Eulerian hydrocodes. Technical report, Personal communication, 2006. (Cited page 97.)

[127] L.G. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *J. Comput. Phys.*, 184(1) :266–298, 2003. (Cited page 98 et 137.)

[128] M. Kuchařík, M. Shashkov, and B. Wendroff. An efficient linearity-and-bound-preserving remapping methods. *J. Comput. Phys.*, 188 :462–471, 2003. (Cited page 98, 103, 115 et 137.)

[129] J. Grandy. Conservative remapping and regions overlays by intersecting arbitrary polyhedra. *J. Comput. Phys.*, 148 :433–466, 1999. (Cited page 98 et 137.)

[130] R. Pember and R. Anderson. Comparison of direct Eulerian Godunov and Lagrange plus remap artificial viscosity schemes for compressible flow. Technical Report 2001-2644, AIAA, 2001. (Cited page 98.)

[131] W.B. VanderHeyden and B.A. Kashiwa. Compatible fluxes from van Leer advection. *J. Comput. Phys.*, 146 :1–28, 1998. (Cited page 98.)

[132] J.K. Dukowicz and J.W. Kodis. Accurate conservative remapping (rezoning) for Arbitrary-Lagrangian-Eulerian computations. *SIAM J. Stat. Comput.*, 8 :305–321, 1987. (Cited page 98.)

[133] D. J. Benson. Momentum advection on a staggered mesh. *J. Comput. Phys.*, 100 :143–162, 1992. (Cited page 98.)

[134] J.K. Dukowicz and J. Baumgardner. Incremental remapping as a transport/advection algorithm. *J. Comput. Phys.*, 160 :318–335, 2000. (Cited page 98 et 115.)

[135] P.J. O'Rourke and M.S. Sahota. A variable explicit/implicit numerical method for calculating advection on unstructured meshes. *J. Comput. Phys.*, 143 :312–345, 1998. (Cited page 98.)

[136] G. Luttwak and J. Falcovitz. Slope limiting for vectors : A novel vector limiting algorithm. *Int. J. Numer. Meth. Fluid*, 2010. doi : 10.1002/fld.2367. (Cited page 98.)

[137] Peter H. Lauritzen, Christoph Erath, and Rashmi Mittal. On simplifying incremental remap-based transport schemes. *Journal of Computational Physics*, 230(22) :7957 – 7963, 2011. (Cited page 98.)

[138] P. Hoch. Mesh quality and conservative projection in Lagrangian compressible hydrodynamic. Conference on Numerical methods for multi-material fluid flows ; Czech Technical University in Prague on September 10 - 14, 2007. (Cited page 98 et 130.)

[139] F. L. Adessio, D. E. Carroll, K. K. Dukowicz, J. N. Johnson, B. A. Kashiwa, M. E. Maltrud, and H. M. Ruppel. CAVEAT : a computer code for fluid dynamics problems with large distortion and internal slip. Technical Report LA-10613-MS, Los Alamos National Laboratory, 1986. (Cited page 100.)

[140] David Bailey, Markus Berndt, Milan Kuchařík, and Mikhail Shashkov. Reduced-dissipation remapping of velocity in staggered arbitrary lagrangian-eulerian methods. *Journal of Computational and Applied Mathematics*, 233(12) :3148 – 3156, 2010. Finite Element Methods in Engineering and Science (FEMTEC 2009). (Cited page 100.)

[141] M. Shashkov and B. Wendroff. The repair paradigm and application to conservation laws. *J. Comput. Phys*, 198 :265–277, 2004. (Cited page 103, 115 et 187.)

[142] L.G Margolin and M. Shashkov. Remapping, recovery and repair on staggered grid. *Computer Methods in Applied Mechanics and Engineering*, 193 :4139–4155, 2004. (Cited page 103, 115 et 187.)

[143] Anabela Oliveira and André B. Fortunato. Toward an oscillation-free, mass conservative, Eulerian-Lagrangian transport model. *Journal of Computational Physics*, 183(1) :142 – 164, 2002. (Cited page 115.)

[144] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87(1) :171 – 200, 1990. (Cited page 115.)

[145] R. Lohner and C. Yang. Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods in Engineering*, 12 :599–608, 1996. (Cited page 130.)

[146] C. Stoker, C. Gay, F. Bay, and J.-L. Chenot. A velocity approach for the ALE-method applied for 2D and 3D problems. In *Simulation of Material Processing : Theory, Methods and Applications*. Balkema, Rotterdam, 1998. (Cited page 130.)

[147] J. K. Dukowicz, M. C. Cline, and F. S. Addessio. A general topology Godunov method. *J. Comput. Phys.*, 82 :29–63, 1989. (Cited page 130, 132, 135 et 137.)

[148] S. Giuliani. An algorithm for continuous rezoning of the hydrodynamic grid in arbitrary Lagrangian-Eulerian computer codes. *Nuclear Engineering and Design*, 72 :205–212, 1982. (Cited page 130.)

[149] J.U. Brackbill and J.S. Saltzman. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys.*, 46 :342–368, 1982. (Cited page 130.)

[150] M.J. Baines. *Moving Finite Elements*. Oxford Science Publications, Clarendon Press, Oxford, 1994. (Cited page 130.)

[151] J.K. Dukowicz. A simplified adaptive mesh technique derived from the moving finite element method. *J. Comput. Phys.*, 56 :324–342, 1984. (Cited page 130.)

[152] K. Lipnikov and M. Shashkov. The error-minnimization-based strategy for moving mesh methods. *Commun. in Comput. Phys.*, 1(1) :53–81, 2006. (Cited page 130.)

[153] A.J. Barlow. ALE and AMR mesh refinement techniques for multi-material hydrodynamics problems. ICFD Workshop on Mesh Refinement Techniques 7th December 2005, `http://www.icfd.rdg.ac.uk/Workshops/AMR/meshreftechf.pdf`. (Cited page 130.)

[154] A.J. Barlow. Challenges and recent progress in developing numerical methods for multi-material ALE hydrocodes. ICFD 25 year Anniversary Conference 15-16th September 2008 Oxford University, `http://www.icfd.rdg.ac.uk/ICFD25/Talks/ABarlow.pdf`. (Cited page 130.)

[155] J.R. Pasta and S. Ulam. Heuristic numerical work in some problems of hydrodynamics. *Math. Tables and other Aids to Computation*, XIII(65) :1–12, 1959. (Cited page 131.)

[156] M.J. Fritts, W.P. Crowley, and H. Trease, editors. *The Free-Lagrange Method : Proceedings of the First International Conference of Free-Lagrange Methods*. Springer-Verlag, New York, 1985. Lecture Notes in Physics, Vol. 238. (Cited page 131.)

[157] J.U. Brackbill and J.J. Monaghan, editors. *Proceedings of the Workshop on Particle methods in fluid dynamics and plasma physics*. Elsevier, 1987. *Comp. Phys. Commun.*, 48(1), 1987. (Cited page 131.)

[158] H. Trease, M.J. Fritts, and W.P. Crowley, editors. *Advances in the Free-Lagrange Method : including contributions on adaptive gridding and the smooth particle hydrodynamics method : Proceedings of the Next Free-Lagrange Conference*. Springer-Verlag, New York, 1990. Lecture Notes in Physics, Vol. 395. (Cited page 131.)

[159] W.P. Crowley. FLAG : A Free-Lagrange method for numerically simulating hydrodynamic flows in two dimensions. In M. Holt, editor, *Proceddings of the Second International Conference on Numerical Methods in Fluid Dynamics, September 15-19, 1970, University of California, Berkeley*, volume 8, pages 37–43, Berlin, 1971. Lecture Notes in Physics, Springer. (Cited page 131.)

[160] Stéphane Del Pino. Metric-based mesh adaptation for 2D Lagrangian compressible flows. *J. Comput. Phys.*, 230(5) :1793–1821, March 2011. (Cited page 131.)

[161] Morrell J. M., Sweby P. K., and A. Barlow. A cell by cell anisotropic adaptive mesh ALE scheme for the numerical solution of the euler equations. *J. Comput. Phys.*, 226 :1152–1180, 2007. (Cited page 131.)

[162] P. Hoch. Semi-conformal polygonal mesh adaptation seen as grid velocity formulation for ALE simulations. Conference on Numerical methods for multi-material fluid flows MULTIMAT'09 ; Pavia, Italy 2009, available at `www.math.univ-toulouse.fr/HYDRO`. (Cited page 131.)

[163] P. Hoch. A two-dimensional discontinuous ALE(DISCALE) finite volume framework on arbitrary unstructured conical meshes. Conference on Numerical methods for multi-material fluid flows MULTIMAT'11 ; Arcachon, France 2011, available at `www.math.univ-toulouse.fr/HYDRO` or `http://multimat2011.celia.u-bordeaux1.fr/Multimat2011/Tuesday_AM/Hoch.pdf`. (Cited page 131.)

[164] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2) :129–137, 1982. (Cited page 133.)

[165] S.S. Sokolov, A.I. Panov, A.A. Voropinov, I.G. Novikov, I.V. Sobolev, and A.V. Yalozo. Method TIM for computation of three-dimensional problems of mechanics of continuum media using unstructured polyhedral Lagrangian meshes (In Russian). *Questions of Atomic Science and Techniques, Series : Mathematical modeling of physical processes*, 3 :37–52, 2005. (Cited page 135.)

[166] S.S. Sokolov, A.A. Voropinov, I.G. Novikov, A.I. Panov, I.V. Sobolev, and A.A. Pushkarev. Method TIM-2D for computation of problems of mechanics of continuum media using unstructured polygonal with arbitrary connections at the nodes(In Russian). *Questions of Atomic Science and Techniques, Series : Mathematical modeling of physical processes*, 4 :29–44, 2006. (Cited page 135.)

[167] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tesselations : applications and algorithms. *SIAM Review*, 41 :637–676, 1999. (Cited page 135 et 136.)

[168] Q. Du, M. Emelianenko, and M. Gunzburger. Convergence of the Lloyd algorithm for computing centroidal Voronoi tesselations. *SAIM J. Numer. Anal.*, 44(1) :102–119, 2006. (Cited page 136.)

[169] L. I. Sedov. *Similarity and Dimensional Methods in Mechanics*. Academic Press, New York, 1959. (Cited page 137 et 159.)

[170] M. Kuchařík and M. Shashkov. Extension of efficient, swept-integration-based conservative remapping method for meshes with changing connectivity. *International Journal for Numerical Methods in Fluids*, 56(8) :1359–1365, 2007. (Cited page 137.)

[171] M. Kuchařík, J. Breil, S. Galera, P.-H. Maire, M. Berndt, and M. Shashkov. Hybrid remap for multi-material ALE. *Computers and Fluids*, 46(1) :293 – 297, 2011. 10th ICFD Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010). (Cited page 137.)

[172] Milan Kuchařík and Mikhail Shashkov. One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 231(7) :2851 – 2864, 2012. (Cited page 137.)

[173] Markus Berndt, Jérôme Breil, Stéphane Galera, Milan Kuchařík, Pierre-Henri Maire, and Mikhail Shashkov. Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 230(17) :6664 – 6687, 2011. (Cited page 137.)

[174] J.-F. Haas and B. Sturtevant. Interaction of weak-shock waves. *J. Fluid Mech.*, 181 :41–76, 1987. (Cited page 144.)

[175] P. D. Lax and X. D. Liu. Solution of two dimensional Riemann problem of gas dynamics by positive schemes. *SIAM J. Sci. Comput.*, 19(2) :319–340, 1998. (Cited page 159.)

[176] X. D. Liu and P. D. Lax. Positive schemes for solving multi-dimensional hyperbolic systems of conservation laws. *CFD Journal*, 5(1) :1–24, 1996. (Cited page 159.)

[177] S. Galera, P.-H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *J. Comput. Phys.*, 229 :5755–5787, 2010. (Cited page 164.)

[178] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39 :201–225, 1981. (Cited page 164.)

[179] H.Stewart and B. Wendroff. Two-phase flow : models and methods. *J. Comput. Phys.*, 56 :363–409, 1984. (Cited page 164.)

[180] R.E. Tipton. CALE mixed zone pressure relaxation model. Private communication, 1989. (Cited page 164.)

[181] A. Murrone and H. Guillard. A five equation reduced model for compressible two phase flow problems. *Journal of Computational Physics*, 202 :664–698, 2005. (Cited page 164.)

[182] Yu. Yanilkin Yu. Bondarenko. Computation of the thermodynamic parameters in the mixed cells in gas dynamics. *Mathematical Modeling*, 14 :63–81, 2002. (Cited page 164.)

[183] B. Despres and F. Lagoutière. Numerical resolution of a two-component compressible fluid model with interfaces. *Progress in Computational Fluid Dynamics*, 7(6) :295–310, 2007. (Cited page 164.)

[184] A. Barlow. A new Lagrangian scheme for multimaterial cells. In *Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering. ECCOMAS Computational Fluid Dynamics Conference*, 4-7 September 2001. Swansea, Wales, U.K. (Cited page 164.)

[185] Sadchikov VV. Delov VI. Comparison of several models for computation of thermodynamical parameters for heterogeneous Lagrangian cells. *VANT, Mathematical Modeling of Physical Processes*, 1 :57–70, 2005. (Cited page 164.)

[186] Yanilkin YV. Goncharov EA. New method for computations of thermodynamical states of the materials in the mixed cells. *VANT, Mathematical Modeling of Physical Processes*, 3 :16–30, 2004. (Cited page 164.)

[187] M. Shashkov. Closure models for multimaterial cells in arbitrary LagrangianâĂŞEulerian hydrocodes. *International Journal for Numerical Methods in Fluids*, 56(8) :1497–1504, 2008. Special Issue : Institute for Computational Fluid Dynamics. (Cited page 164.)

[188] James R. Kamm and Mikhail J. Shashkov. A pressure relaxation closure model for one-dimensional, two-material Lagrangian hydrodynamics based on the riemann problem. *Communications in Computational Physics*, 7 :927–976, 2010. (Cited page 164.)

[189] D.L. Youngs. Time dependent multi-material flow with large fluid distortion. In M.J. Baines (Eds.) K.W. Morton, editor, *Numerical Methods for Fluid Dynamics*, pages 274–285. Academic Press, New York,, 1982. (Cited page 164, 170 et 227.)

[190] D.L. Youngs. An interface tracking method for a 3d Eulerian hydrodynamics code. Technical Report Technical Report AWE/44/92/35, Atomic Weapon Research Establishment, Aldermaston, Berkshire, UK, 1984. (Cited page 164 et 170.)

[191] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *J. Comput. Phys.*, 121 :112–152, 1998. (Cited page 164 et 169.)

[192] Woodward P Noh WF. SLIC (simple line interface calculation). In *5th International Conference on Numerical Methods in Fluid Dynamics*, 1976. In van der Vooren AI and Zandbergen PJ. (Cited page 164.)

[193] Benson DJ. Volume of fluid interface reconstruction methods for multi-material problems. *Applied Mechanics Review*, 55(2) :151–165, 2002. (Cited page 164.)

[194] Scardovelli R. and Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31 :567–603, 1999. (Cited page 164.)

[195] Vadim Dyadechko and Mikhail Shashkov. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics*, 227(11) :5361 – 5384, 2008. (Cited page 165 et 231.)

[196] H.T. Ahn and M. Shashkov. Multi-material interface reconstruction on generalized polyhedral meshes. *J. Comput. Phys.*, 226 :2096–2132, 2007. (Cited page 165.)

[197] M. Kucharík, R. Garimella, S. Schofield, and M. Shashkov. A comparative study of interface reconstruction methods for multi-material ALE simulations. *J. Comput. Phys.*, 2009. (Cited page 165.)

[198] M.P. Allen and D.J. Tildesley. *Computer simulation of liquids*. Oxford University Press, New York,, 1987. (Cited page 166.)

[199] J.M. Haile. *Molecular dynamics simulation : elementary methods*. Wiley-Interscience, New-York, 1992. (Cited page 166.)

[200] W. Benz. *Smooth particle hydrodynamics : a review*, volume In *The Numerical modelling of nonlinear stellar pulsations : problems and prospect*, NATO ASI series. Series C, Mathematical and physical sciences of *Series C, Mathematical and physical sciences*. Kluwer Academic Publishers, 1989. (Cited page 166.)

[201] R.A. Gingold and J.J Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Monthly Notices Royal Astronomical Society*, 181 :375–389, 1977. (Cited page 166.)

[202] S.P. Schofield, R.V. Garimella, M.M. Francois, and R. Loubère. Multi-material interface reconstruction using particles and power diagrams. Technical Report LAUR-06-8740, Los Alamos National Laboratory Report, 2006. (Cited page 167.)

[203] A. Harten. ENO scheme with subcell resolution. *J. Comput. Phys.*, 83 :148, 1989. (Cited page 168.)

[204] B.K. Swartz. The second-order sharpening of blurred smooth borders. *athematics of Computation*, 52(186) :675–, 1989. (Cited page 168.)

[205] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for Euler and Navier-Stokes equations. *VKI/NASA/AGARD Special Course on Unstructured Grid Methods for Advection Dominated Flows AGARD Publication R-787*, 1995. (Cited page 169.)

[206] T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In D. Kroner, M. Ohlberger, and C. Rohde, editors, *An introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws*, pages 195–284, Berlin, 1997. Lecture Notes in Computational Science and Engineering, Springer. (Cited page 169.)

[207] S. F. Bockman. Generalizing the formula for areas of polygons to moments. *The American Mathematical Monthly*, 92 :131–, 1989. (Cited page 169.)

[208] F. Aurenhammer. Power diagrams : properties, algorithms and applications. *SIAM Journal of Computing*, 16 :78–96, 1987. (Cited page 170.)

[209] F. Aurenhammer. Voronoi diagrams - a survey of fundamental geometric data structures. *ACM Computing Surveys*, 23(3) :345–405, 1991. (Cited page 170.)

[210] Jean-Michel Ghidaglia, Anela Kumbaro, and Gérard Le Coq. On the numerical solution to two fluid models via a cell centered finite volume method. *European Journal of Mechanics - B/Fluids*, 20(6) :841 – 867, 2001. (Cited page 186 et 227.)

[211] J.-P. Braeunig. *Sur la simulation d'Ecoulements multimatériaux par une méthode Eulérienne directe avec capture d'interface en dimensions 1, 2 et 3*. PhD thesis, Ecole Normale Supérieure de Cachan, 2007. (Cited page 186 et 227.)

[212] J.-P. Braeunig, B. Desjardins, and J.-M. Ghidaglia. A totally Eulerian finite volume solver for multi-material fluid flows. *European Journal of Mechanics - B/Fluids*, 28(4) :475 – 485, 2009. (Cited page 186 et 227.)

[213] B. van Leer. Towards the ultimate conservative difference scheme. V - A second-order sequel to Godunov's method. *J. Comput. Phys.*, 32 :101–136, July 1979. (Cited page 187.)

[214] A. Harten and S. Osher. Uniformly High-Order Accurate Nonoscillatory Schemes. I. *SIAM Journal on Numerical Analysis*, 24 :279–309, April 1987. (Cited page 187.)

[215] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes,. *J. Comput. Phys.*, 71(2) :231–303, August 1987. (Cited page 187.)

[216] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *J. Comput. Phys.*, 83(1) :32–78, July 1989. (Cited page 187.)

[217] B. van Leer. Towards the Ultimate Conservation Difference Scheme. II. Monotonicity and Conservation Combined in a Second-Order Scheme. *J. Comput. Phys.*, 14 :361, March 1974. (Cited page 187.)

[218] B. van Leer. Towards the ultimate conservative difference scheme. III - Upstream-centered finite-difference schemes for ideal compressible flow. IV - A new approach to numerical convection. *J. Comput. Phys.*, 23 :263–299, March 1977. (Cited page 187.)

[219] G. D. van Albada, B. van Leer, and W. W. Roberts, Jr. A comparative study of computational methods in cosmic gas dynamics. *Astron. Astrophysics*, 108 :76–84, April 1982. (Cited page 187.)

[220] P. K. Sweby. High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws. *SIAM Journal on Numerical Analysis*, 21 :995–1011, October 1984. (Cited page 187.)

[221] Timothy J Barth and Dennis C Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA Paper*, 89(89-0366) :1–12, 1989. (Cited page 187.)

[222] V. Venkatakrishnan. Convergence to steady state solutions of the euler equations on unstructured grids with limiters. *Journal of Computational Physics*, 118(1) :120 – 130, 1995. (Cited page 187.)

[223] D. Kuzmin. On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection. *J. Comput. Phys.*, 219 :513–531, December 2006. (Cited page 187.)

[224] P. L. Roe. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics*, 18 :337–365, 1986. (Cited page 187.)

[225] G.S. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys*, 126 :202–228, 1995. (Cited page 187.)

[226] P. Hoch. An arbitrary lagrangian-eulerian strategy to solve compressible fluid flows. HAL, `http://hal.archives-ouvertes.fr/hal-00366858`, 2009. private communication. (Cited page 187.)

[227] G. Carre, S. Del Pino, K. Pichon Gostaf, E. Labourasse, and A. Shapeev. Polynomial Least-Squares reconstruction for semi-Lagrangian cell-centered hydrodynamic schemes. *ESAIM Proceedings*, 28 :100–116, 2009. (Cited page 188.)

[228] Carlo Cercignani. *The Boltzmann equation and its applications.* Applied Mathematical Sciences, Vol. 67. New York etc. : Springer-Verlag. XII, 455 p. ; DM 98.00 , 1988. (Cited page 212.)

[229] G. A. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon, Oxford, 1994. (Cited page 212.)

[230] Russel E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7(-1) :1–49, 1998. (Cited page 212.)

[231] L. Pareschi R. E. Caflisch. Towards an hybrid method for rarefied gas dynamics. *IMA Vol. App. Math.*, 135 :57–73, 2004. (Cited page 212.)

[232] Kenichi Nanbu. Direct simulation scheme derived from the Boltzmann equation. I. monocomponent gases. *Journal of the Physical Society of Japan*, 49(5) :2042–2049, 1980. (Cited page 212.)

[233] G. Russo F. Filbet. High order numerical methods for the space non-homogeneous Boltzmann equation. *J. Comput. Phys.*, 186 :457–480, 2003. (Cited page 212.)

[234] P. Bertrand F. Filbet, E. Sonnendrücker. Conservative numerical schemes for the vlasov equation. *J. Comput. Phys.*, 172 :166–187, 2001. (Cited page 212 et 213.)

[235] L. Mieussens. Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamic. *Math. Models Meth. App. Sci.*, 10 :1121–1149, 2000. (Cited page 212 et 213.)

[236] Andrzej Palczewski, Jacques Schneider, and Alexandre V. Bobylev. A consistency result for a discrete-velocity model of the Boltzmann equation. *SIAM J. Numer. Anal.*, 34(5) :1865–1883, October 1997. (Cited page 212.)

[237] G. Puppo S. Pieraccini. Implicit-explicit schemes for BGK kinetic equations. *SIAM J. Sci. Comp.*, 32 :1–28, 2007. (Cited page 212.)

[238] Nicolas Crouseilles, Thomas Respaud, and Eric Sonnendrücker. A Forward semi-Lagrangian Method for the Numerical Solution of the Vlasov Equation. *Computer Physics Communications*, 180(10), 2009. (Cited page 212 et 213.)

[239] Nicolas Crouseilles, Michel Mehrenberger, and Eric Sonnendrücker. Conservative semi-Lagrangian schemes for Vlasov equations. *J. Comput. Phys.*, pages 1927–1953, 2010. (Cited page 212 et 213.)

[240] J. Schneider A.V. Bobylev, A. Palczewski. On approximation of the Boltzmann equation by discrete velocity models. *C. R. Acad. Sci. Paris Ser. I. Math.*, 320 :639–644, 1995. (Cited page 212.)

[241] J. Schneider A. Palczewski. Existence, stability, and convergence of solutions of discrete velocity models to the Boltzmann equation. *J. Statist. Phys.*, 91 :307–326, 1998. (Cited page 212.)

[242] S. Mischler L. Desvillettes. About the splitting algorithm for Boltzmann and BGK equations. *Math. Mod. & Meth. in App. Sci.*, 6 :1079–1101, 1996. (Cited page 212.)

[243] G. Strang. On the construction and the comparison of difference schemes. *SIAM J. Numer. Anal.*, pages 506–517, 1968. (Cited page 212 et 213.)

[244] Jun S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2001. (Cited page 213.)

Ce document a été préparé à l'aide de l'éditeur de texte GNU Emacs et du logiciel de composition typographique LaTeX 2ε. Les figures ont été réalisées avec le logiciel Xfig, les résultats numériques ont été majoritairement réalisés à l'aide de Gnuplot et Paraview.

Titre Contribution au domaine des méthodes numériques Lagrangiennes et Arbitrary-Lagrangian-Eulerian

Résumé Ce mémoire présente des travaux portant (i) sur les méthodes numériques lagrangiennes et (ii) sur le développement des méthodes dites arbitrairement lagrangienne-eulérienne (ALE). Ces deux thématiques ont en commun de tenter de résoudre les équations de la mécanique des fluides compressibles en multi-dimensions sur des maillages mobiles se déplaçant soit à la vitesse du fluide (lagrangienne), soit à une vitesse arbitraire (ALE). En particulier nous abordons les problèmes de viscosité artificielle, de consistance et précision, de stabilité, de consistance en volume, le traitement des points exceptionnels ou encore les lignes de glissement. Dans le chapitre ALE nous proposons des études sur les phase de projection conservative, correction a posteriori, reconnexion topologique de maillage ou de reconstruction d'interface dans des mailles mixtes. (iii) Une troisième partie propose un ensemble de sujets plus hétéroclites : reconstruction d'interface dans des schémas multi-matériaux sur maillage fixe, schémas cinétiques ultra rapides, et des schémas de type volumes finis d'ordre très élevé.

Mots-clés Schéma lagrangien, ALE, projection, remaillage, reconnection de maillages, tesselation de Voronoi, méthode MOOD, schéma cinétique, 3D, reconstruction d'interfaces

Title Contribution to Lagrangian and Arbitrary-Lagrangian-Eulerian numerical schemes

Abstract This thesis presents our work related to (i) Lagrangian schemes and (ii) Arbitrary-Lagrangian-Eulerian numerical methods (ALE). Both types of methods have in commun to solve the multidimension compressible Euler equations on a moving grid. The grid moves with either the fluid velocity (Lagrangian) or an arbitrary velocity (ALE). More specifically we deal with some problems related to artifical viscosity, internal consistency, stability, accuracy, exceptional points and slide line treatments. In the ALE chapter we study the remap and rezone phases but also the mesh reconnection to build a ReconnectionALE scheme and further some interface reconstruction techniques. In a third chapter (iii) other resarch topics are presented like interface reconstruction techniques on a fixed grid finite volume scheme, ultra fast kinetic scheme, and very high-order finite volume schemes.

Keywords Lagrangian scheme, Arbitrary-Lagrangian-Eulerian scheme, remap, rezone, reconnect, repair, Voronoi tesselation, MOOD method, kinetic scheme, 3D, interface reconstruction.