– **Typeset by GMNI &** FoilTEX –

SHARK-FV 2015 Conference
SHARING HIGHER-ORDER ADVANCED RESEARCH KNOW-HOW on FINITE VOLUME
Ofir, Portugal
May 18 - 22, 2015

HIGHER-ORDER FV-MLS METHOD FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

L. Ramírez[a], X. Nogueira[a], S. Khelladi[b], J.C Chassaing[c], I. Colominas[a]

a: Dept. of Applied Mathematics
Civil Engineering School
University of A Coruña, Spain

b: Arts et Métiers Paris Tech
151 Boulevard de l'Hôpital
75013 Paris, France

c:Institute Jean Le Rond d'Alembert
Case 162, 4 Place Jussieu
75252 Paris, France

e-mail: lramirez@udc.es
web page: http://caminos.udc.es/gmni

Outline

- Introduction
- The FV-MLS method
- A high-order formulation for incompressible flows
- High-order Fluid-Structure-Interaction techniques
- Conclusions

- Introduction
- The FV-MLS method
- A high-order formulation for incompressible flows
- High-order Fluid-Structure-Interaction techniques
- Conclusions

- FV-MLS Applications
  - All-speed Navier-Stokes
  - Incompressible Navier-Stokes
  - Linearized Euler Equations (acoustics)
  - Navier-Stokes Korteweg equations
  - Turbulence (ILES)
  - High-order Sliding mesh applications
  - Cavitating flows
  - ...

FV-MLS Applications

- FV-MLS Applications
  - All-speed Navier-Stokes
  - Incompressible Navier-Stokes
  - Linearized Euler Equations (acoustics)
  - Navier-Stokes Korteweg equations
  - Turbulence (ILES)
  - High-order Sliding mesh applications
  - Cavitating flows
  - ...

Wednesday @11:00 by Xesús Nogueira

- Introduction
- The FV-MLS method
- A high-order formulation for incompressible flows
- High-order Fluid-Structure-Interaction techniques
- Conclusions

The Finite Volume Method

- Let us consider a generic conservation law for the 2D domain Ω_T

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{F}} = S$$

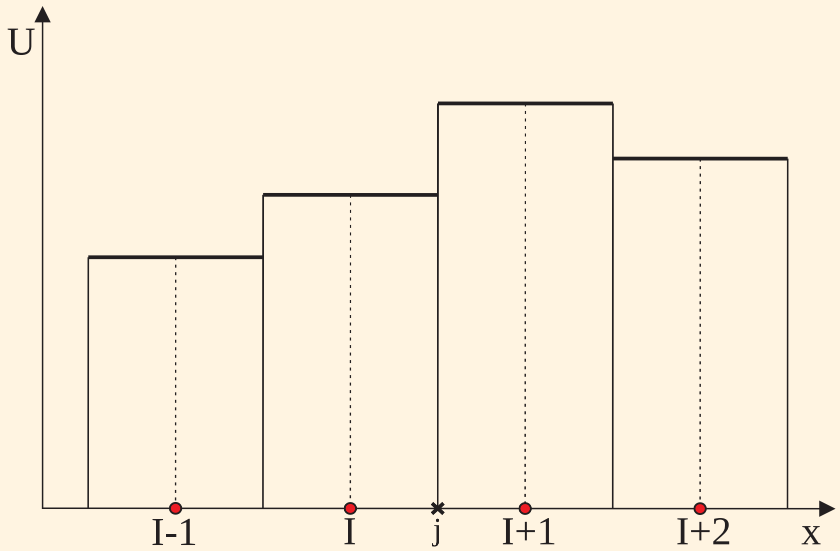
- Finite Volume discretization over Ω_I:

$$\Omega_I \int_{\Omega_I} \frac{\partial \boldsymbol{U}_I}{\partial t} d\Omega + \int_{\Gamma_I} \left(\boldsymbol{\mathcal{F}}^{\boldsymbol{\mathcal{H}}} - \boldsymbol{\mathcal{F}}^{\boldsymbol{\mathcal{V}}} \right) \cdot \boldsymbol{n} \, d\Gamma = \boldsymbol{0}$$

  - $\boldsymbol{\mathcal{F}}^{\boldsymbol{\mathcal{H}}} \rightarrow$ Hyperbolic-like term
  - $\boldsymbol{\mathcal{F}}^{\boldsymbol{\mathcal{V}}} \rightarrow$ Elliptic-like term

The Finite Volume Method

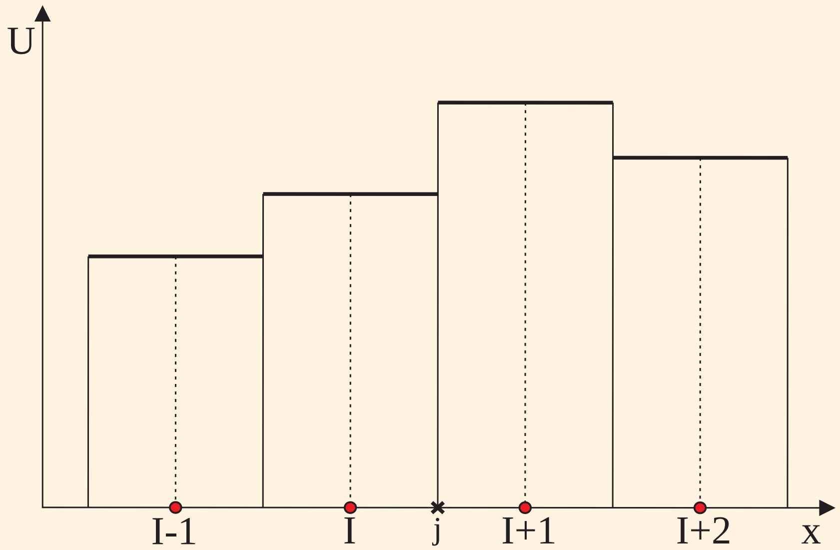
- Hyperbolic term:
  - Godunov approach



  - $\mathcal{F}^{\mathcal{H}}$ is the solution of a Riemann problem
  - Initial values $\rightarrow$ variables at both sides of the interface.

The Finite Volume Method

- Hyperbolic term:
  - Godunov approach

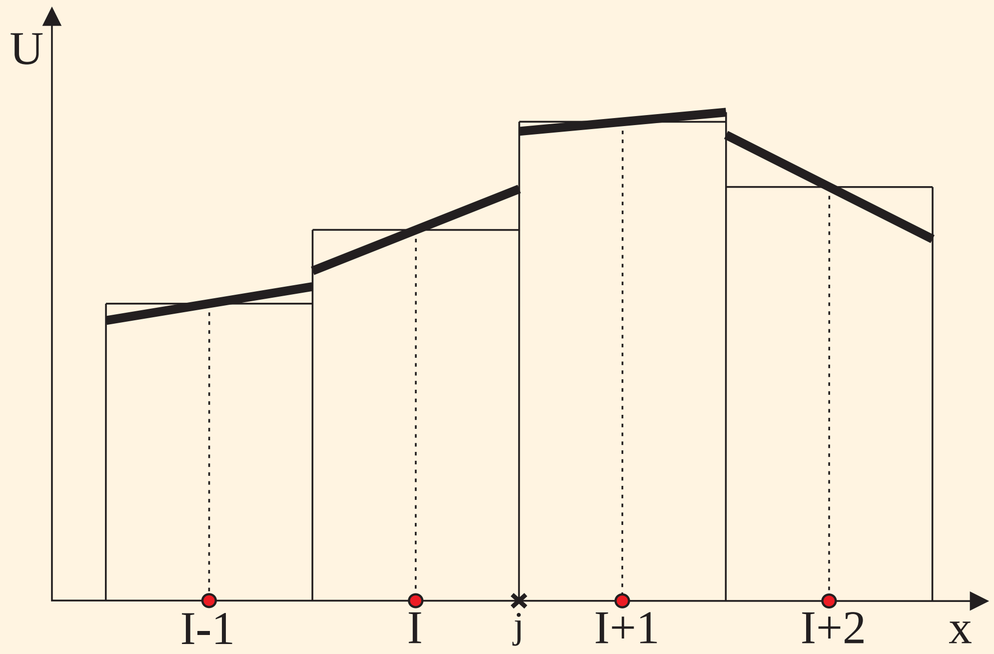


  - $\mathcal{F}^{\mathcal{H}}$ is the solution of a Riemann problem
  - Initial values → variables at both sides of the interface.

$$\boldsymbol{U}_L \neq \boldsymbol{U}_R$$

SHARK-FV 2015 Conference
SHARING HIGHER-ORDER ADVANCED RESEARCH KNOW-HOW on FINITE VOLUME
Ofir, Portugal
May 18 - 22, 2015

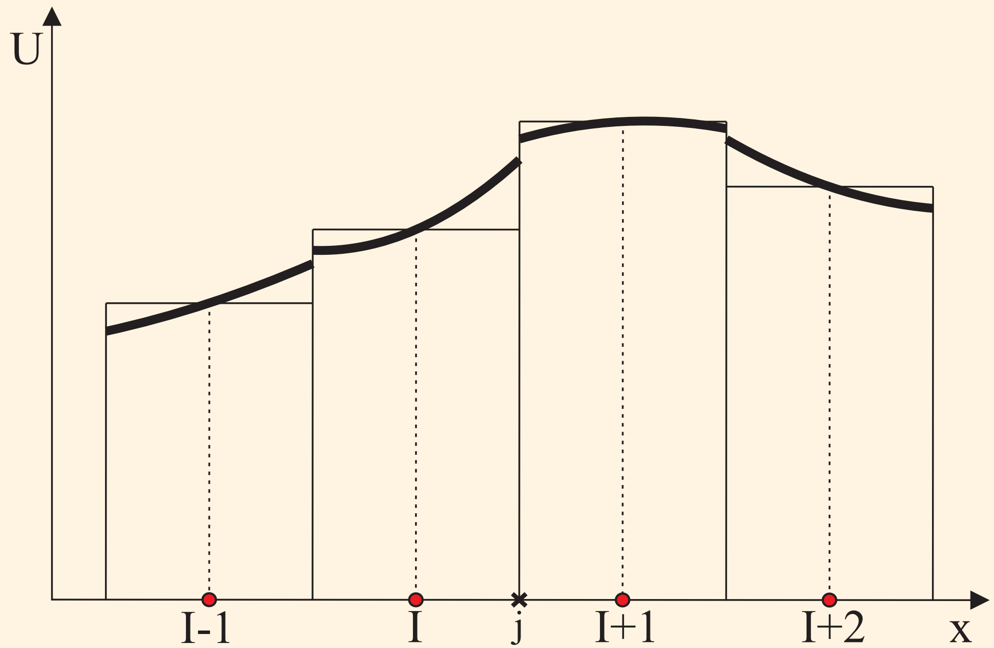
The Finite Volume Method



$$\boldsymbol{U}_L = \boldsymbol{U}_I + \nabla \boldsymbol{U}_I \cdot (\boldsymbol{x}_j - \boldsymbol{x}_I)$$

$$\boldsymbol{U}_R = \boldsymbol{U}_{I+1} + \nabla \boldsymbol{U}_{I+1} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_{I+1})$$

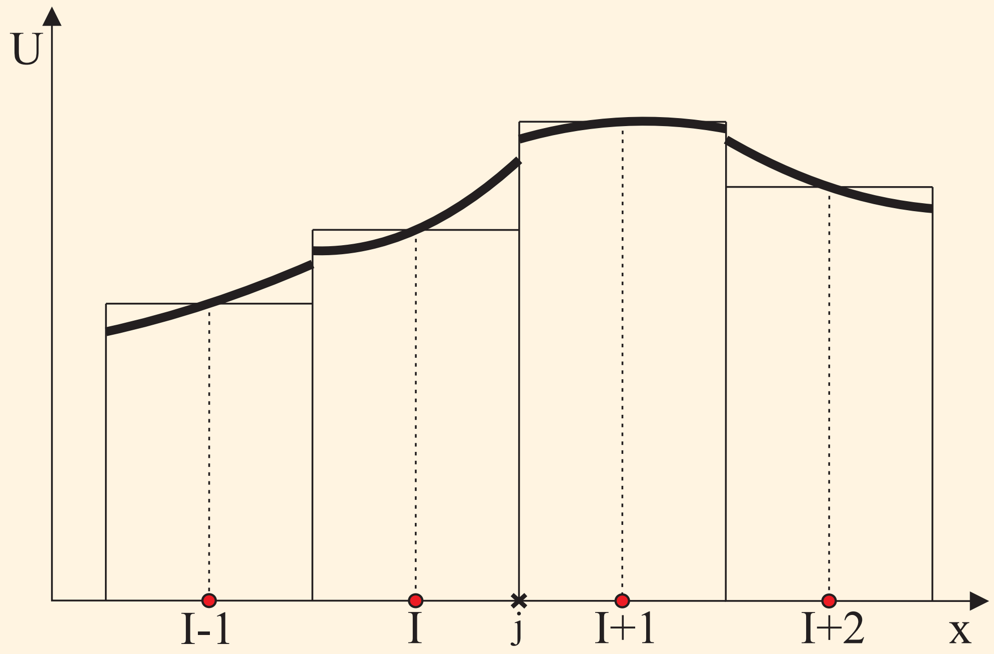
The Finite Volume Method



$$\boldsymbol{U}_L = \boldsymbol{U}_I + \nabla \boldsymbol{U}_I \cdot (\boldsymbol{x}_j - \boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{x}_I)^T \boldsymbol{H}_I \, (\boldsymbol{x}_j - \boldsymbol{x}_I)$$

$$\boldsymbol{U}_R = \boldsymbol{U}_{I+1} + \nabla \boldsymbol{U}_{I+1} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_{I+1}) + \frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{x}_{I+1})^T \boldsymbol{H}_{I+1} \, (\boldsymbol{x}_j - \boldsymbol{x}_{I+1})$$

The Finite Volume Method



$$\boldsymbol{U}_L = \boldsymbol{U}_I + \nabla \boldsymbol{U}_{\boldsymbol{I}} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_I) + \frac{1}{2} (\boldsymbol{x}_j - \boldsymbol{x}_I)^T \boldsymbol{H}_{\boldsymbol{I}} (\boldsymbol{x}_j - \boldsymbol{x}_I)$$

$$\boldsymbol{U}_R = \boldsymbol{U}_{I+1} + \nabla \boldsymbol{U}_{\boldsymbol{I+1}} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_{I+1}) + \frac{1}{2} (\boldsymbol{x}_j - \boldsymbol{x}_{I+1})^T \boldsymbol{H}_{\boldsymbol{I+1}} (\boldsymbol{x}_j - \boldsymbol{x}_{I+1})$$

Finite Volume Method. High-order schemes (II)

- Computation of high-order derivatives:
  - Easy on structured grids.
  - Unstructured grids⇒PROBLEM.
- We propose:
  - The use of Moving Least Squares (MLS) to obtain an accurate and multidimensional approximation of derivatives on unstructured grids.

- Kernel approximation is based on the properties of Dirac's Delta distribution:

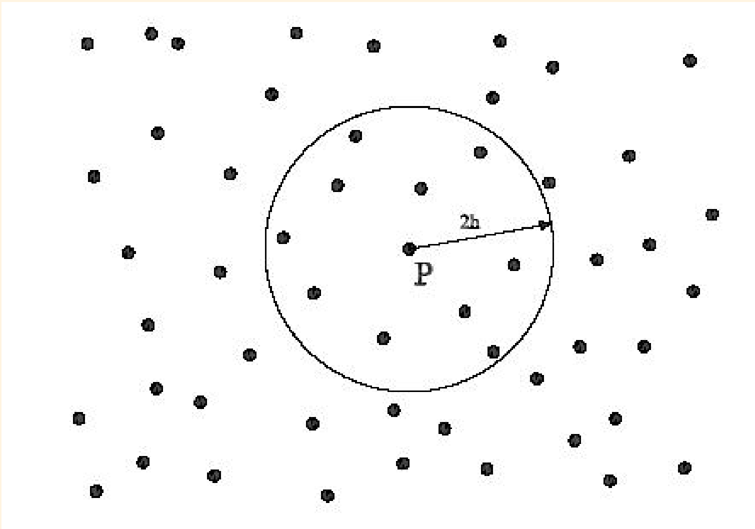
$$u(\boldsymbol{x}) = \int_{\boldsymbol{y}\in\Omega} u(\boldsymbol{y})\delta(\boldsymbol{x}-\boldsymbol{y})d\Omega$$

- Kernel approximation is defined as:

$$u^h(\boldsymbol{x}) = \int_{\boldsymbol{y}\in\Omega} u(\boldsymbol{y})W(\boldsymbol{x}-\boldsymbol{y},\rho)d\Omega$$

The basis: Kernel approximations (II)

- In discrete form:



$$\hat{u}(\boldsymbol{x}) = \sum_{j=1}^{n} u_j W(\boldsymbol{x} - \boldsymbol{x}_j, h) V_j$$

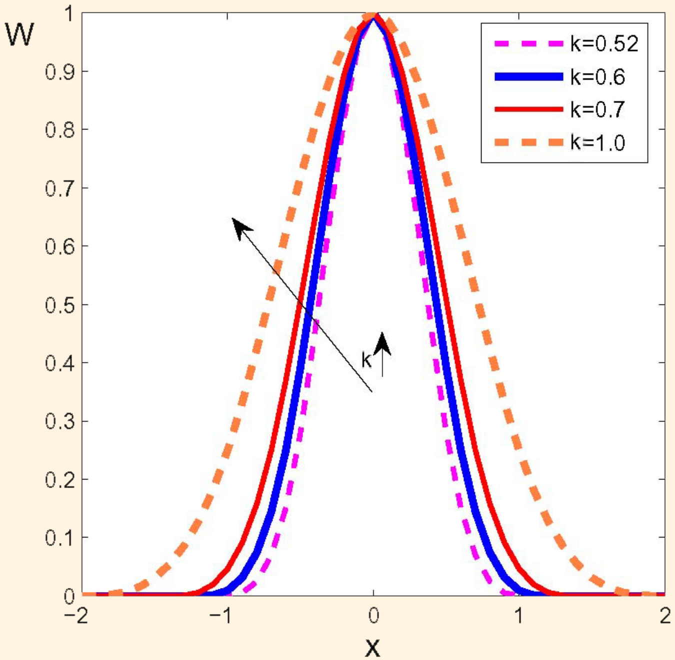
- V_j is the statistical volume of a particle j.
- Compact support with $r = 2h$
- h is the smoothing length.

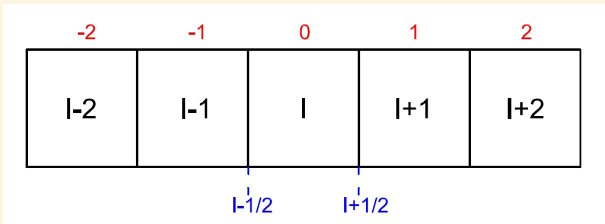
The basis: Kernel approximations (IV)

- Many functions used as kernels: splines, gaussians
- An example, the cubic spline:

$$W_j(\boldsymbol{x}) = W(\boldsymbol{x}-\boldsymbol{x}_j, h) = \frac{\alpha}{h^\nu}\begin{cases} 1-\frac{3}{2}s^2+\frac{3}{4}s^3 & s \leq 1 \\ \frac{1}{4}(2-s)^3 & 1 < s \leq 2 \\ 0 & s > 2 \end{cases}$$

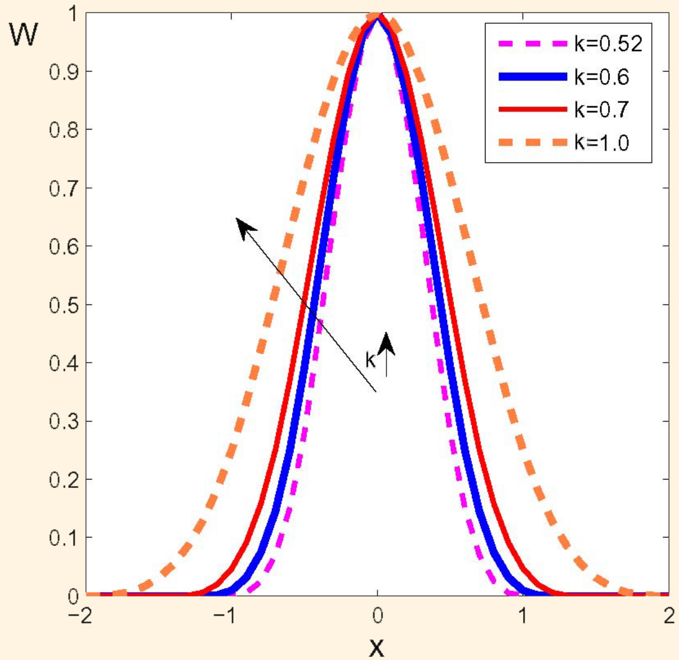
$$s = \frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|}{h} \qquad h = k \max\left(\|\boldsymbol{x}-\boldsymbol{x}_j\|\right)$$



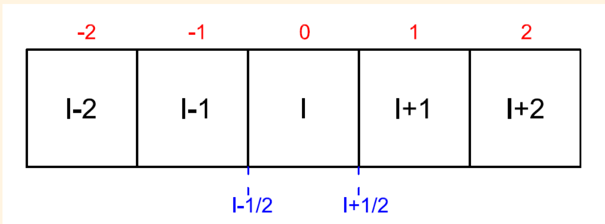


- Many functions used as kernels: splines, gaussians
- An example, the cubic spline:

$$W_j(\boldsymbol{x}) = W(\boldsymbol{x} - \boldsymbol{x}_j, h) = \frac{\alpha}{h^\nu} \begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3 & s \leq 1 \\ \frac{1}{4}(2-s)^3 & 1 < s \leq 2 \\ 0 & s > 2 \end{cases}$$



$$s = \frac{\|\boldsymbol{x} - \boldsymbol{x}_j\|}{h} \qquad h = \boldsymbol{k} \max\left(\|\boldsymbol{x} - \boldsymbol{x}_j\|\right)$$



The basis: Kernel approximations (V)

- Another example: Exponential Kernel.

$$W(x, x^*, \kappa) = \frac{e^{-\left(\frac{s}{c}\right)^2} - e^{-\left(\frac{dm}{c}\right)^2}}{1 - e^{-\left(\frac{dm}{c}\right)^2}}$$

$$s = |x - x^*|\,, d_m = 2\max\left(|x_j - x^*|\right), c = \frac{d_m}{2\kappa}$$

- 2D kernel$\Rightarrow$ product of two 1D kernels:

$$W_j(\boldsymbol{x}, \boldsymbol{x}^*, \kappa_x, \kappa_y) = W_j(x, x^*, \kappa_x) W_j(y, y^*, \kappa_y)$$

- Another example: Exponential Kernel.

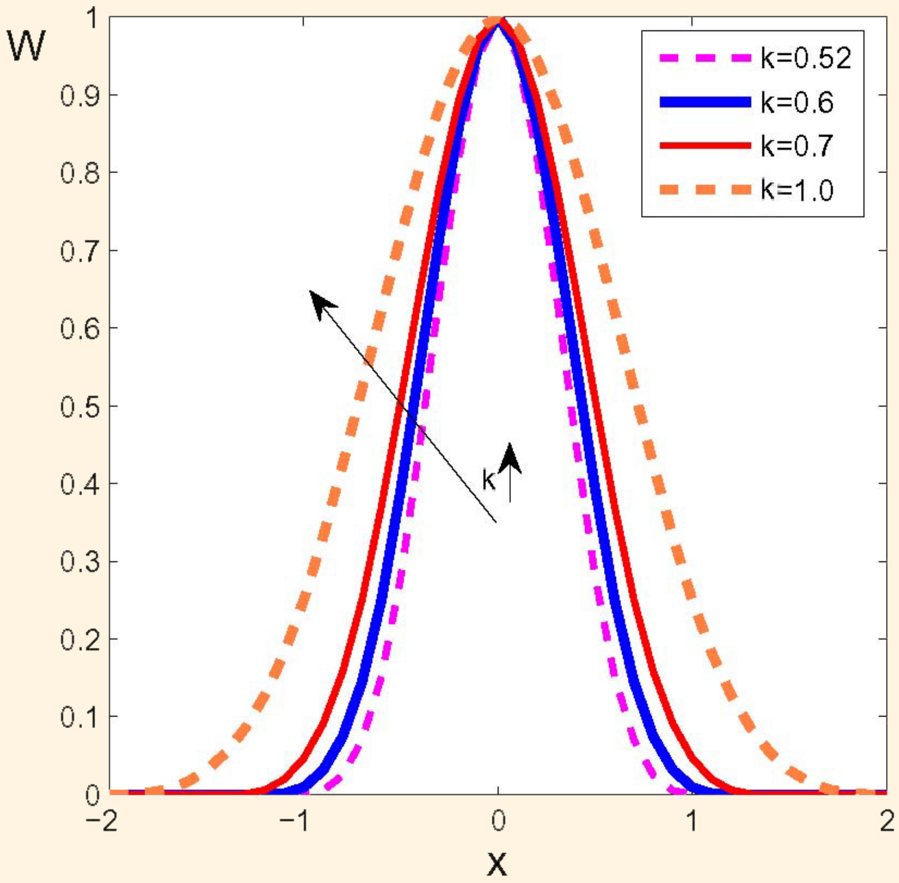
$$W(x, x^*, \boldsymbol{\kappa}) = \frac{e^{-\left(\frac{s}{c}\right)^2} - e^{-\left(\frac{dm}{c}\right)^2}}{1 - e^{-\left(\frac{dm}{c}\right)^2}}$$

$$s = |x - x^*|, d_m = 2\max\left(|x_j - x^*|\right), c = \frac{d_m}{2\boldsymbol{\kappa}}$$

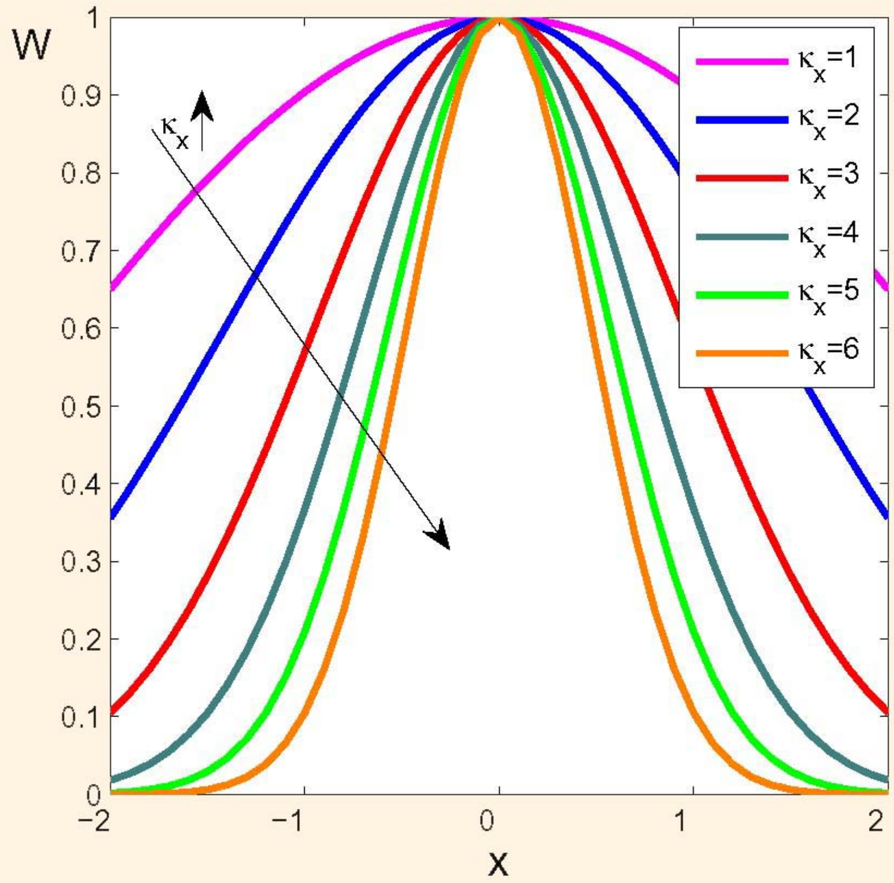
- 2D kernel$\Rightarrow$ product of two 1D kernels:

$$W_j(\boldsymbol{x}, \boldsymbol{x}^*, \kappa_x, \kappa_y) = W_j(x, x^*, \kappa_x) W_j(y, y^*, \kappa_y)$$

The basis: Kernel approximations (VI)



CUBIC SPLINE



EXPONENTIAL KERNEL

Moving Least Squares (I)

- Reconstruction of $u(\boldsymbol{x})$ at a point $\boldsymbol{x}$ by using a weighted LS approximation in the vicinity of $\boldsymbol{x}$:

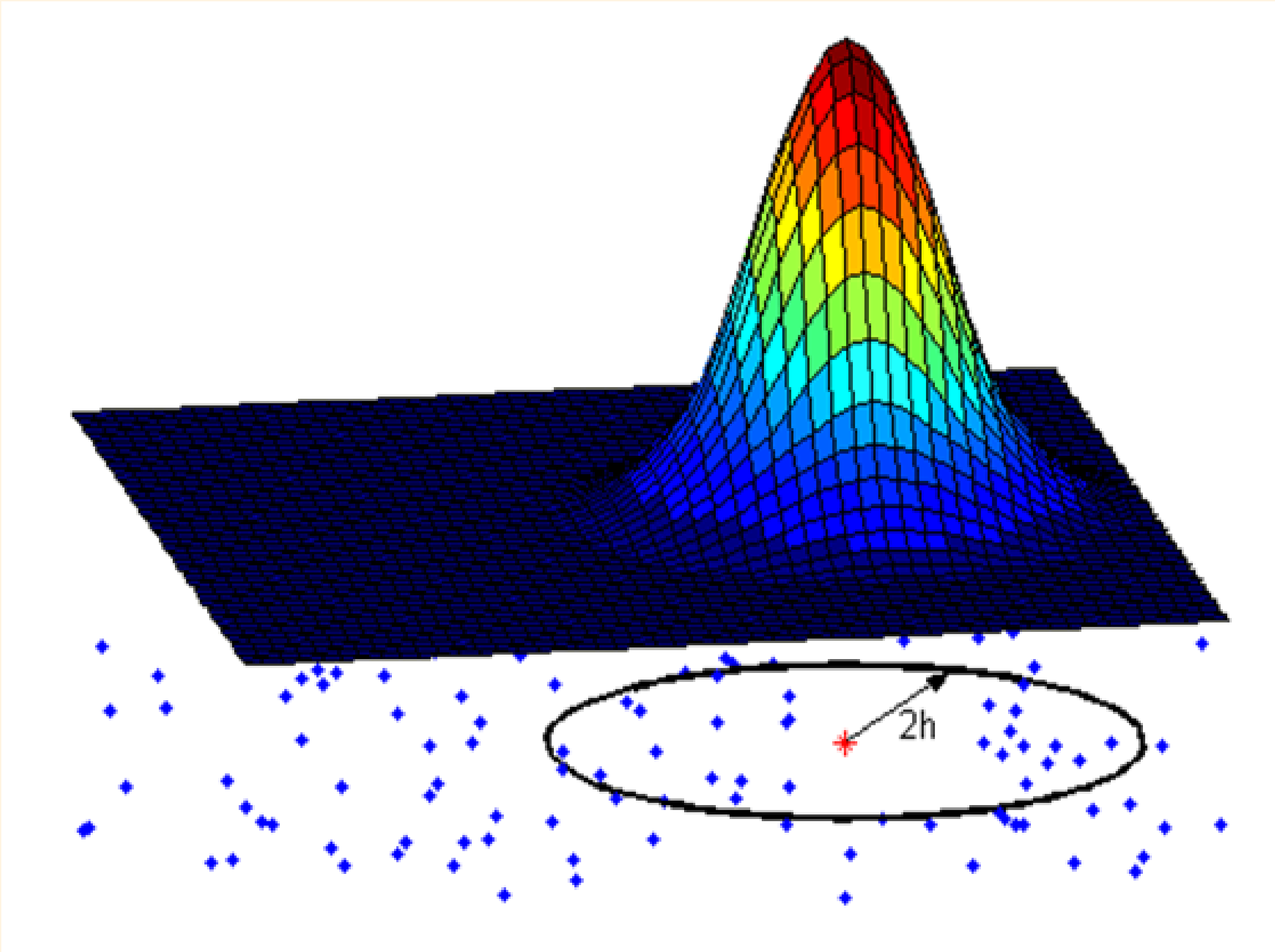
$$u(\boldsymbol{x}) \approx \hat{u}(\boldsymbol{x}) = \sum_{i=1}^{m} p_i(\boldsymbol{x})\alpha_i(\boldsymbol{z})\,|_{\boldsymbol{z}=\boldsymbol{x}} = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{\alpha}(\boldsymbol{z})\,|_{\boldsymbol{z}=\boldsymbol{x}}$$

  - $\boldsymbol{p}^T(\boldsymbol{x})$: base of functions with dimension m.
  - $\boldsymbol{\alpha}(\boldsymbol{z})\,|_{\boldsymbol{z}=\boldsymbol{x}}$: Parameters that minimize the error functional:

$$J(\boldsymbol{\alpha}(\boldsymbol{z})\,|_{\boldsymbol{z}=\boldsymbol{x}}) = \int_{\boldsymbol{y}\in\Omega_{\mathrm{x}}} W(\boldsymbol{z}-\boldsymbol{y},h)\,|_{\boldsymbol{z}=\boldsymbol{x}} \left[u(\boldsymbol{y}) - \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{\alpha}(\boldsymbol{z})\,|_{\boldsymbol{z}=\boldsymbol{x}}\right]^2 d\Omega_{\boldsymbol{x}}$$

  - $W(\boldsymbol{z}-\boldsymbol{y},h)\,|_{\boldsymbol{z}=\boldsymbol{x}}$: kernel (smoothing function) with compact support ($\Omega_{\boldsymbol{x}}$) centered in $\boldsymbol{z}=\boldsymbol{x}$.
  - h: smoothing length.

Moving Least Squares (III)



Moving Least Squares (IV)

- Minimization of J leads to:

$$\int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} \boldsymbol{p}(\boldsymbol{y}) W(\boldsymbol{z}-\boldsymbol{y},h)\Big|_{\boldsymbol{z}=\boldsymbol{x}} u(\boldsymbol{y})\, d\Omega_{\boldsymbol{x}} = \boldsymbol{M}(\boldsymbol{x})\boldsymbol{\alpha}(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{x}}$$

- $\boldsymbol{M}(\boldsymbol{x})$ is the *moment matrix* defined as:

$$\boldsymbol{M}(\boldsymbol{x}) = \int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} \boldsymbol{p}(\boldsymbol{y}) W(\boldsymbol{z}-\boldsymbol{y},h)\Big|_{\boldsymbol{z}=\boldsymbol{x}} \boldsymbol{p}^T(\boldsymbol{y})$$

Moving Least Squares (III)

- In practice, Ω is a set of scattered points. Previous integrals are evaluated using points in $\Omega_{\boldsymbol{x}}$ as quadrature points:

$$\boldsymbol{\alpha}(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{x}} = \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}}$$

  - $\boldsymbol{u}_{\Omega_{\mathrm{x}}}$ contains nodal values of the function $u_{\boldsymbol{x}}$ to be approximated, at $n_{\boldsymbol{x}}$ nodes in $\Omega_{\boldsymbol{x}}$

$$\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \left(u(\boldsymbol{x}_1)\; u(\boldsymbol{x}_2)\; \cdots \; u\left(\boldsymbol{x}_{n_{\boldsymbol{x}}}\right)\right)^T$$

Moving Least Squares (IV)

- Discrete expression of the moment matrix is a $m \times m$ matrix equals to $\boldsymbol{M}(\boldsymbol{x}) = \boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{P}^T_{\Omega_{\boldsymbol{x}}}$
  - $\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}$(dimension $m \times n_{\boldsymbol{x}}$), and $\boldsymbol{W}(\boldsymbol{x})$(dimension $n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}$) are obtained by

$$\boldsymbol{P}_{\Omega_{\boldsymbol{x}}} = \left(\boldsymbol{p}\left(\boldsymbol{x}_1\right)\ \boldsymbol{p}\left(\boldsymbol{x}_2\right)\ \cdots\ \boldsymbol{p}\left(\boldsymbol{x}_{n_{\boldsymbol{x}}}\right)\right)$$

$$\boldsymbol{W}(\boldsymbol{x}) = diag\left\{W_i\left(\boldsymbol{x} - \boldsymbol{x}_i\right)\right\} \quad i = 1, \ldots, n_{\boldsymbol{x}} \qquad (1)$$

- Finally, MLS approximation is written by:

$$\widehat{u}(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \boldsymbol{N}^T(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \sum_{j=1}^{n_{\boldsymbol{x}}} N_j(\boldsymbol{x})u_j$$

Moving Least Squares (V)

- Interpolation can be written as:

$$\hat{u}(\boldsymbol{x}) = \sum_{j=1}^{n_{\mathbf{x}}} N_j(\boldsymbol{x}) u_j$$

with

$$\boldsymbol{N}^T(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}(\boldsymbol{x})$$

- N_j can be considered as "shape functions".
- N_j depends on the number of neighbors, the kernel and the base ($\boldsymbol{p}^T$).
- N_j is a function of the grid.

Moving Least Squares (VI)

- A practical note about the polynomial basis

$$\boldsymbol{p}(\boldsymbol{x}) = \begin{pmatrix} 1 & x & y & xy & x^2 & y^2 \end{pmatrix}^T$$

  - We define locally and scale the monomials of the basis
  - Better conditioning of the momentum matrix
  - If MLS shape functions $\boldsymbol{N}(\boldsymbol{x})$ are evaluated at a point $\boldsymbol{x}_I$, the basis is evaluated at $\frac{\boldsymbol{x}-\boldsymbol{x}_I}{h}$
  - Then we can write:

$$\boldsymbol{N}^T(\boldsymbol{x}_I) = \boldsymbol{p}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x}_I)\boldsymbol{P}_{\Omega_{\boldsymbol{x}_I}}\boldsymbol{W}(\boldsymbol{x}_I) = \boldsymbol{p}^T(\boldsymbol{0})\boldsymbol{C}(\boldsymbol{x}_I)$$

with

$$\boldsymbol{C}(\boldsymbol{x}_I) = \boldsymbol{M}^{-1}(\boldsymbol{x}_I)\boldsymbol{P}_{\Omega_{\boldsymbol{x}_I}}\boldsymbol{W}(\boldsymbol{x}_I)$$

Moving Least Squares (VII)

- Computation of derivatives
  - First derivatives

$$\frac{\partial \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x} = \frac{\partial \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x}\boldsymbol{C}(\boldsymbol{x}) + \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x}$$

  - Second derivatives

$$\begin{aligned}
\frac{\partial^2 \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^2} &= \frac{\partial^2 \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^2}\boldsymbol{C}(\boldsymbol{x}) + 2\frac{\partial \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x}\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x} + \boldsymbol{p}(\boldsymbol{x})\frac{\partial^2 \boldsymbol{C}(\boldsymbol{x})}{\partial x^2} \\
\frac{\partial^2 \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x \partial y} &= \frac{\partial^2 \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x \partial y}\boldsymbol{C}(\boldsymbol{x}) + \frac{\partial \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x}\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial y} + \frac{\partial \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial y}\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x} \\
&+ \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\frac{\partial^2 \boldsymbol{C}(\boldsymbol{x})}{\partial x \partial y}
\end{aligned}$$

Moving Least Squares (VII)

- where $\dfrac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x}$ is given by

$$\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x} = \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial \boldsymbol{W}(\boldsymbol{x})}{\partial x}\left(\boldsymbol{I} - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})\right)$$

- and the second derivatives of $\boldsymbol{C}(\boldsymbol{x})$

$$\begin{aligned}
\frac{\partial^2 \boldsymbol{C}(\boldsymbol{x})}{\partial x^2} = & \frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x}\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial W}{\partial x}\left(\boldsymbol{I} - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})\right) \\
& + \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial^2 W(\boldsymbol{x})}{\partial x^2}\left(\boldsymbol{I} - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})\right) \\
& - \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial W(\boldsymbol{x})}{\partial x}\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial W(\boldsymbol{x})}{\partial x}\left(\boldsymbol{I} - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})\right) \\
& - \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial W(\boldsymbol{x})}{\partial x}\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x}
\end{aligned}$$

Moving Least Squares (VII)

$$
\begin{aligned}
\frac{\partial^2 \boldsymbol{C}(\boldsymbol{x})}{\partial x \partial y} = & \frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial y} \boldsymbol{W}^{-1}(\boldsymbol{x}) \frac{\partial W(\boldsymbol{x})}{\partial x} \left(\boldsymbol{I} - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{C}(\boldsymbol{x}) \right) \\
& + \boldsymbol{C}(\boldsymbol{x}) \boldsymbol{W}^{-1}(\boldsymbol{x}) \frac{\partial^2 W(\boldsymbol{x})}{\partial x \partial y} \left(\boldsymbol{I} - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{C}(\boldsymbol{x}) \right) \\
& - \boldsymbol{C}(\boldsymbol{x}) \boldsymbol{W}^{-1}(\boldsymbol{x}) \frac{\partial W(\boldsymbol{x})}{\partial y} \boldsymbol{W}^{-1}(\boldsymbol{x}) \frac{\partial W(\boldsymbol{x})}{\partial x} \left(\boldsymbol{I} - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{C}(\boldsymbol{x}) \right) \\
& - \boldsymbol{C}(\boldsymbol{x}) \boldsymbol{W}^{-1}(\boldsymbol{x}) \frac{\partial W(\boldsymbol{x})}{\partial x} \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial y}
\end{aligned}
$$

Moving Least Squares (VII)

- Computation of derivatives
  - The *diffuse* derivatives are obtained by neglecting all derivatives of $\boldsymbol{C}(\boldsymbol{x})$

$$\frac{\partial^2 \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^2} \approx \frac{\partial^2 \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^2}\boldsymbol{C}(\boldsymbol{x})$$

$$\frac{\partial^2 \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x \partial y} \approx \frac{\partial^2 \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x \partial y}\boldsymbol{C}(\boldsymbol{x})$$

$$\frac{\partial^n \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^n} \approx \frac{\partial^n \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^n}\boldsymbol{C}(\boldsymbol{x})$$

Huerta et al.,*Pseudo-divergence-free Element Free Galerkin method for incompressible fluid flow*,CMAME,2004

Moving Least Squares (VII)

- However, as we already have computed the first derivatives of $\boldsymbol{C}(\boldsymbol{x})$, it is possible to use a *semi-diffuse* approach without extra effort:

$$\frac{\partial^2 \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^2} \approx \frac{\partial^2 \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^2}\boldsymbol{C}(\boldsymbol{x}) + 2\frac{\partial \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x}\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x}$$

$$\frac{\partial^2 \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x \partial y} \approx \frac{\partial^2 \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x \partial y}\boldsymbol{C}(\boldsymbol{x}) + \frac{\partial \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial x}\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial y} + \frac{\partial \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})}{\partial y}\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x}$$

Derivatives	L_1 Error	L_2 Error
Diffuse	1.631×10^{-5}	4.784×10^{-5}
Semi-Diffuse	1.586×10^{-5}	4.710×10^{-5}
Full	1.288×10^{-5}	3.656×10^{-5}

- ▷ It has been proved that use of diffuse or semi-diffuse derivatives does not decrease the *order of accuracy*.
- ▷ It should be noted that the *accuracy* is affected

Accuracy assesment of a high-order moving least squares finite volume method for compressible flows,C&F,2013

- Note that due to the local scaling $\frac{\boldsymbol{x}-\boldsymbol{x}_I}{h}$

$$\boldsymbol{p}^{\mathrm{T}}(\mathbf{0}) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial \boldsymbol{p}^{\mathrm{T}}(\mathbf{0})}{\partial x} = \begin{pmatrix} 0 & \frac{1}{h} & 0 & 0 & 0 & 0 \end{pmatrix}$$

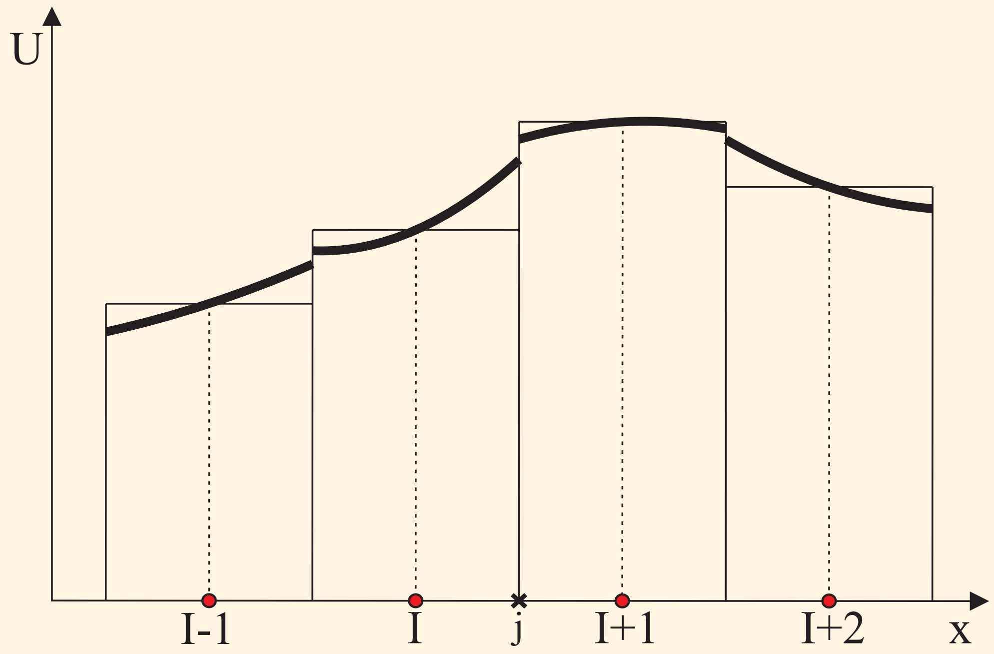
$$\frac{\partial \boldsymbol{p}^{\mathrm{T}}(\mathbf{0})}{\partial y} = \begin{pmatrix} 0 & 0 & \frac{1}{h} & 0 & 0 & 0 \end{pmatrix}$$

The FV-MLS method

- This scheme acknowledge the different nature of convective and diffusive terms.
- We start from a high-order, continuous MLS approximation of the solution:
- Convective terms discretization:
  - Breaks the continuous representation of the MLS approximation.
  - Obtains a continuous representation of the variables inside each cell.
- Diffusive terms discretization is:
  - Centered$\rightarrow$ Direct interpolation at Gauss points with MLS.
  - Continuous.
  - Highly accurate.

The FV-MLS method

- Hyperbolic term → Flux Difference Splitting

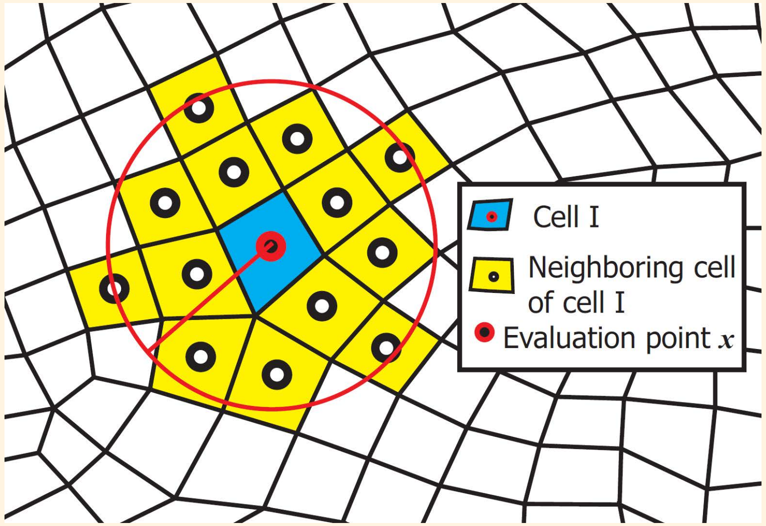


$$\boldsymbol{U}_L = \boldsymbol{U}_I + \nabla \boldsymbol{U}_{\boldsymbol{I}} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{x}_I)^T \boldsymbol{H}_{\boldsymbol{I}} (\boldsymbol{x}_j - \boldsymbol{x}_I)$$

$$\boldsymbol{U}_R = \boldsymbol{U}_{I+1} + \nabla \boldsymbol{U}_{\boldsymbol{I+1}} \cdot (\boldsymbol{x}_j - \boldsymbol{x}_{I+1}) + \frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{x}_{I+1})^T \boldsymbol{H}_{\boldsymbol{I+1}} (\boldsymbol{x}_j - \boldsymbol{x}_{I+1})$$

The FV-MLS method

- Hyperbolic-like terms:
  - MLS is used to compute the gradients and high-order derivatives required for the reconstruction of the variable at integration points placed at interface.



$$\nabla \boldsymbol{U}_I = \sum_{j=1}^{n_{\mathbf{x}}} \boldsymbol{U}_j \nabla \boldsymbol{N}_j(\boldsymbol{x}_I)$$

$$\left.\frac{\partial \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x}\right|_{x=x_I} = \frac{\partial \boldsymbol{p}^{\mathrm{T}}(\mathbf{0})}{\partial x}\boldsymbol{C}(\boldsymbol{x}_I) + \boldsymbol{p}^{\mathrm{T}}(\mathbf{0})\frac{\partial \boldsymbol{C}(\boldsymbol{x}_I)}{\partial x}$$

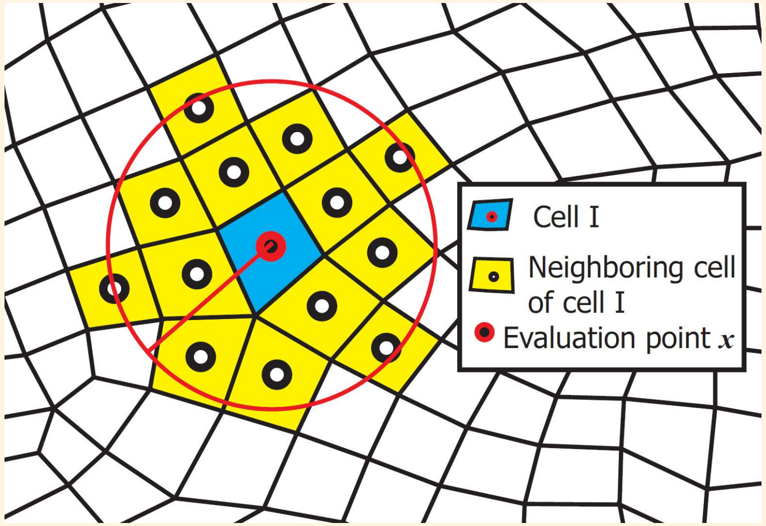
$$\boldsymbol{C}(\boldsymbol{x}_I) = \boldsymbol{M}^{-1}(\boldsymbol{x}_I)\boldsymbol{P}_{\Omega_{\boldsymbol{x}_I}}\boldsymbol{W}(\boldsymbol{x}_I)$$

$$\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x} = \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial \boldsymbol{W}(\boldsymbol{x})}{\partial x}\left(\boldsymbol{I} - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})\right)$$

The FV-MLS method

Hyperbolic-like terms:

- MLS is used to compute the gradients and high-order derivatives required for the reconstruction of the variable at integration points placed at interface.



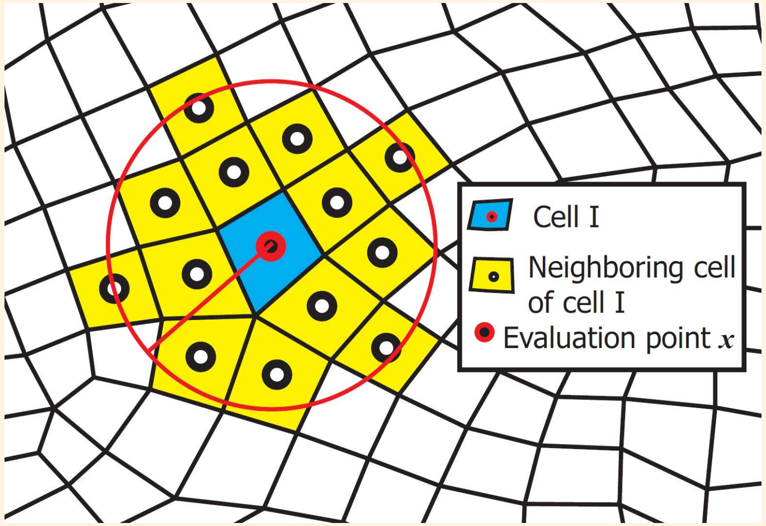
$$\boldsymbol{C}(\boldsymbol{x}_I) = \boldsymbol{M}^{-1}(\boldsymbol{x}_I)\boldsymbol{P}_{\Omega_{\boldsymbol{x}_I}}\boldsymbol{W}(\boldsymbol{x}_I)$$

$$\frac{\partial^2 \boldsymbol{U}_I}{\partial x^2} = \sum_{j=1}^{n_{\mathrm{x}}} \boldsymbol{U}_j \frac{\partial^2 N_j(\boldsymbol{x}_I)}{\partial x^2}$$

$$\left.\frac{\partial^2 \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^2}\right|_{x=x_I} \approx \frac{\partial^2 \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{0})}{\partial x^2}\boldsymbol{C}(\boldsymbol{x}_I)$$

The FV-MLS method

- Hyperbolic-like terms:
  - MLS is used to compute the gradients and high-order derivatives required for the reconstruction of the variable at integration points placed at interface.

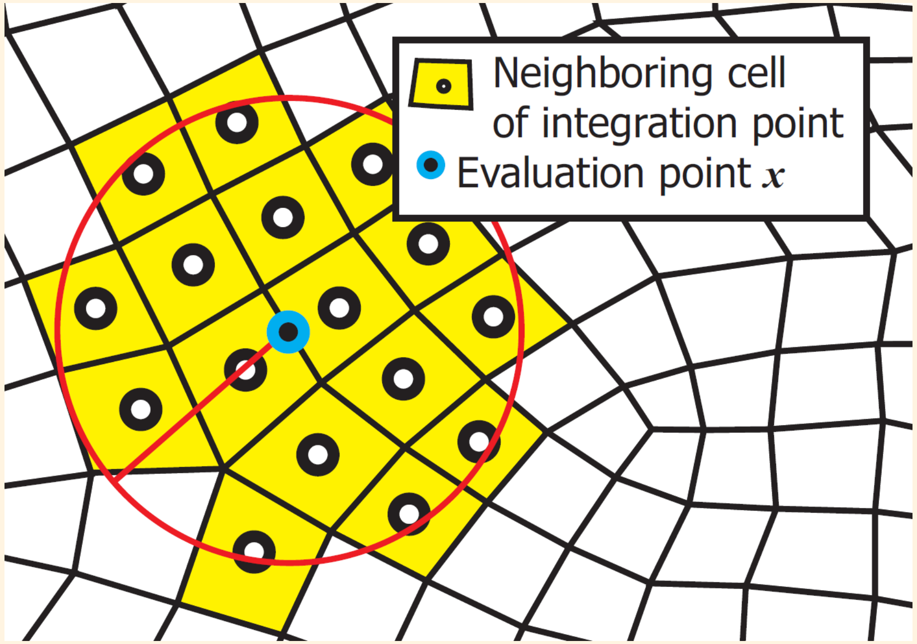


$$\frac{\partial^n \boldsymbol{U}_I}{\partial x^n} = \sum_{j=1}^{n_{\mathrm{x}}} \boldsymbol{U}_j \frac{\partial^n N_j(\boldsymbol{x}_I)}{\partial x^n}$$

$$\left.\frac{\partial^n \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x^n}\right|_{x=x_I} \approx \frac{\partial^n \boldsymbol{p}^{\mathrm{T}}(\mathbf{0})}{\partial x^n} \boldsymbol{C}(\boldsymbol{x}_I)$$

$$\boldsymbol{C}(\boldsymbol{x}_I) = \boldsymbol{M}^{-1}(\boldsymbol{x}_I) \boldsymbol{P}_{\Omega_{\boldsymbol{x}_I}} \boldsymbol{W}(\boldsymbol{x}_I)$$

- Elliptic-like terms:
  - Direct interpolation at Gauss points with MLS.

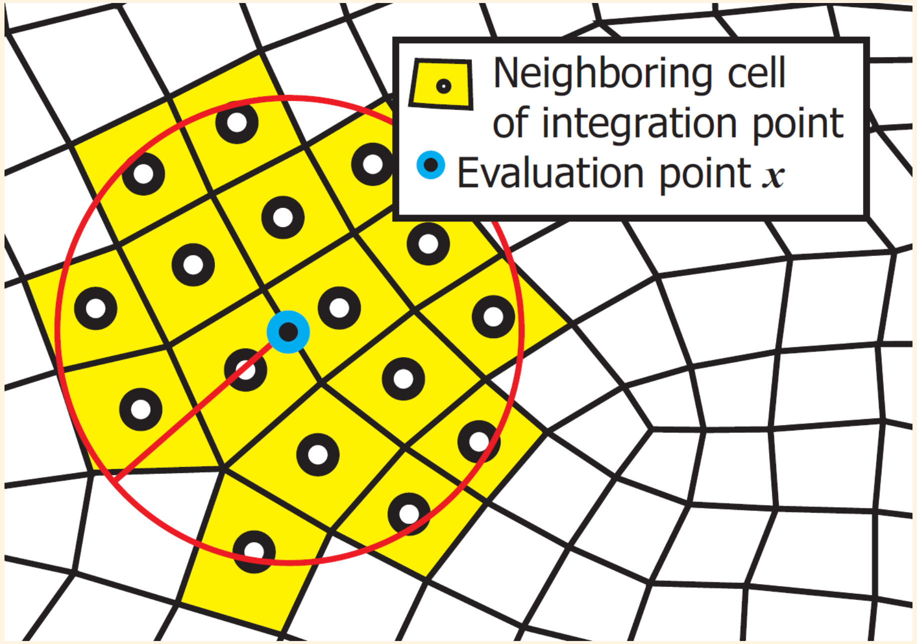


$$\boldsymbol{U}_{iq} = \sum_{j=1}^{n_{iq}} \boldsymbol{U}_j \boldsymbol{N}_j(\boldsymbol{x}_{iq})$$

$$\boldsymbol{N}^T(\boldsymbol{x}_{iq}) = \boldsymbol{p}^T(\boldsymbol{0})\boldsymbol{C}(\boldsymbol{x}_{iq})$$

The FV-MLS method

- Elliptic-like terms:
  - Direct interpolation at Gauss points with MLS.

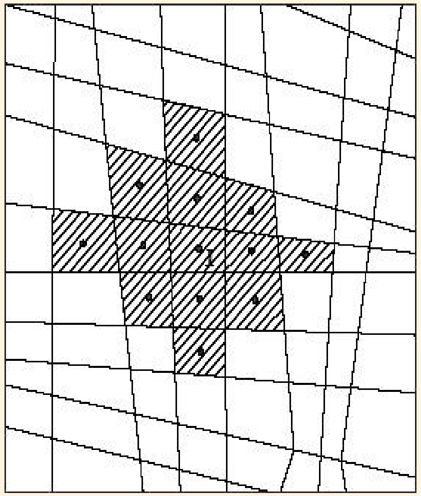


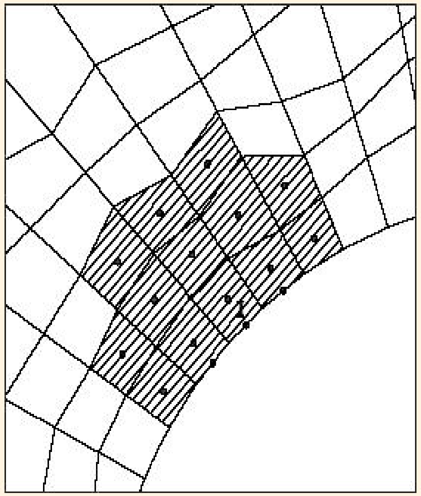
$$\nabla U_{iq} = \sum_{j=1}^{n_{iq}} U_j \nabla N_j(\boldsymbol{x}_{iq})$$

$$\left.\frac{\partial \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})}{\partial x}\right|_{\boldsymbol{x}=\boldsymbol{x}_{iq}} = \frac{\partial \boldsymbol{p}^{\mathrm{T}}(\mathbf{0})}{\partial x}\boldsymbol{C}(\boldsymbol{x}_{iq}) + \boldsymbol{p}^{\mathrm{T}}(\mathbf{0})\frac{\partial \boldsymbol{C}(\boldsymbol{x}_{iq})}{\partial x}$$

The FV-MLS method (V)

- Vertices and/or centroids of the control cells are the "particles" to perform the MLS approximation.
- We need to define stencils to "mark" the neighbor particles that define the cloud of points.





The FV-MLS method (VI)

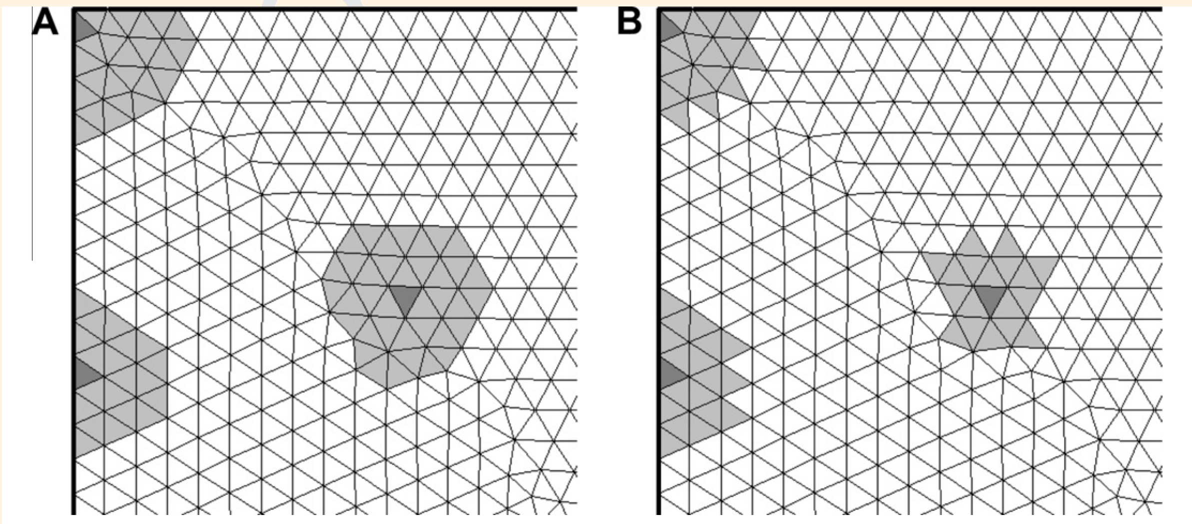
- How to define stencils?
- There exists an optimal size n_{xI} of points in the stencil such as $N_{min} < n_{xI}$

$$N_{min} = \frac{(d + order)!}{d!order!}$$

- If it is large $\Rightarrow$ excessive dissipation
- Maybe optimization??

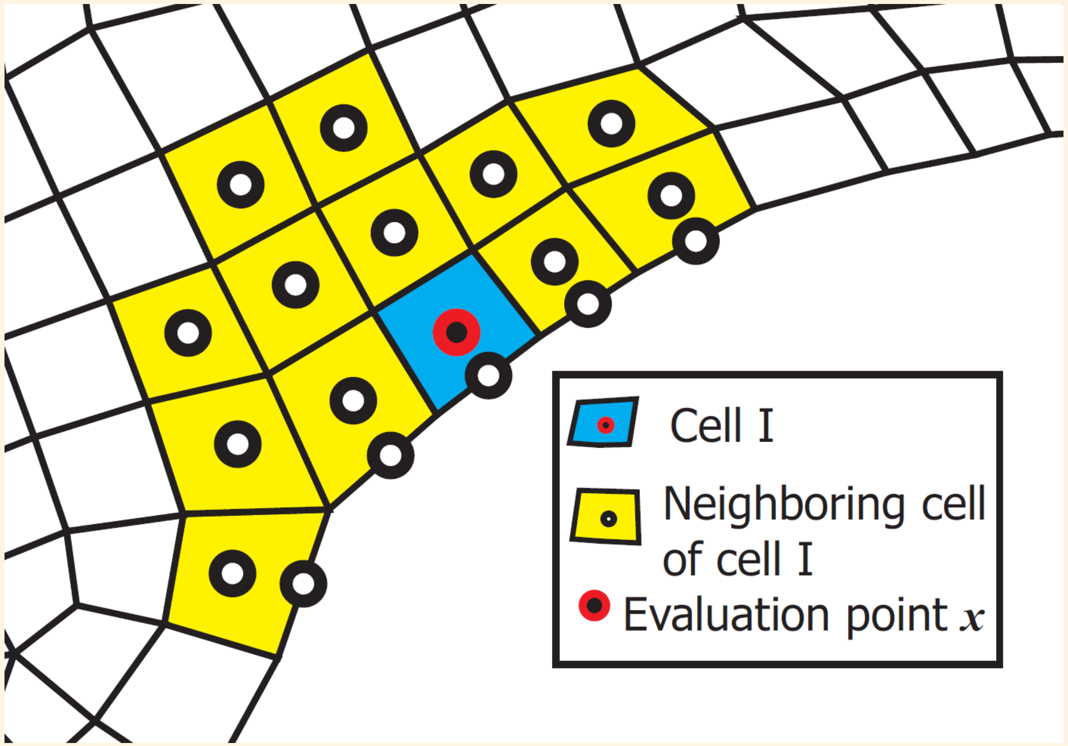
The FV-MLS method (VII)

- We want stencils as compact as possible by using layers of cells around the active cell
- In practice this requires a high number of points in the stencil
- To overcome this inconvenient, last particles are placed such as satisfying a barycentric equilibrium



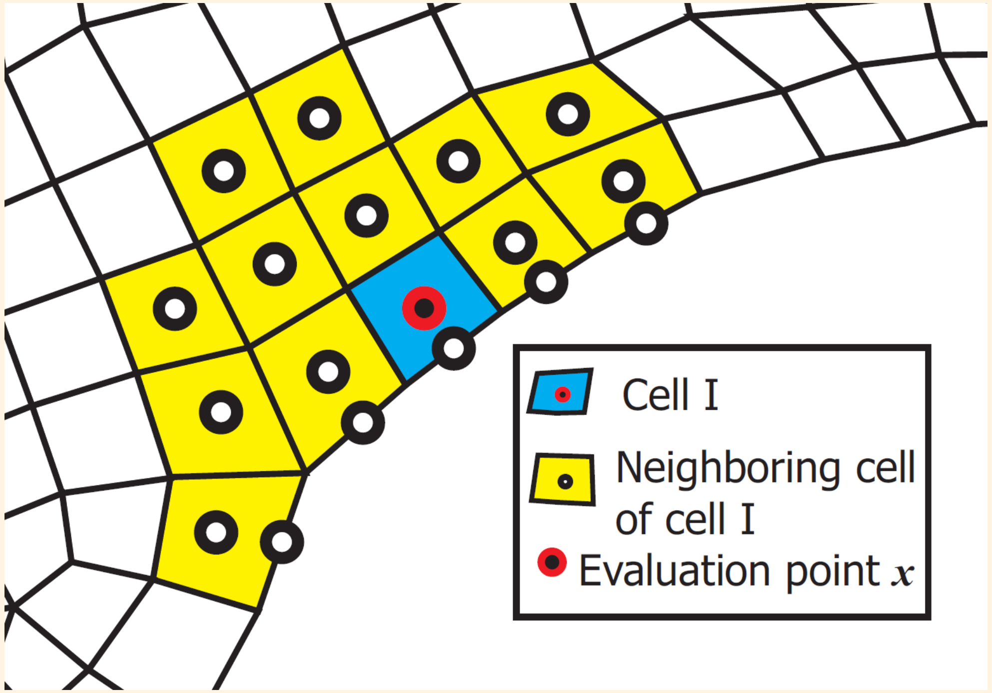
The FV-MLS method (VII)

- Boundary conditions: We impose them on the numerical fluxes



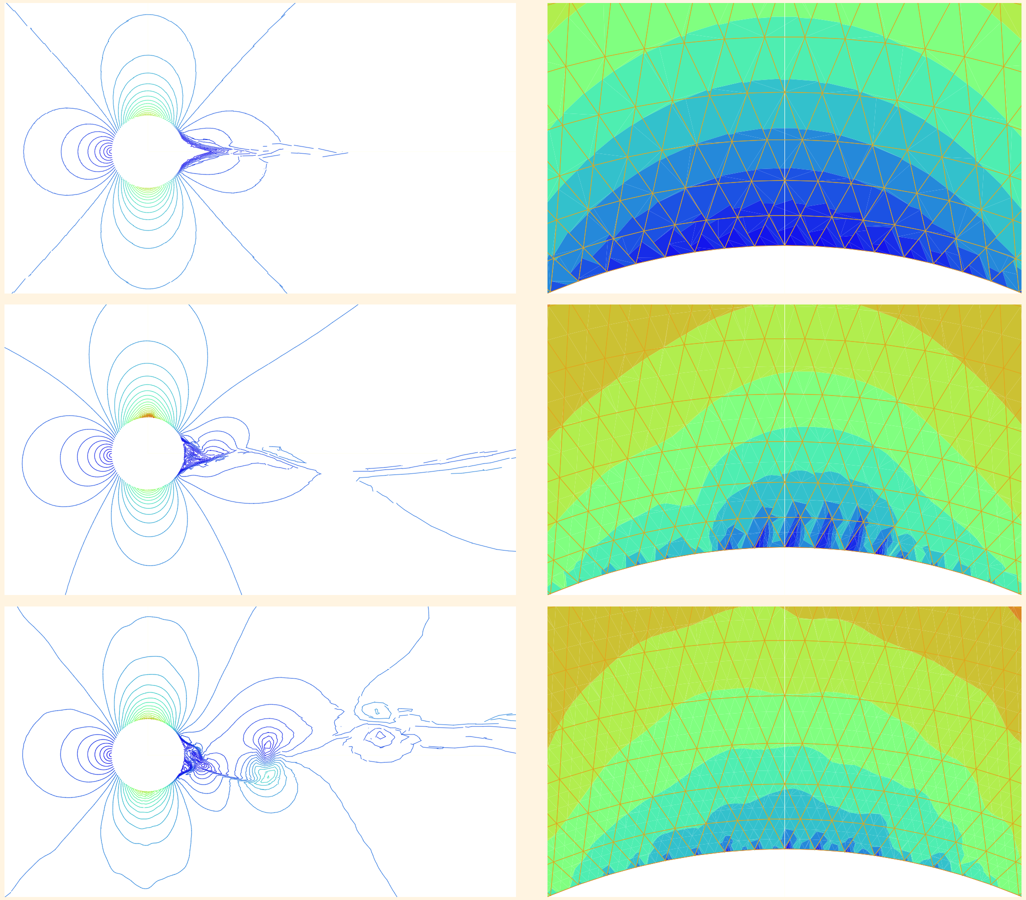
The FV-MLS method (VII)

- However, in order to improve the reconstruction we include ghost cells in the stencil



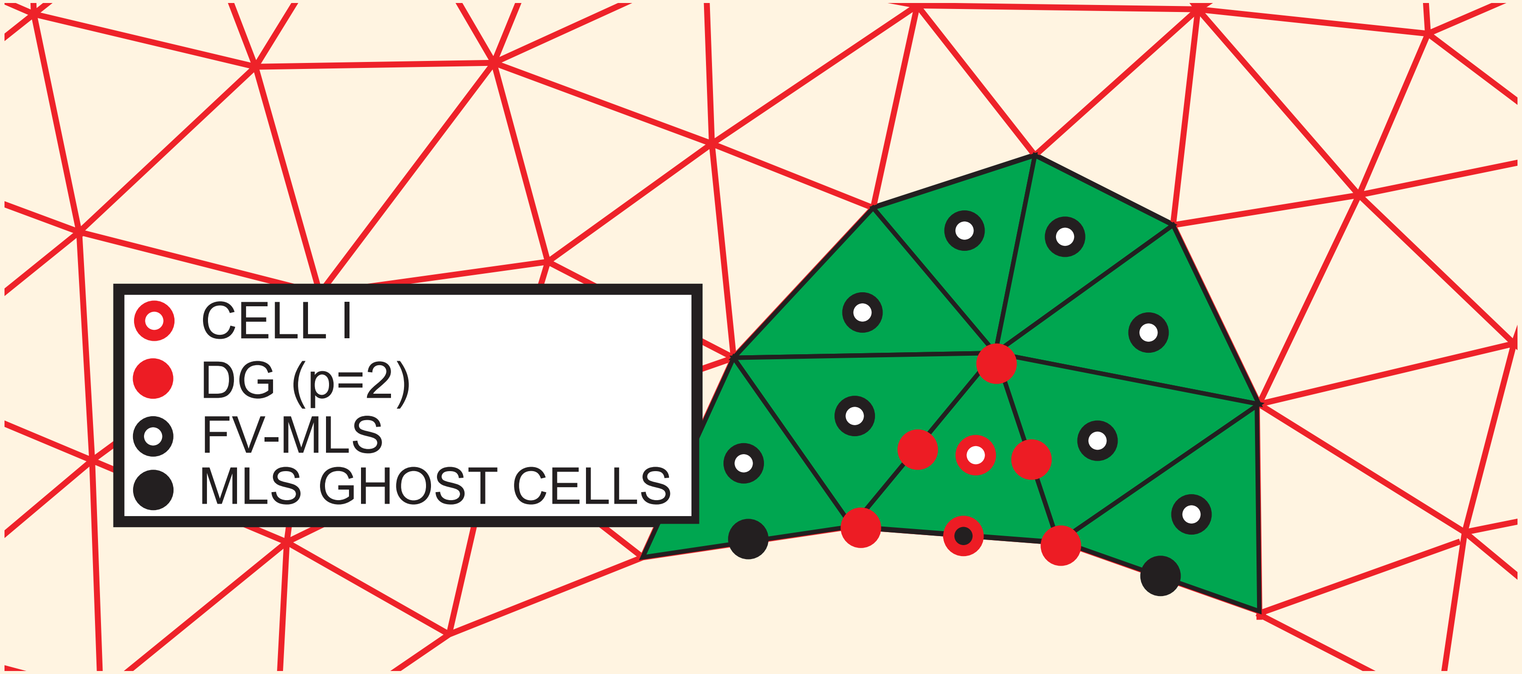
$$\nabla \boldsymbol{U}_I = \sum_{j=1}^{n_x} \boldsymbol{U}_j \nabla \boldsymbol{N}_j(\boldsymbol{x}_I)$$

A note on curved boundaries



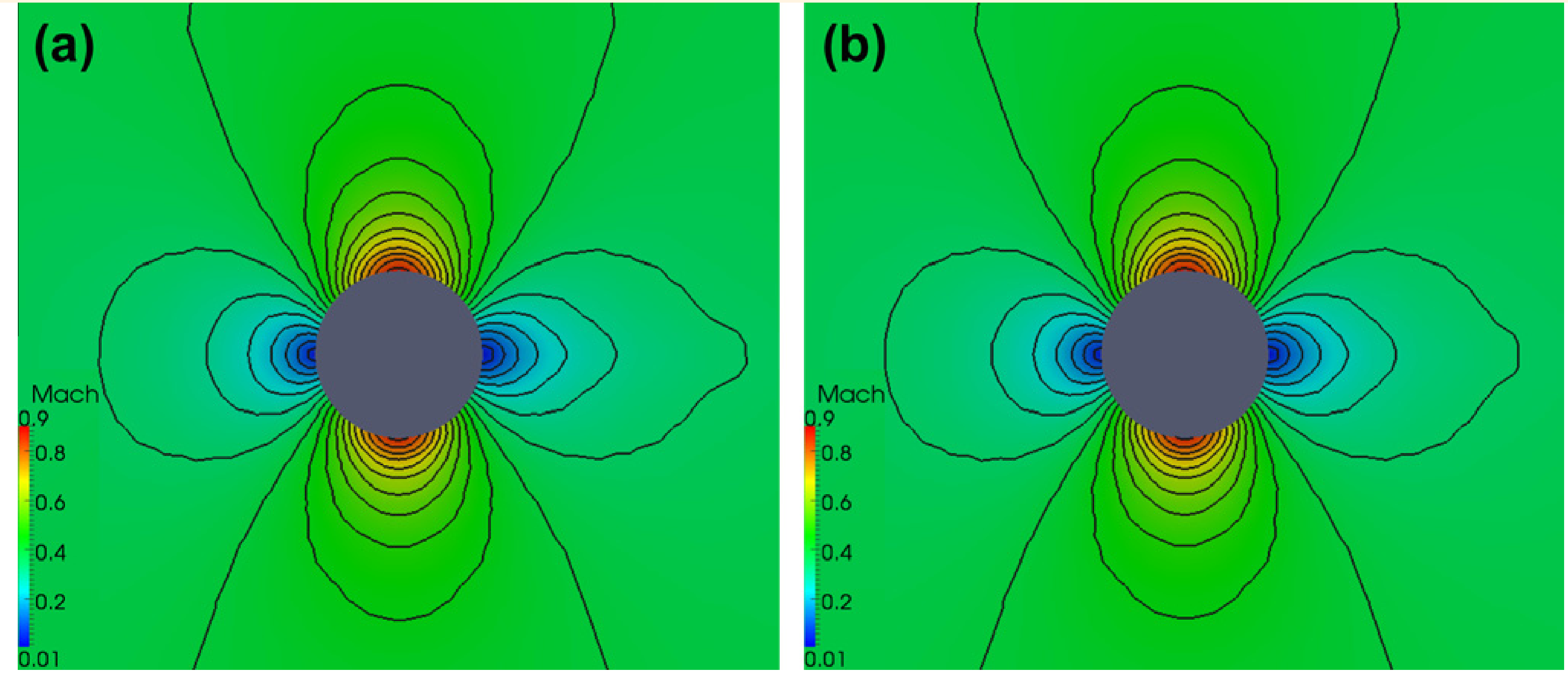
Mach isolines (left) and density at the top of the cylinder (right) with reflecting boundary conditions. $p = 1$, 2, 3, from top to bottom. Taken from Krivodonova and Berger, High-Order Accurate Implementation of Solid Wall Boundary Conditions in Curved Geometries, JCP, 2006

A note on curved boundaries



Schematic representation of the differences on curved boundary discretization between FV-MLS and DG. Shaded cells represent the MLS stencil.

A note on curved boundaries



Third-order accurate FV-MLS computation on a 64×16 grid using boundary normal evaluations based on a straight representation (a) or on a physical representation (b) for curved geometry.

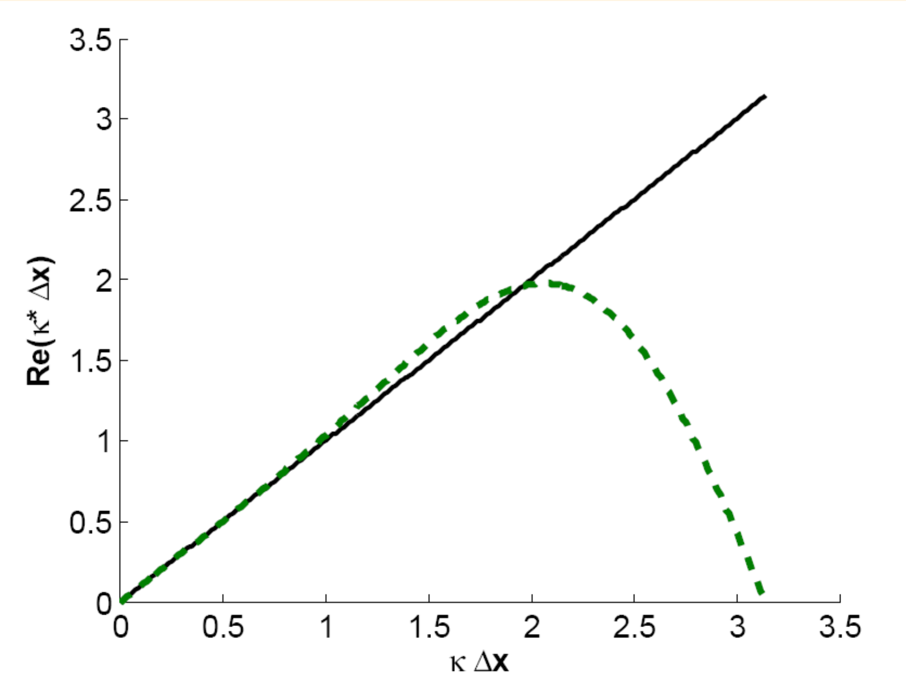
Accuracy assesment of a high-order moving least squares finite volume method for compressible flows, C&F, 2013

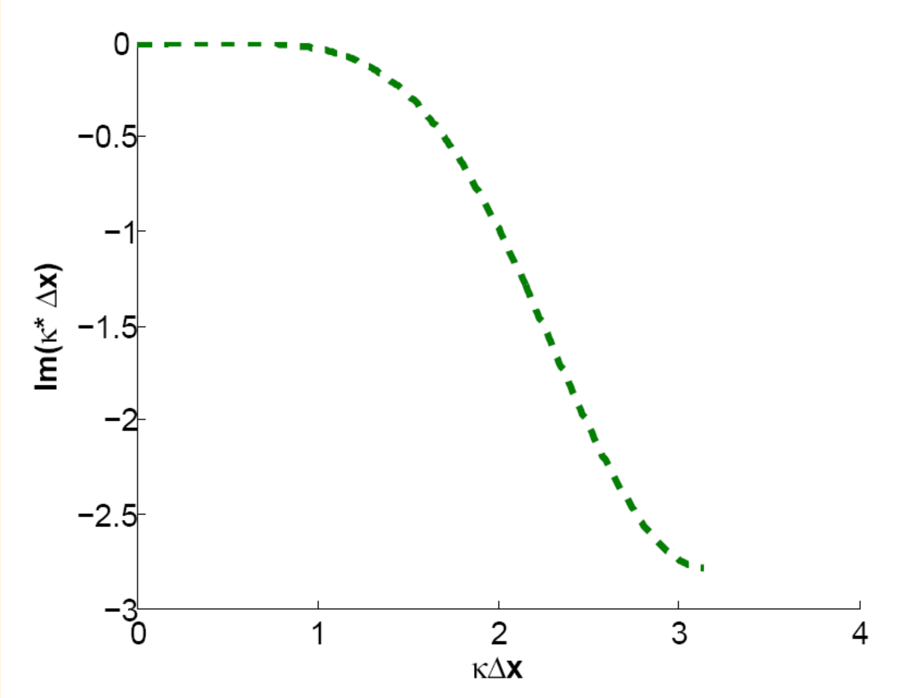
Properties of the FV-MLS method

- We perform a Fourier Analysis for the 1D linear advection equation.
- We obtain the dispersion-dissipation properties.
- We compare MLS interpolation with Piecewise Polynomial Interpolation.
- We check the order of convergence.

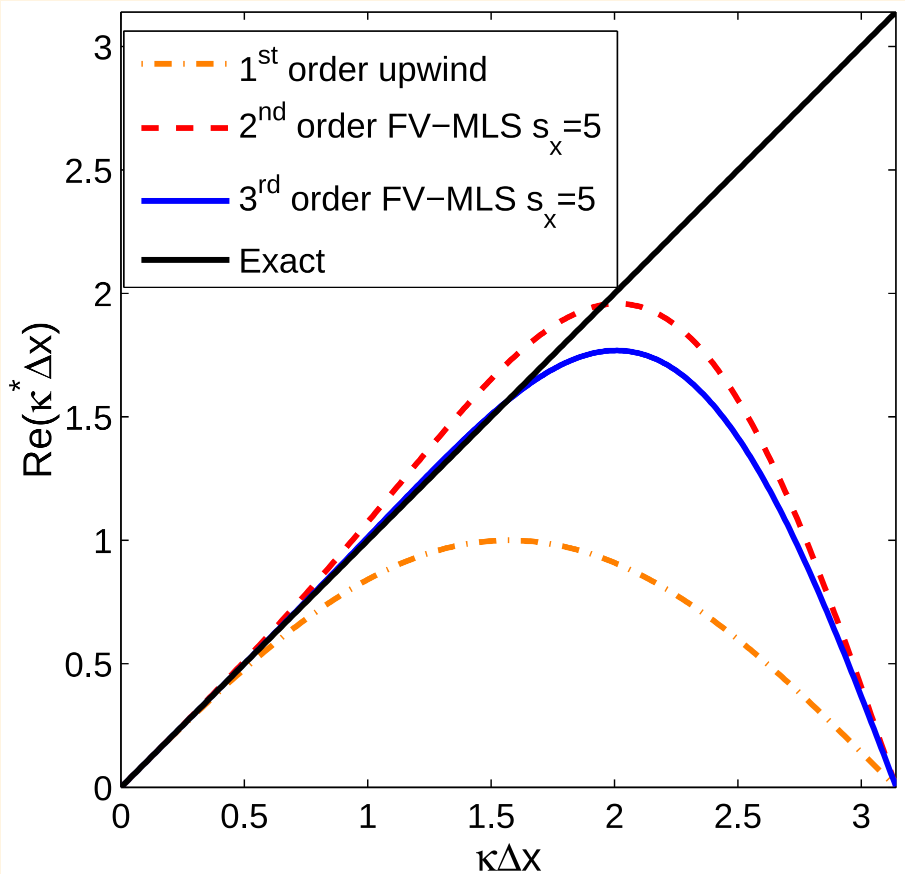
Dispersion and dissipation (I)

- Dispersion error: Associated with the error in the speed of the wave propagation
- Dissipation error: Associated with the error in the wave amplitude

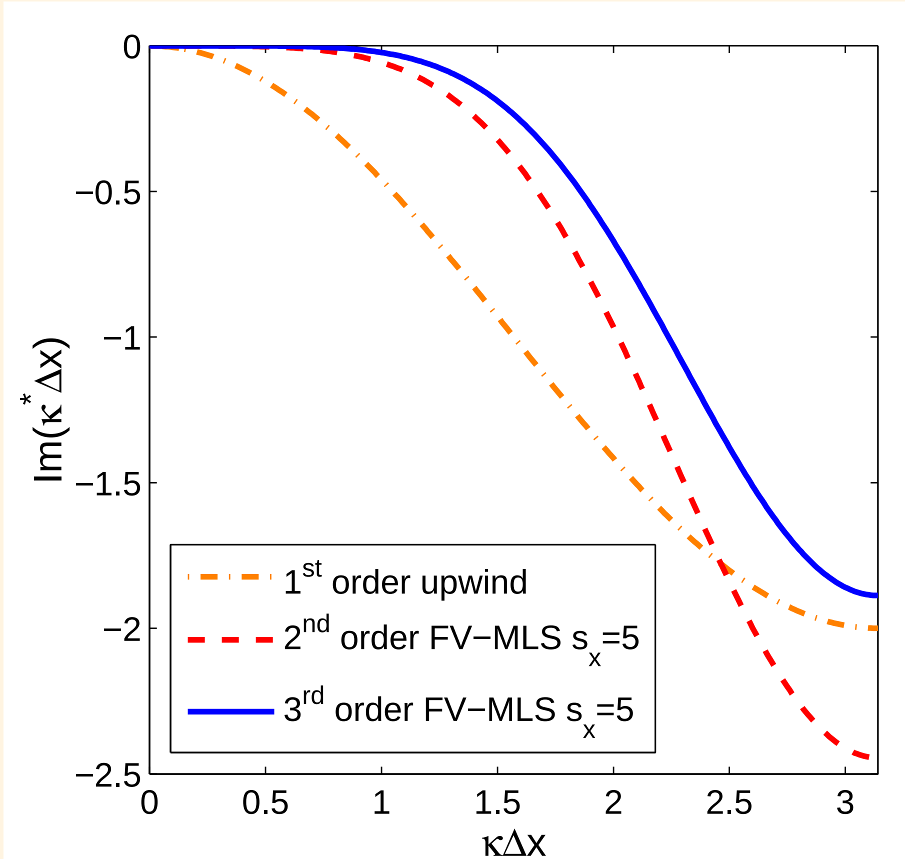




Dispersion and dissipation

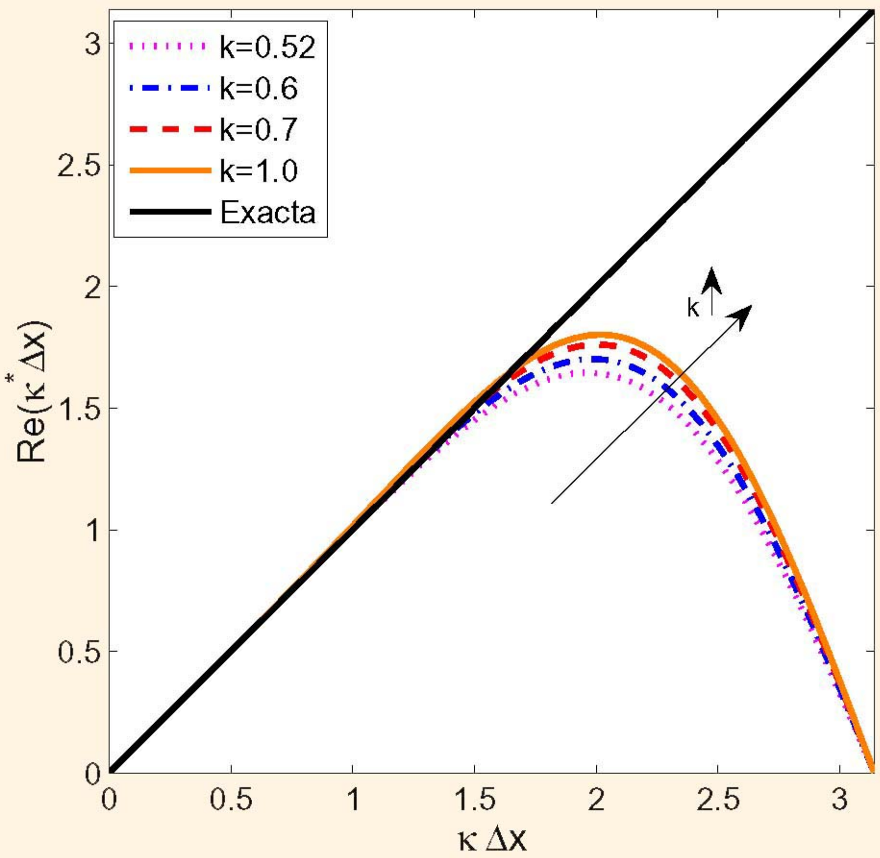


DISPERSION

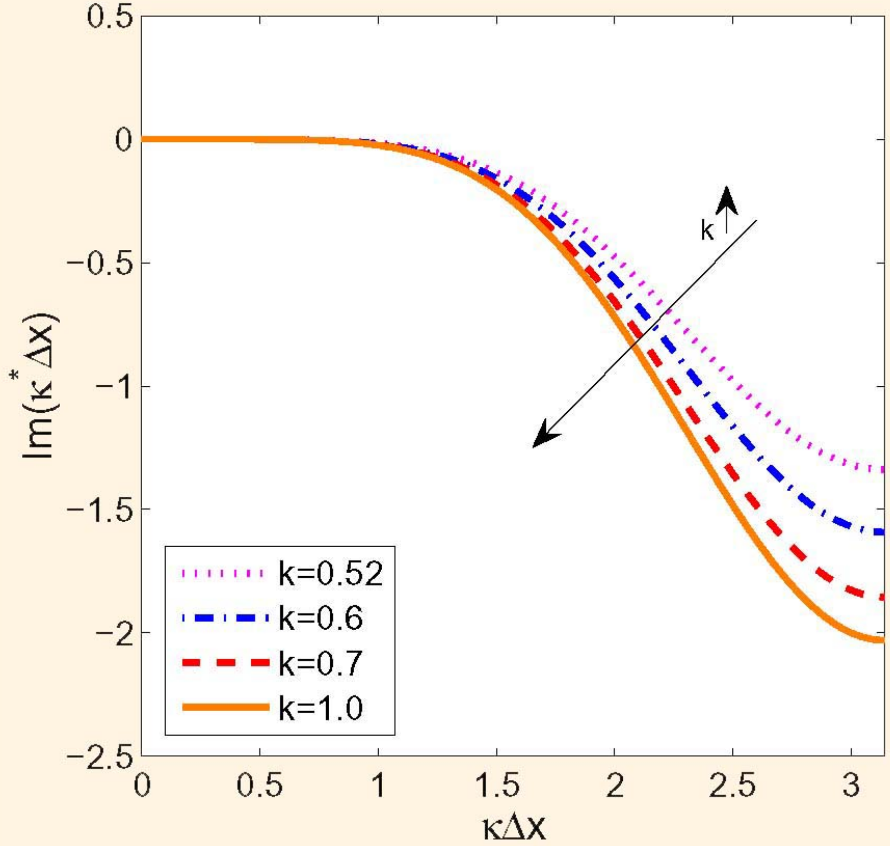


DISSIPATION

Dispersion-Dissipation Properties. Cubic spline kernel

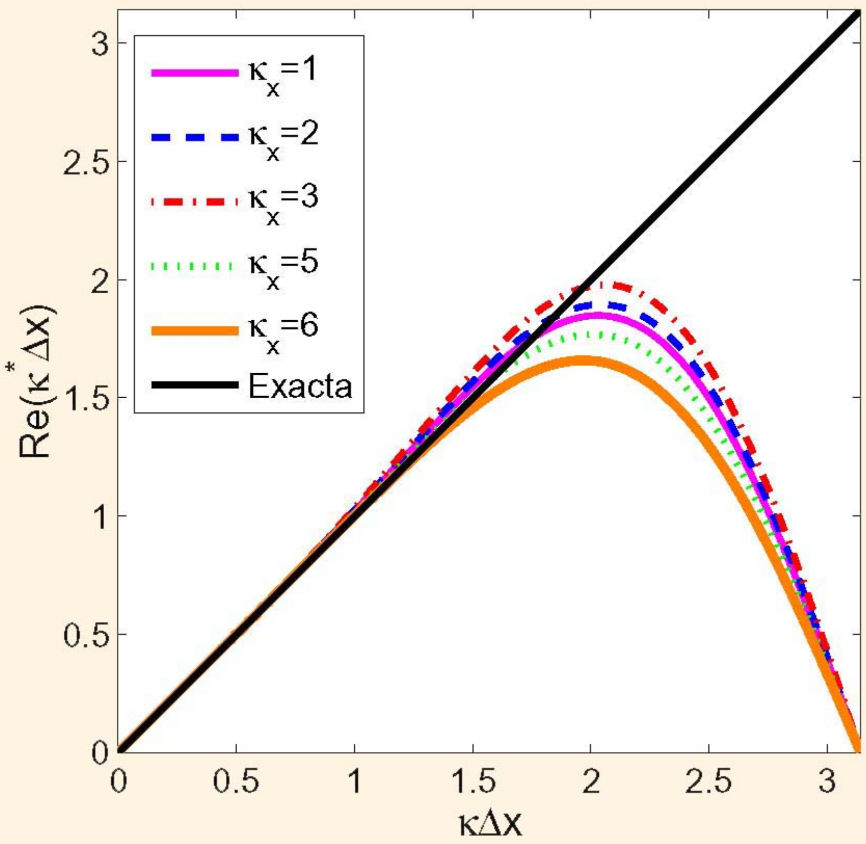


DISPERSION

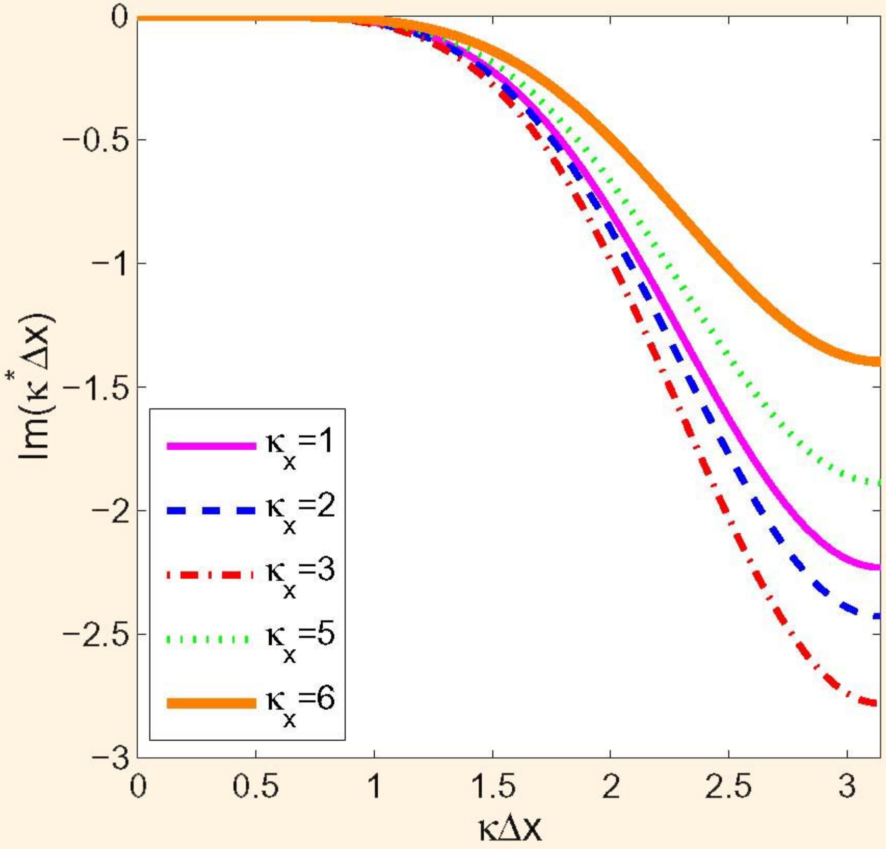


DISSIPATION

Dispersion-Dissipation Properties. Exponential kernel

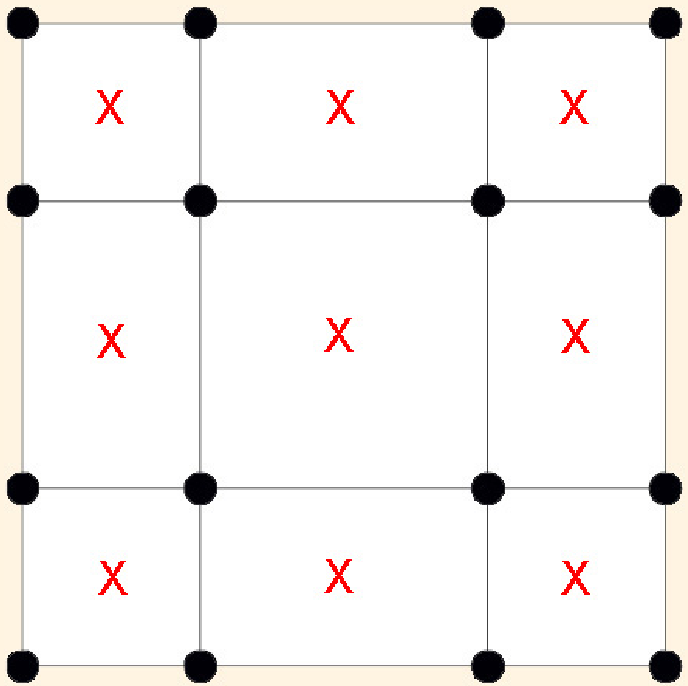


DISPERSION



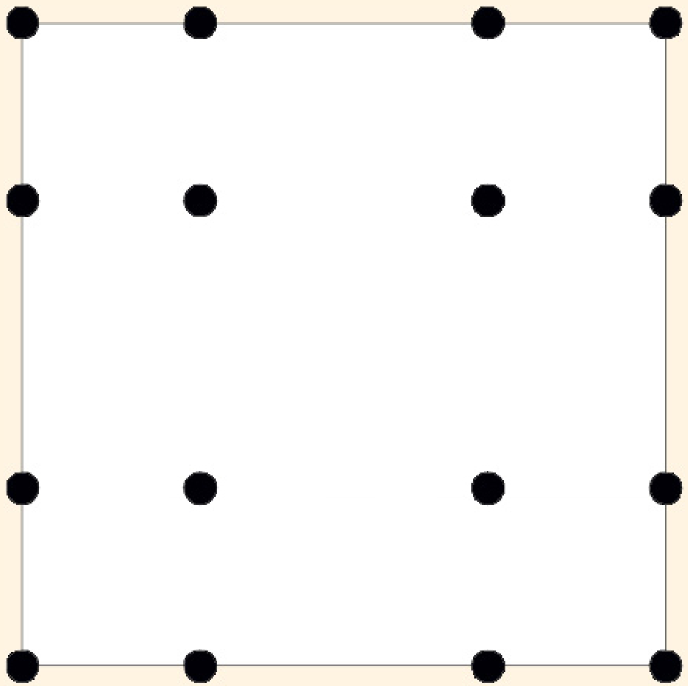
DISSIPATION

A COMPARISON BETWEEN Piecewise Polynomial Interpolation (PPI) AND FV-MLS



IS MLS INTERPOLATION ACCURATE?

- We compare interpolation with equivalent spatial resolution by using Moving Least Squares (MLS) (cubic basis) and Piecewise Polynomial Interpolation (PPI) $(p = 3)$.

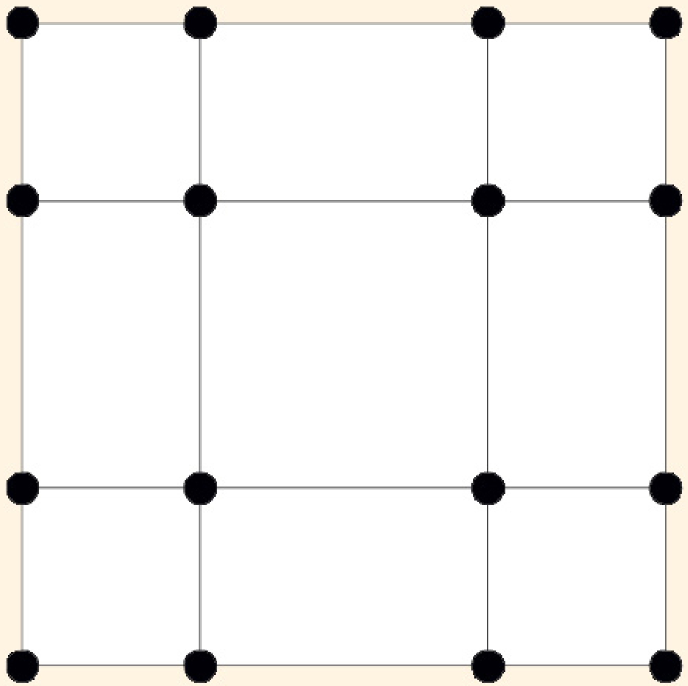


Division of a $p = 3$ element to obtain a FV-MLS grid with equivalent spatial resolution.

IS MLS INTERPOLATION ACCURATE?

SHARK-FV 2015 Conference
SHARING HIGHER-ORDER ADVANCED RESEARCH KNOW-HOW on FINITE VOLUME
Ofir, Portugal
May 18 - 22, 2015

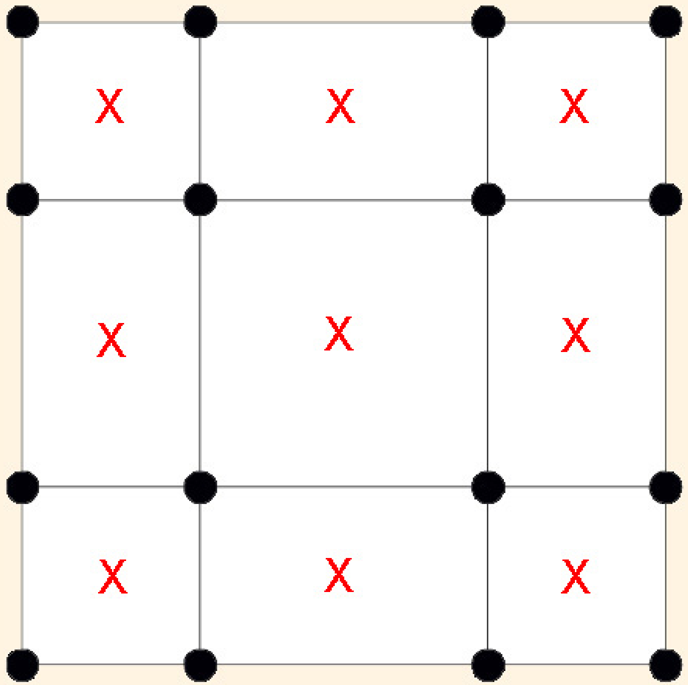
- We compare interpolation with equivalent spatial resolution by using Moving Least Squares (MLS) (cubic basis) and Piecewise Polynomial Interpolation (PPI) $(p = 3)$.



Division of a $p = 3$ element to obtain a FV-MLS grid with equivalent spatial resolution.

IS MLS INTERPOLATION ACCURATE?

- We compare interpolation with equivalent spatial resolution by using Moving Least Squares (MLS) (cubic basis) and Piecewise Polynomial Interpolation (PPI) $(p = 3)$.



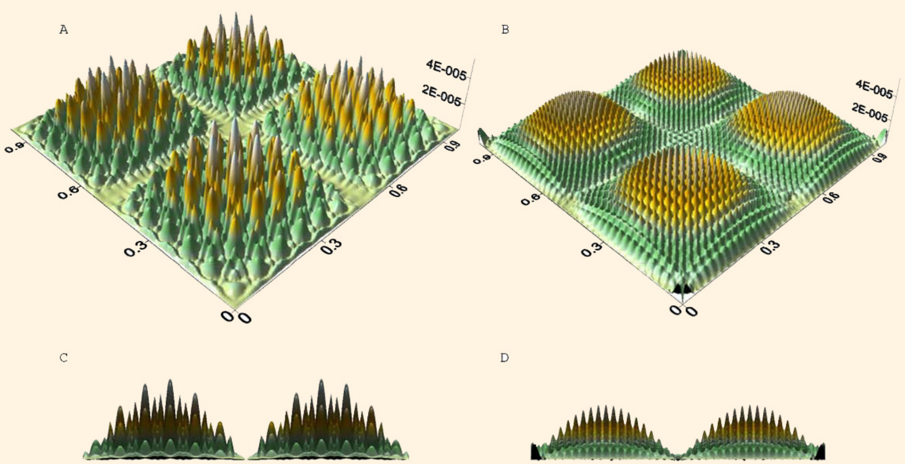
Division of a $p = 3$ element to obtain a FV-MLS grid with equivalent spatial resolution.

Interpolation on a structured grid

- Function: $u(x, y) = \sin(2\pi x) \sin(2\pi y)$
  - 13×13 $p = 3$ elements on a cartesian $[0, 1] \times [0, 1]$ grid.
  - 39×39 FV-MLS elements on a cartesian $[0, 1] \times [0, 1]$ grid.
  - We interpolate for both grids at the same points (located at the 4×4 Gauss-Legendre points of each FV-MLS element).

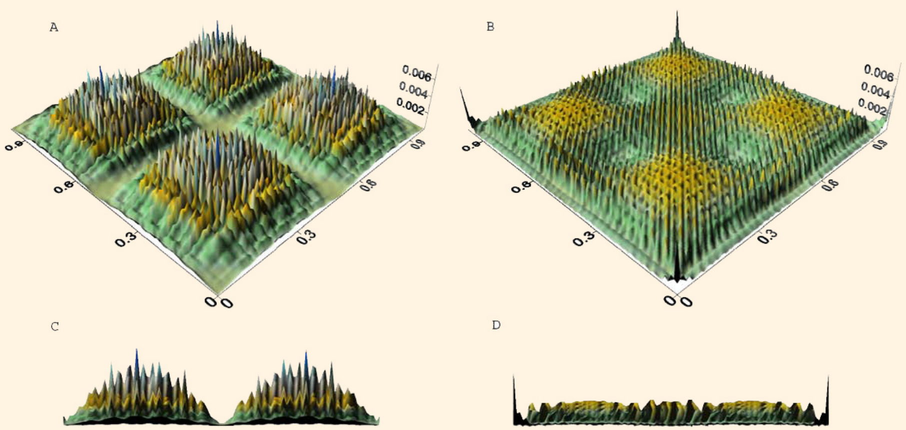
Absolute value of the error in the variable

PPI | MLS



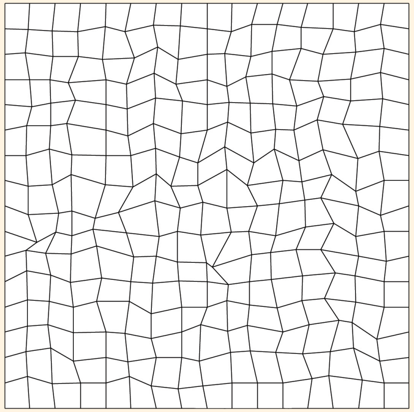
Absolute value of the error in the derivative

PPI | MLS



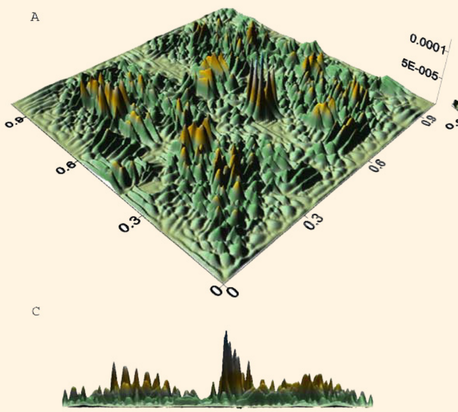
Interpolation on a distorted grid

- Function: $u(x, y) = \sin(2\pi x)\sin(2\pi y)$

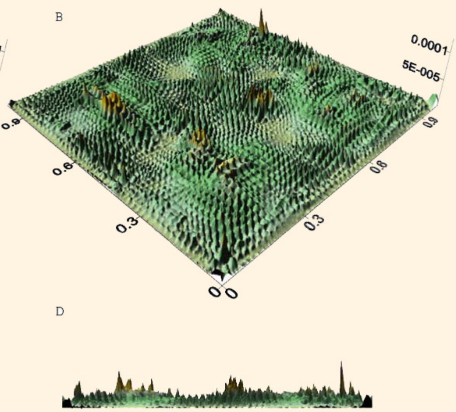


Absolute value of the error in the variable

PPI

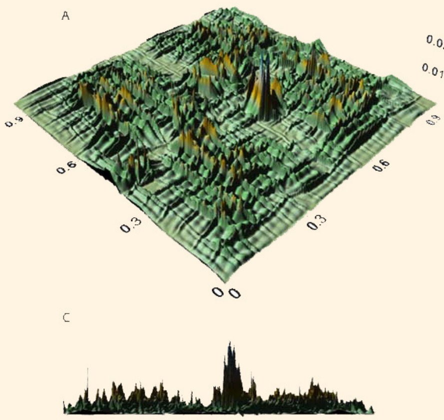


MLS

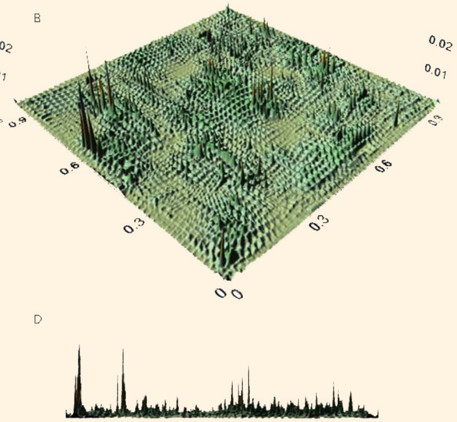


Absolute value of the error in the derivative

PPI

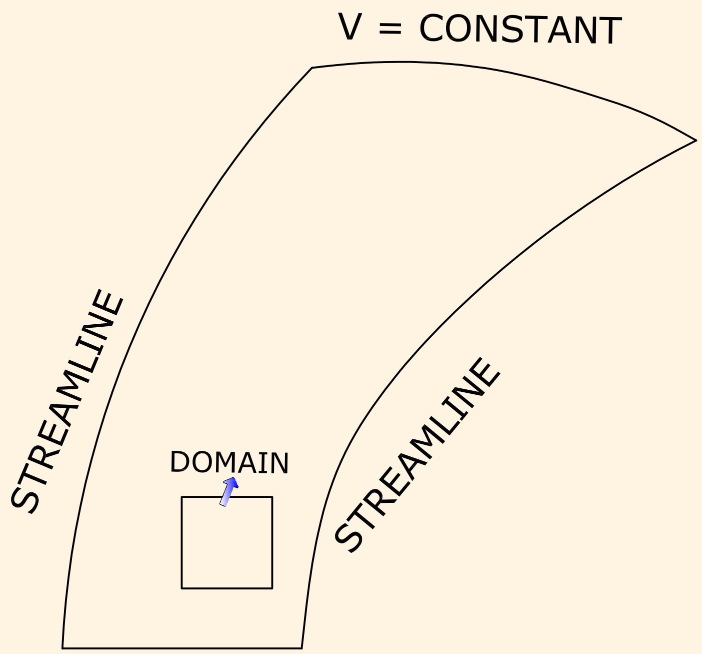


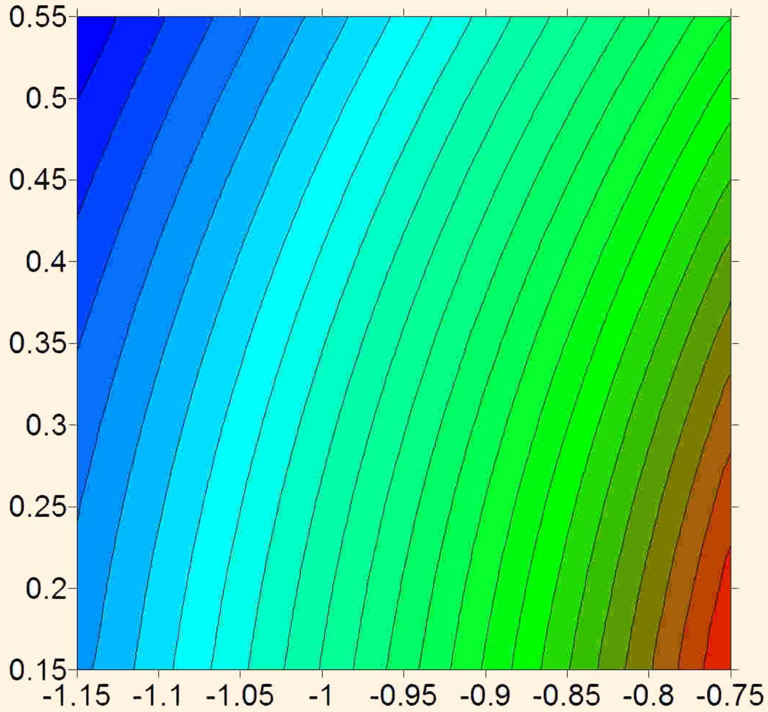
MLS



Order of Convergence. Ringleb Flow (I)

- Domain: $-1.15 \leq x \leq -0.75$, $0.15 \leq y \leq 0.55$

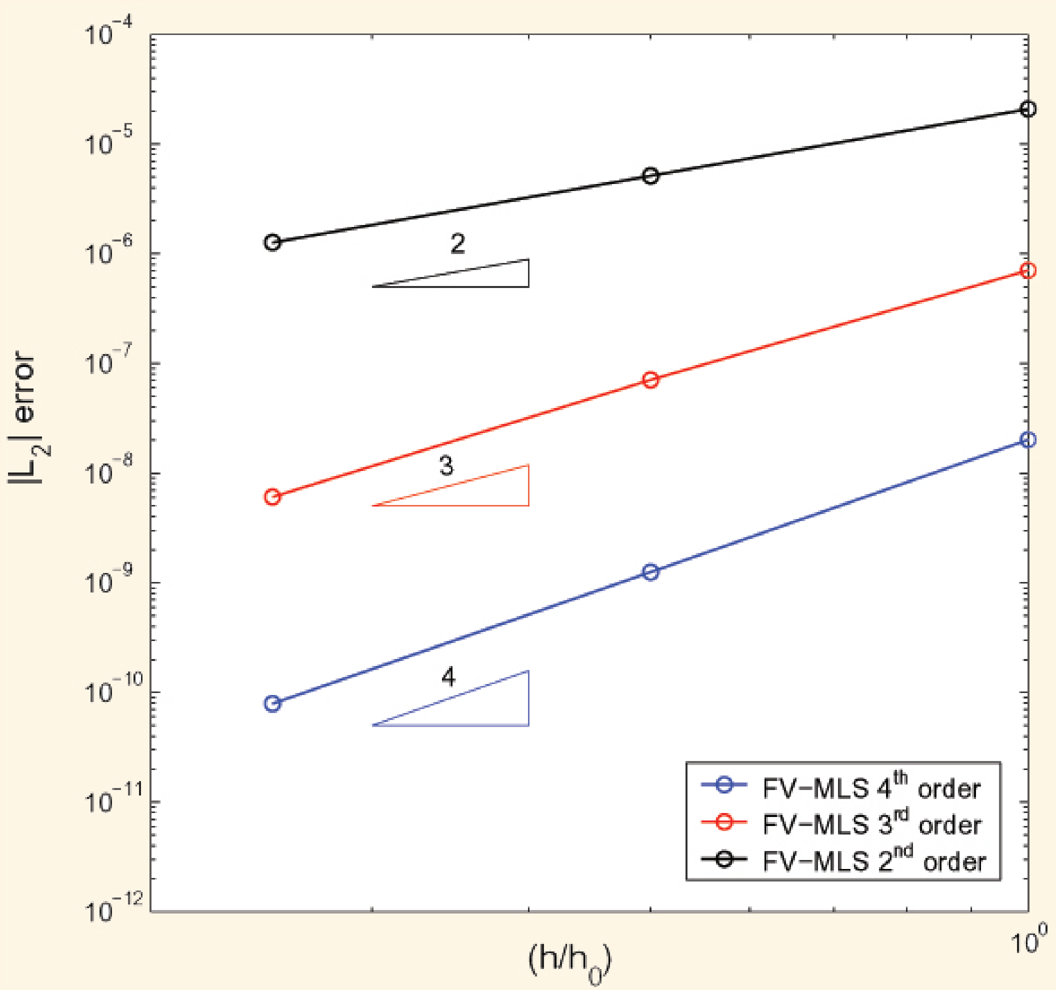




Mach isolines.

SHARK-FV 2015 Conference
SHARING HIGHER-ORDER ADVANCED RESEARCH KNOW-HOW on FINITE VOLUME
Ofir, Portugal
May 18 - 22, 2015

Order of Convergence. Ringleb Flow (II)



Grids

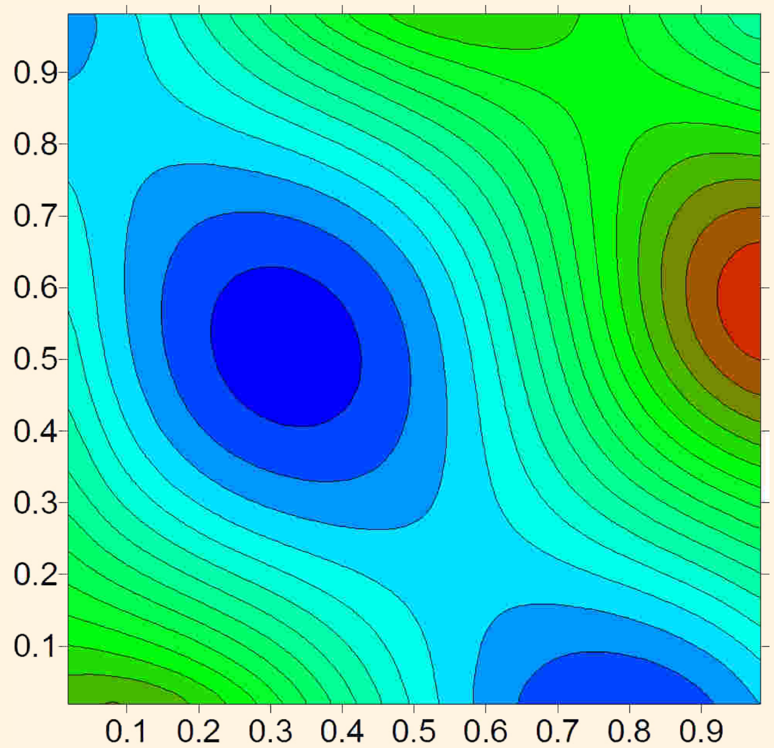
FV-MLS
15 x 15
30 x 30
60 x 60

- The order of convergence is the expected one.

Order of Convergence. Poisson (I)

$$\begin{array}{rcccc} -\Delta u & = & f & in & \Omega \\ u & = & g_D & on & \Gamma_D \end{array}$$

$$u(x,y) = \exp(\alpha \sin(Ax + By) + \beta \cos(Cx + Dy))$$



Isolines of the exact solution for u.

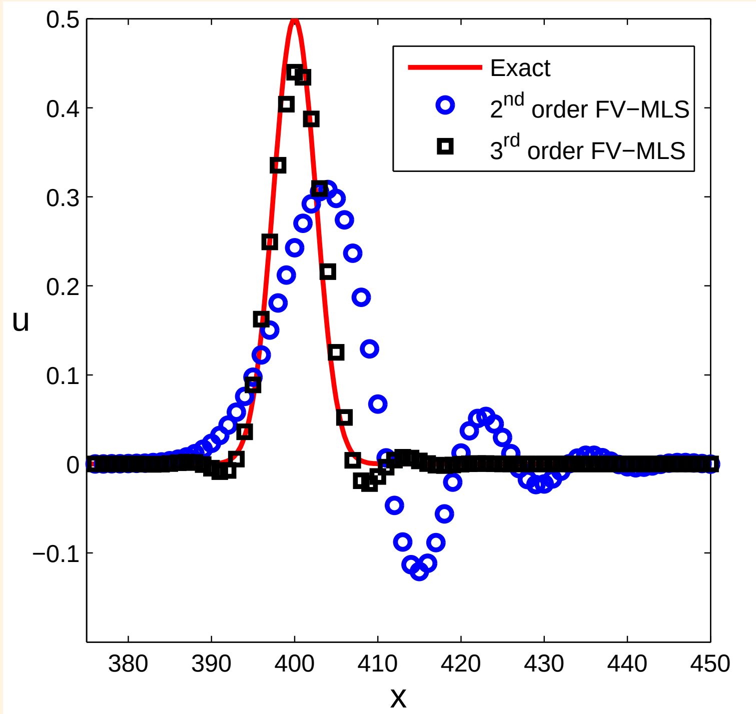
Order of Convergence. Poisson (II)

h/h_0	DG $p=3$ Error L_2 u	FV-MLS Error L_2 u
1	$2.50\ E-04$	$8.34\ E-05$
0.5	$1.20\ E-05$	$5.60\ E-06$
0.25	$6.05\ E-07$	$3.75\ E-07$
0.125	$3.16\ E-08$	$2.52\ E-08$

h/h_0	DG $p=3$ Order of Convergence u	FV-MLS Order of Convergence u	DG $p=3$ Order of Convergence s	FV-MLS Order of Convergence s
1	–	–	–	–
0.5	4.38	3.86	3.83	3.54
0.25	4.31	3.99	3.69	3.52
0.125	4.26	3.89	3.60	3.46

A first motivating example. 1D Linear advection equation

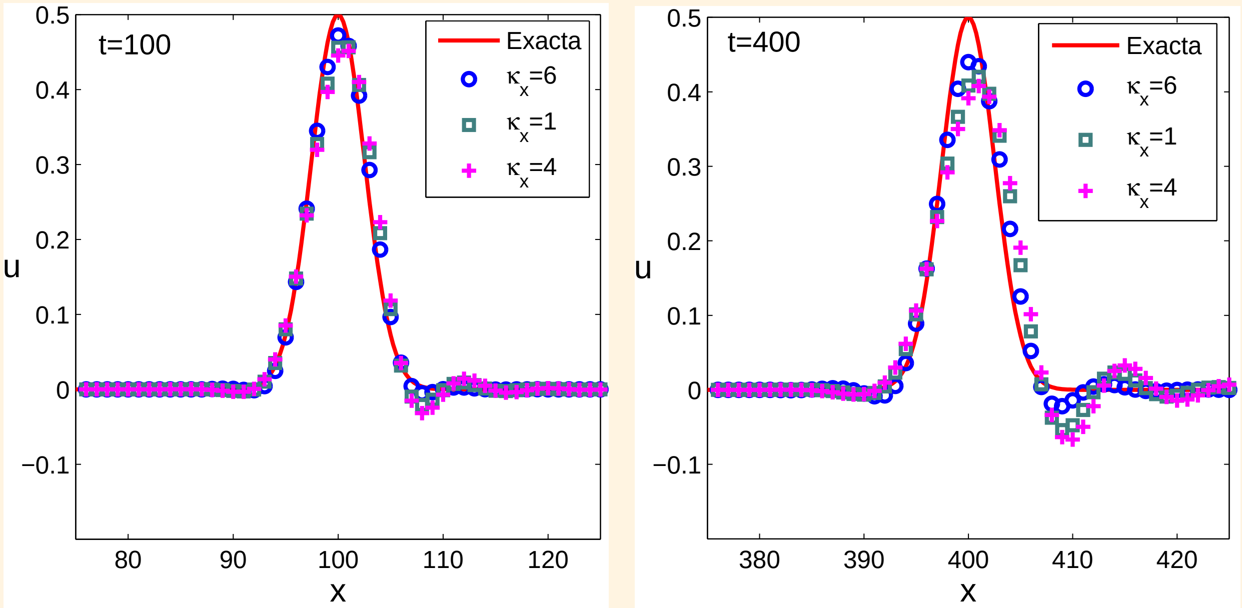
- First ICASE/LaRC Workshop on Benchmark problems in CAA
- We solve $\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$ with $u(x,0) = 0.5e^{-\ln(2)\left(\frac{x}{3}\right)^2}$



1D Linear advection equation, $a = 1$, $t = 400$, $\Delta x = 1$, $CFL = 0.6$

A first motivating example. 1D Linear advection equation

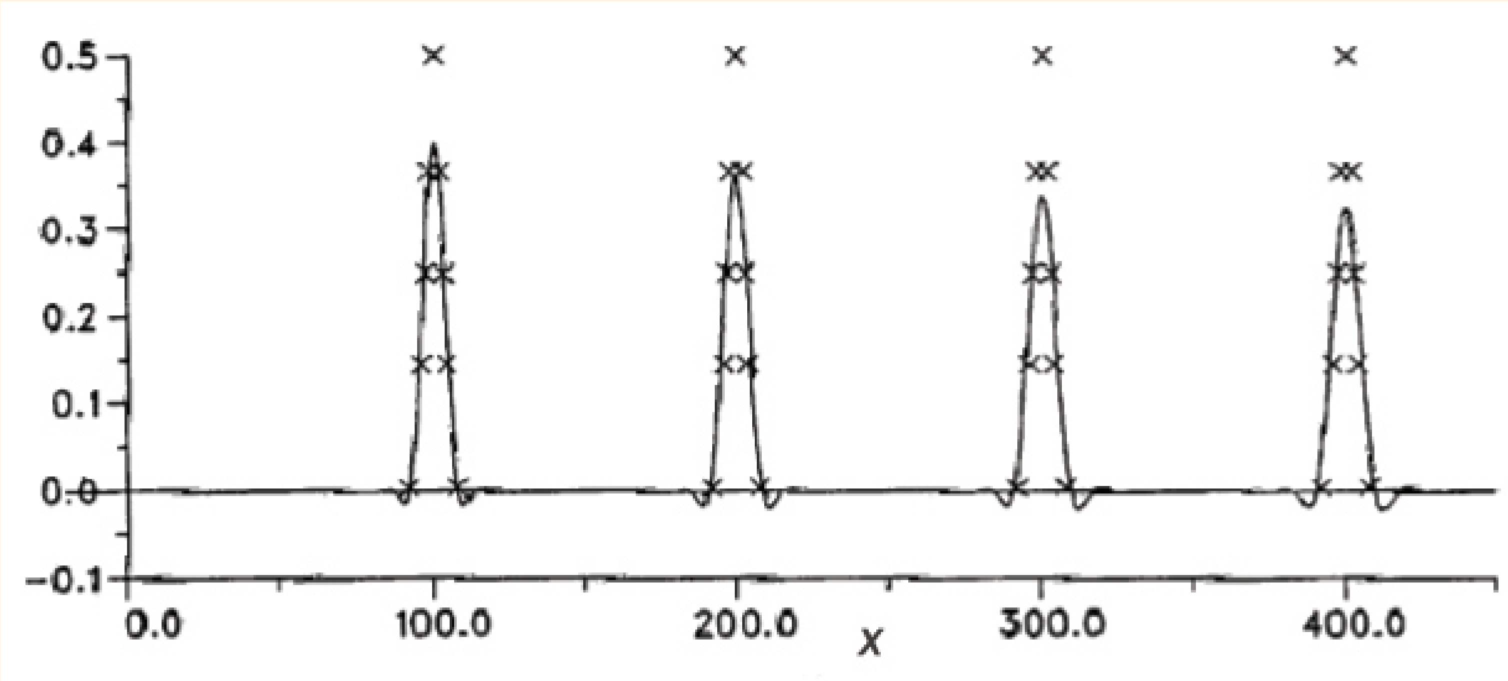
- First ICASE/LaRC Workshop on Benchmark problems in CAA
- We solve $\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$ with $u(x,0) = 0.5e^{-\ln(2)\left(\frac{x}{3}\right)^2}$



1D Linear advection equation, $a = 1$, $t = 400$, $\Delta x = 1$, $CFL = 0.6$

A first motivating example. 1D Linear advection equation

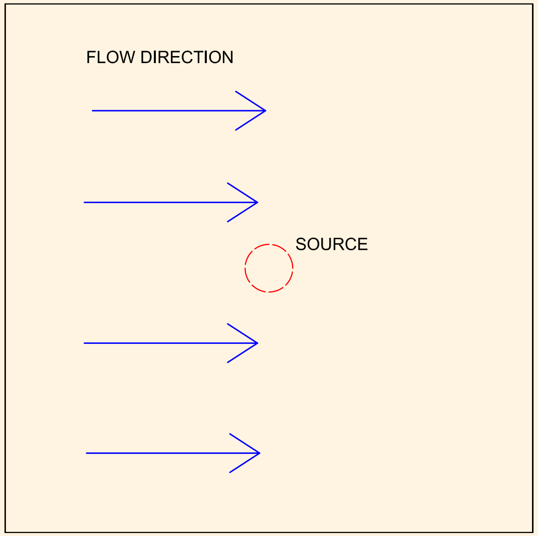
- Solution with a fourth order MacCormack scheme, $\Delta x = 1$, $CFL = 0.2$



Viswanathan, Sankar, A Comparative Study of Upwind and MacCormac schemes for CAA Benchmark problems, First ICASE/LaRC Workshop on Benchmark problems in CAA, NASA Conference Publication 3300, 185-195, 1995

A CAA example on an unstructured grid.

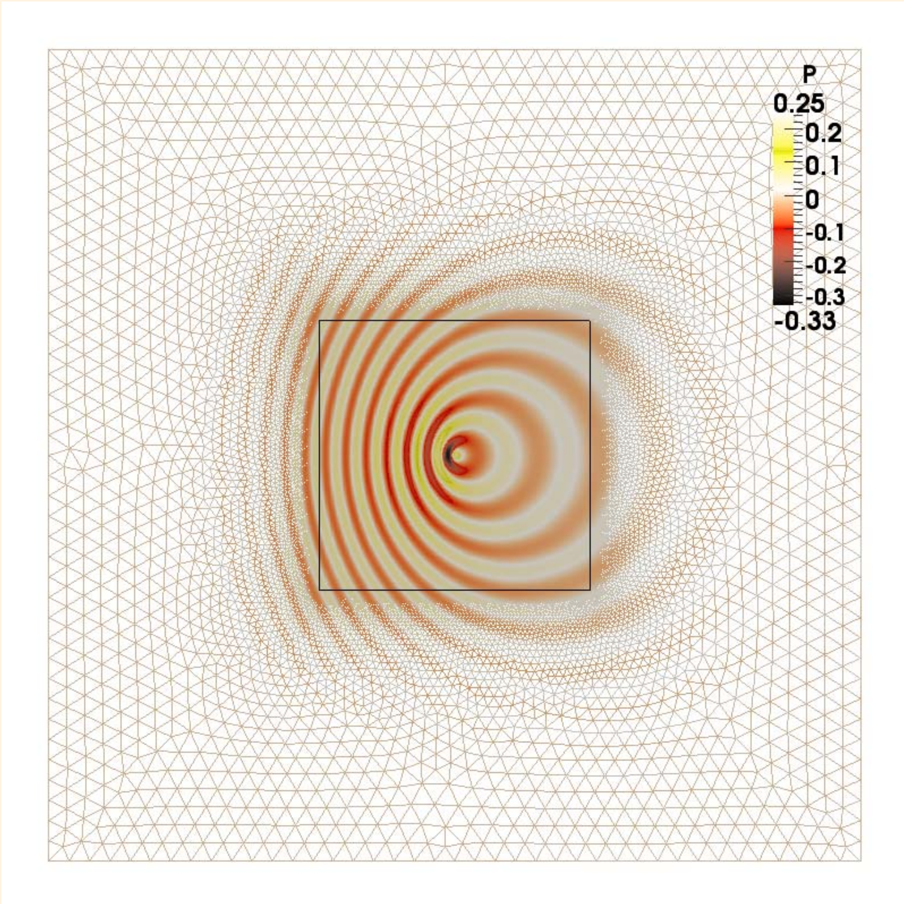
- Solve the LEE for the convection of a monopolar source

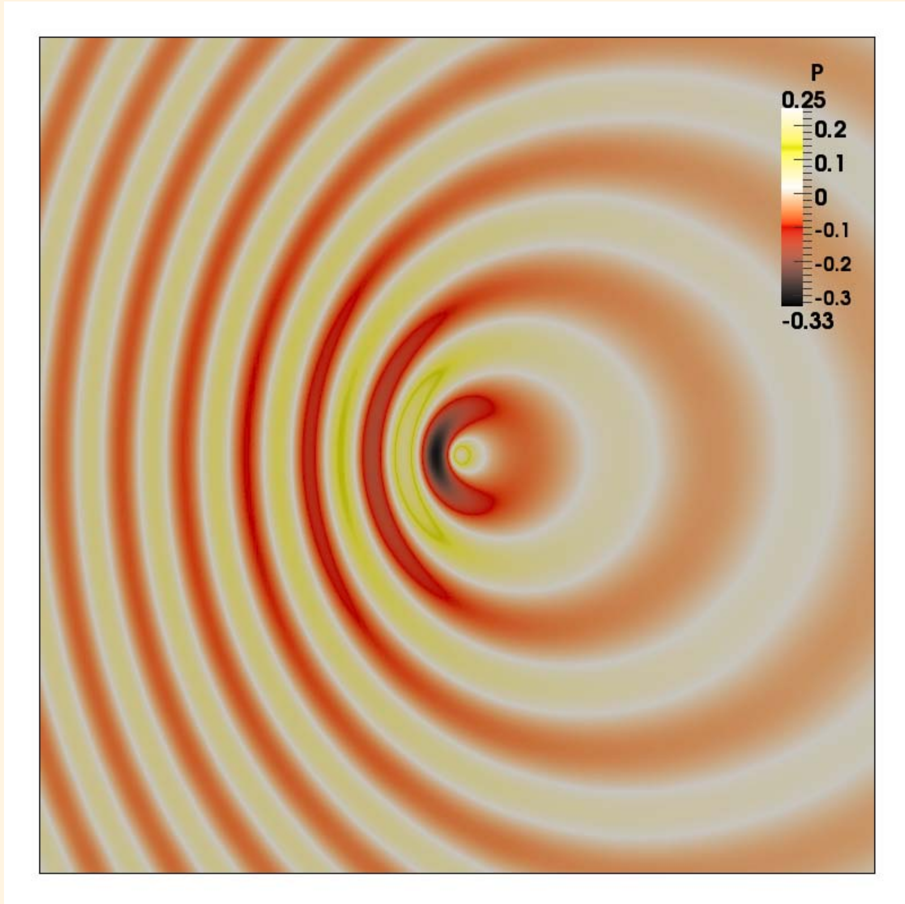


Sketch of the problem.

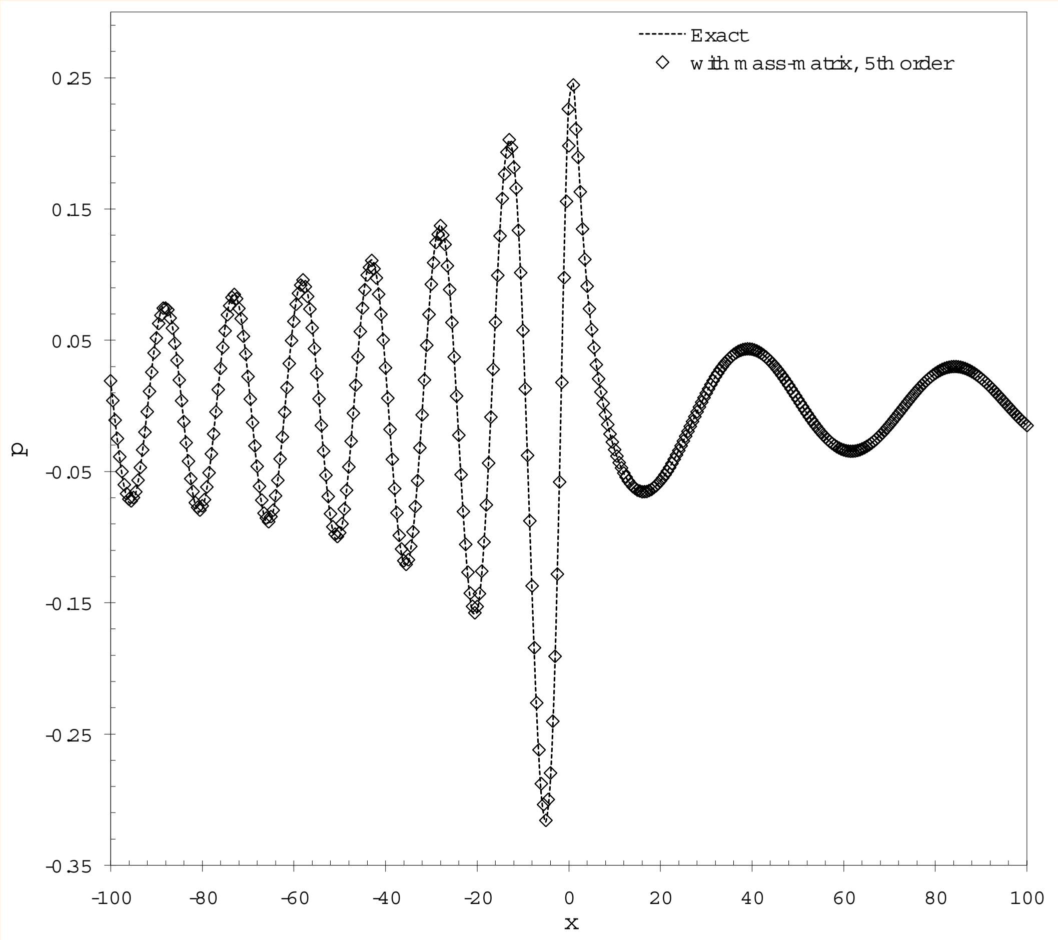
$$S = \epsilon e^{-\alpha\left[(x-x_s)^2+(y-y_s)^2\right]}\sin wt$$

A CAA example on an unstructured grid.





A CAA example on an unstructured grid.



Acoustic pressure profile across $y = 0$

- Introduction
- The FV-MLS method
- A high-order formulation for incompressible flows
- High-order Fluid-Structure-Interaction techniques
- Conclusions

A high-order formulation for incompressible flows

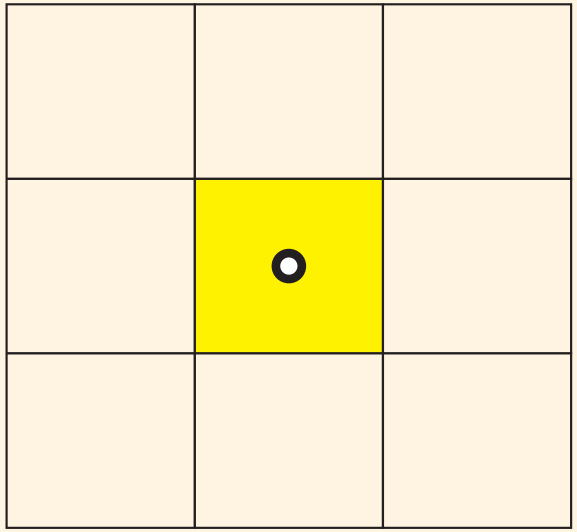
- Introduction
- Formulation
- Numerical Examples

- Introduction
- Formulation
- Numerical Examples

- Incompressibility assumption:

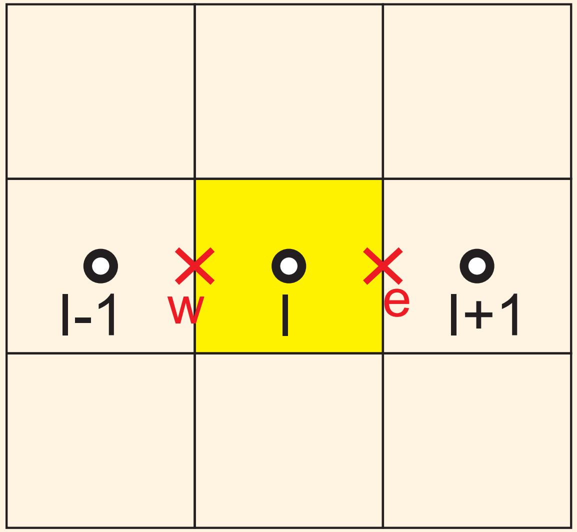
Introduction

- Checkerboard:



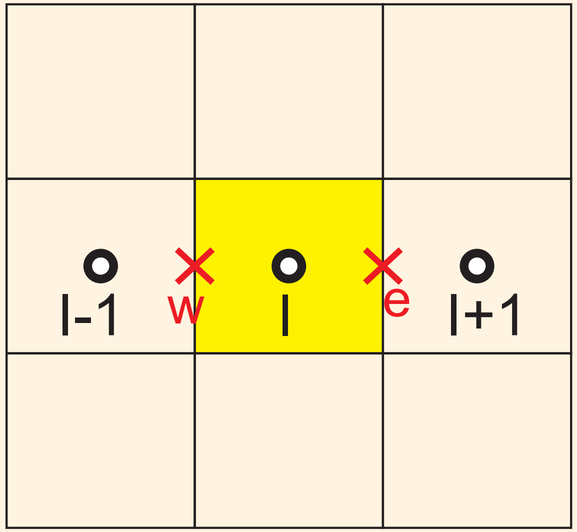
$$\int_{\Omega_I} \frac{\partial p}{\partial x} d\Omega = \sum_{j=1}^{N_f} \sum_{ig=1}^{N_G} \left[p_j \hat{n}_{xj}\right]_{ig} \mathcal{W}_{ig}$$

- Checkerboard:



$$\int_{\Omega_I} \frac{\partial p}{\partial x} d\Omega = (p\hat{n}_x)_e + (p\hat{n}_x)_w$$

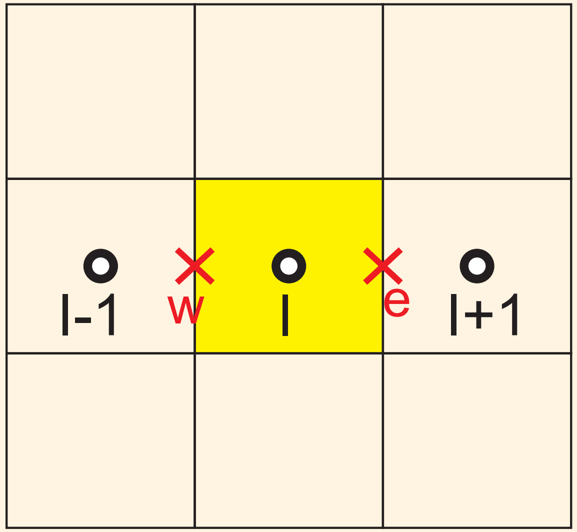
- Checkerboard:



$$\int_{\Omega_I} \frac{\partial p}{\partial x} d\Omega = (p)_e - (p)_w$$

SHARK-FV 2015 Conference
Sharing Higher-order Advanced Research Know-how on Finite Volume
Ofir, Portugal
May 18 - 22, 2015

- Checkerboard:

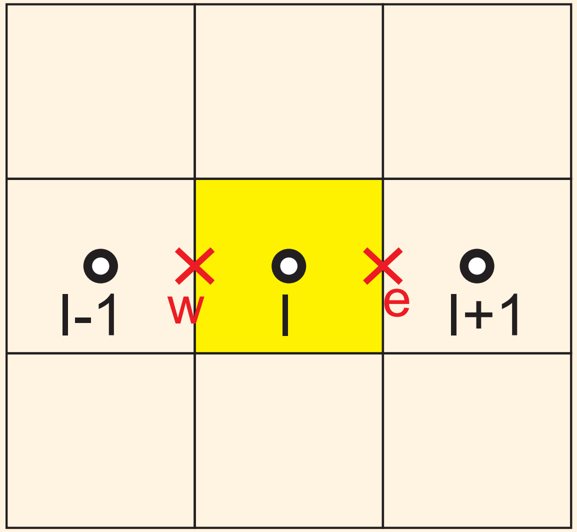


$$(p)_e = \frac{p_{i+1} + p_i}{2} \qquad (p)_w = \frac{p_i + p_{i-1}}{2}$$

$$\int_{\Omega_I} \frac{\partial p}{\partial x} d\Omega = (p)_e - (p)_w$$

Introduction

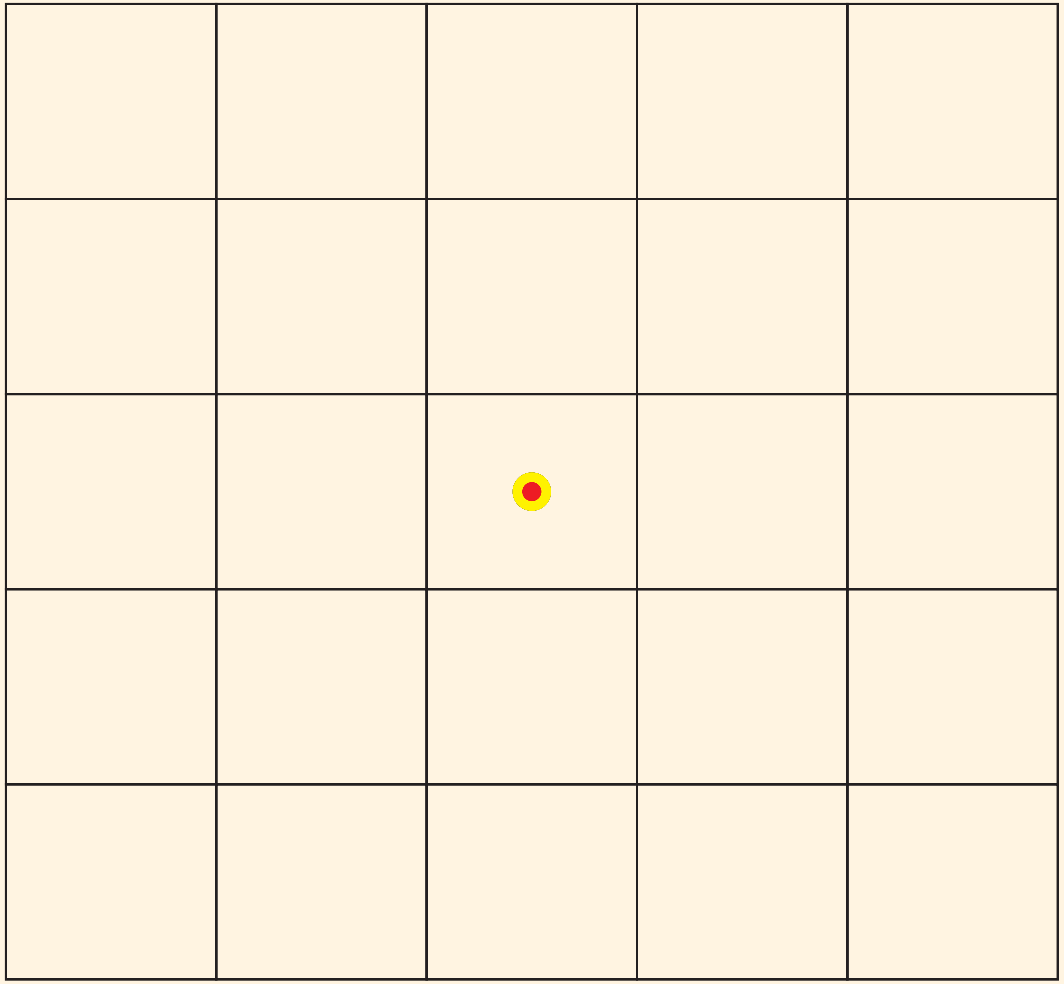
- Checkerboard:



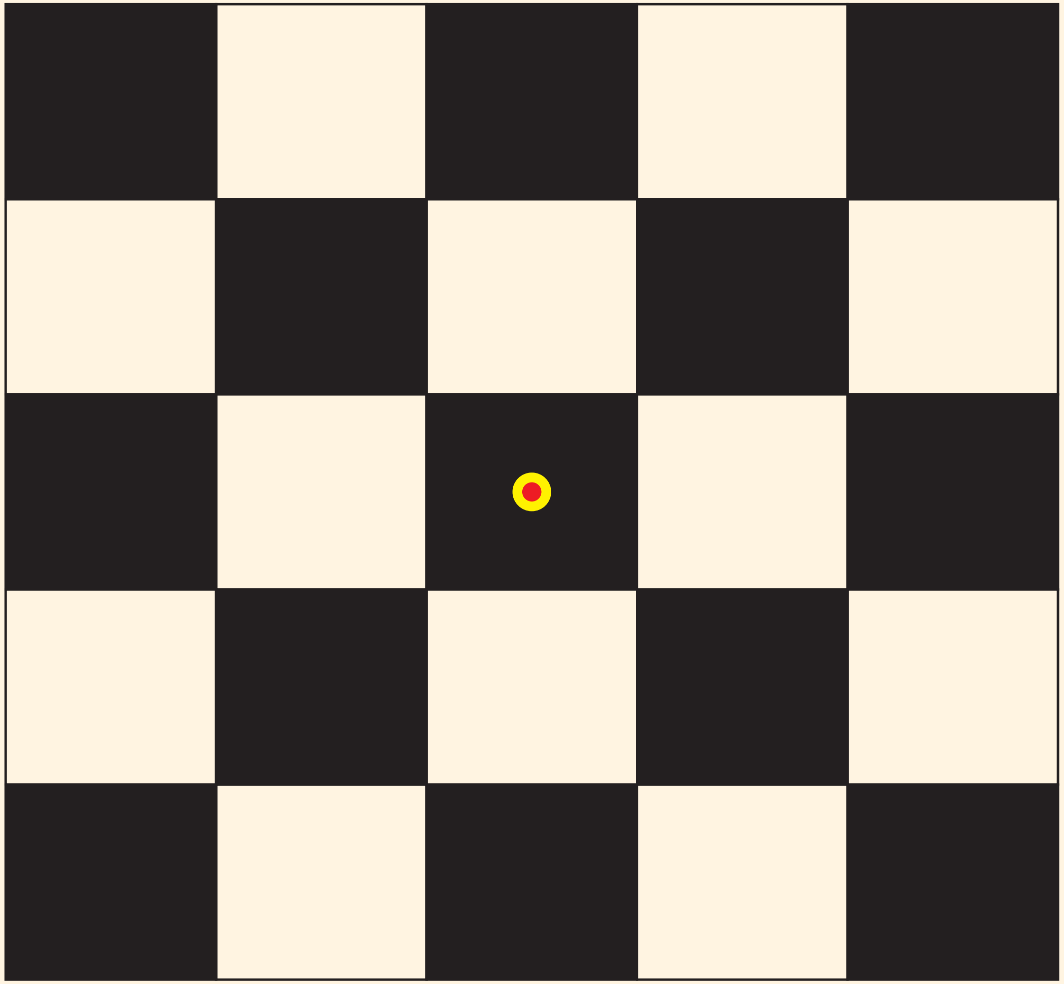
$$(p)_e = \frac{p_{i+1} + p_i}{2} \qquad (p)_w = \frac{p_i + p_{i-1}}{2}$$

$$\int_{\Omega_I} \frac{\partial p}{\partial x} d\Omega = \frac{p_{i+1} - p_{i-1}}{2}$$

- Checkerboard:



- Checkerboard:

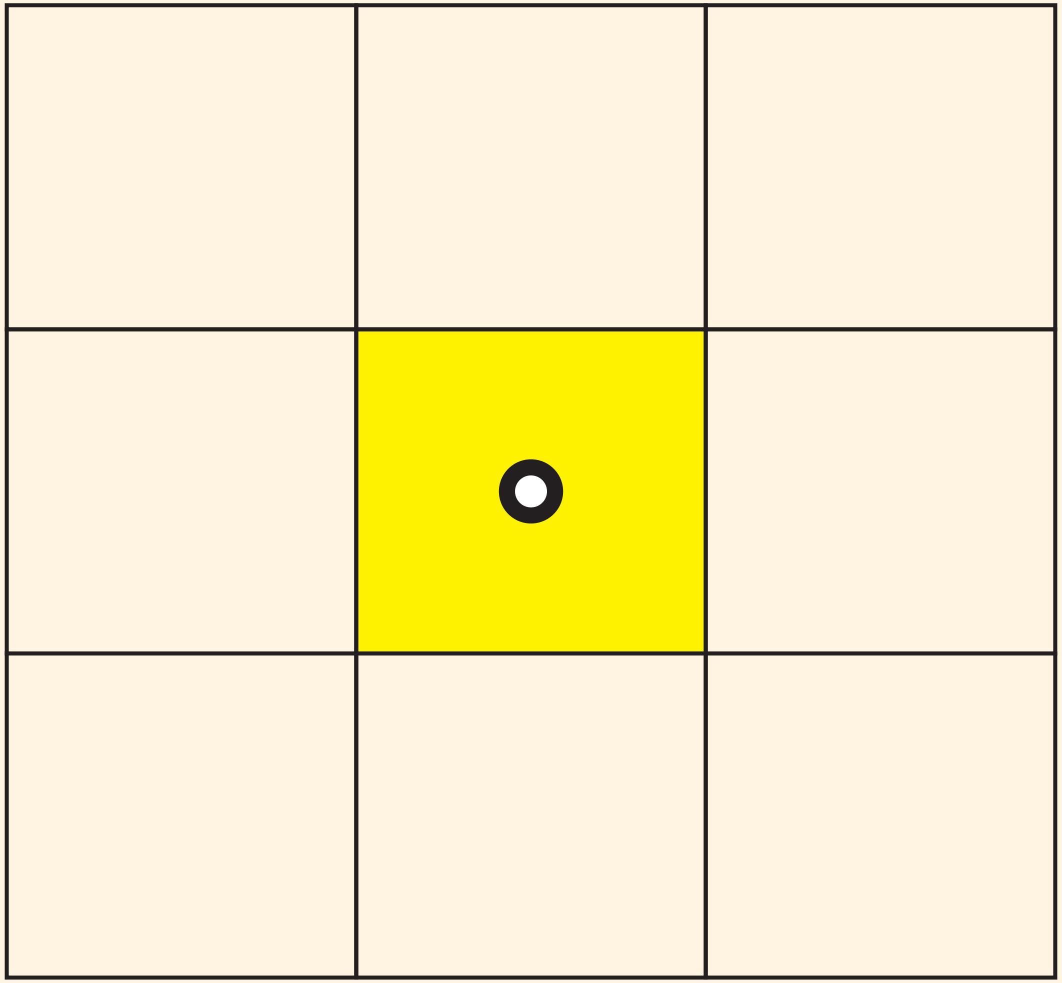


Introduction

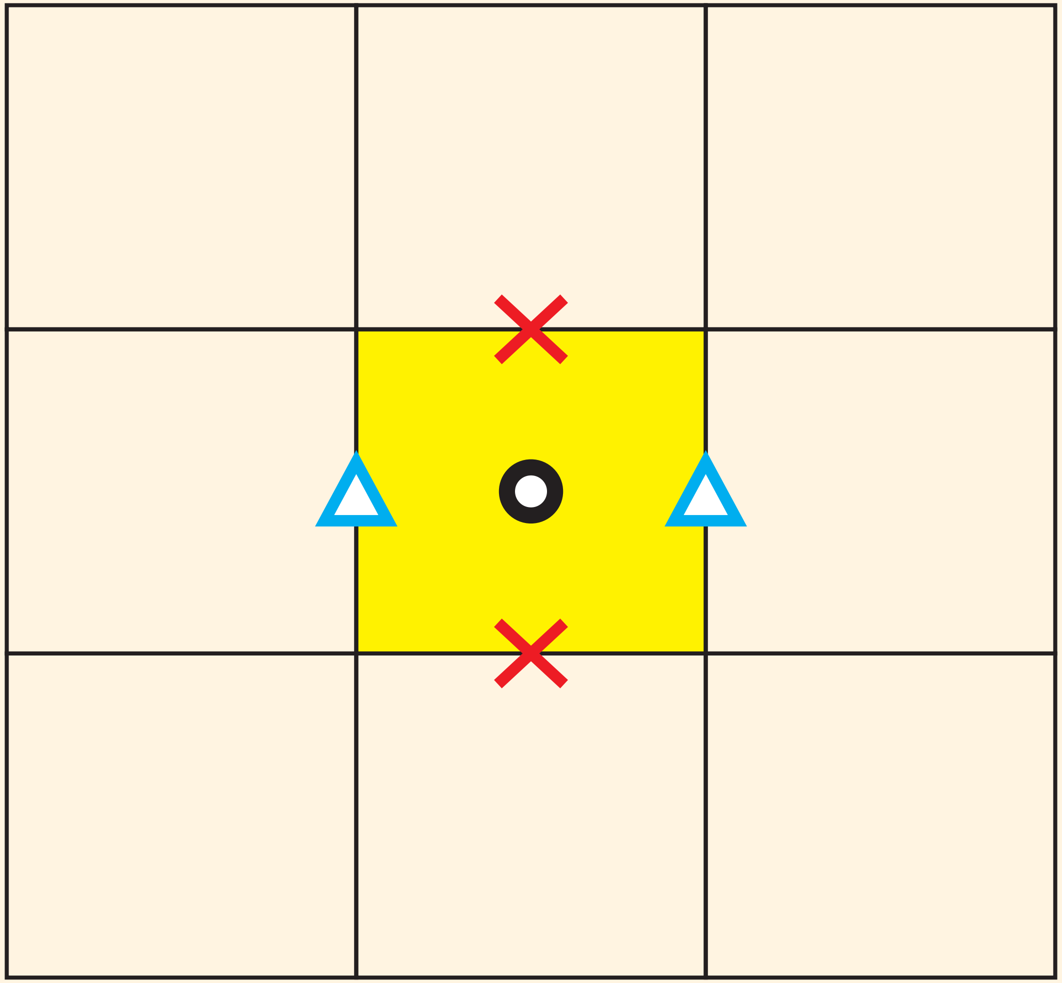
- In order to solve the checkerboard:
  - Collocated grid arrangement $\rightarrow$ Special interpolation (MIM)
  - Staggered grid arrangement $\rightarrow$ Special location of the variables

Introduction

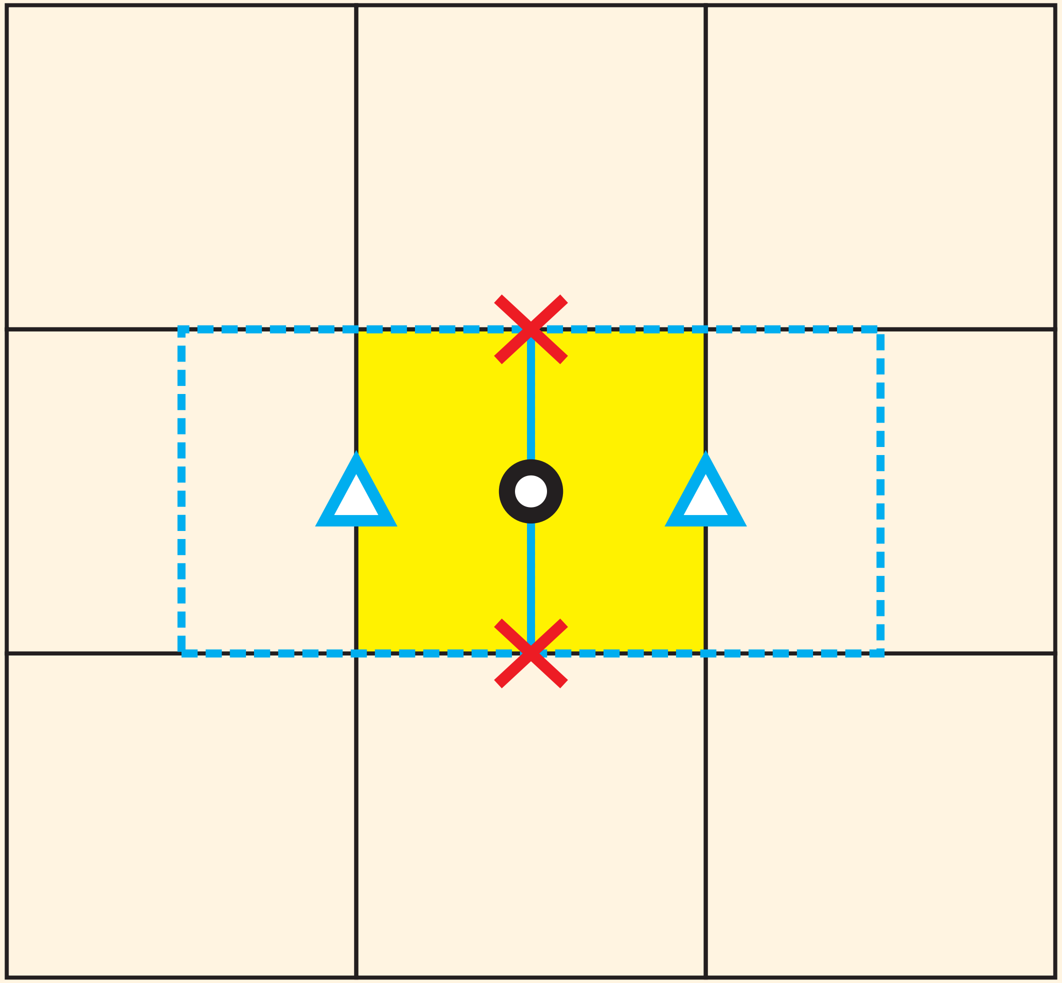
- Collocated grid arrangement $\rightarrow u, v, p$ located at cell centroid.



- Staggered grid $\rightarrow u, v, p$ located at different locations.

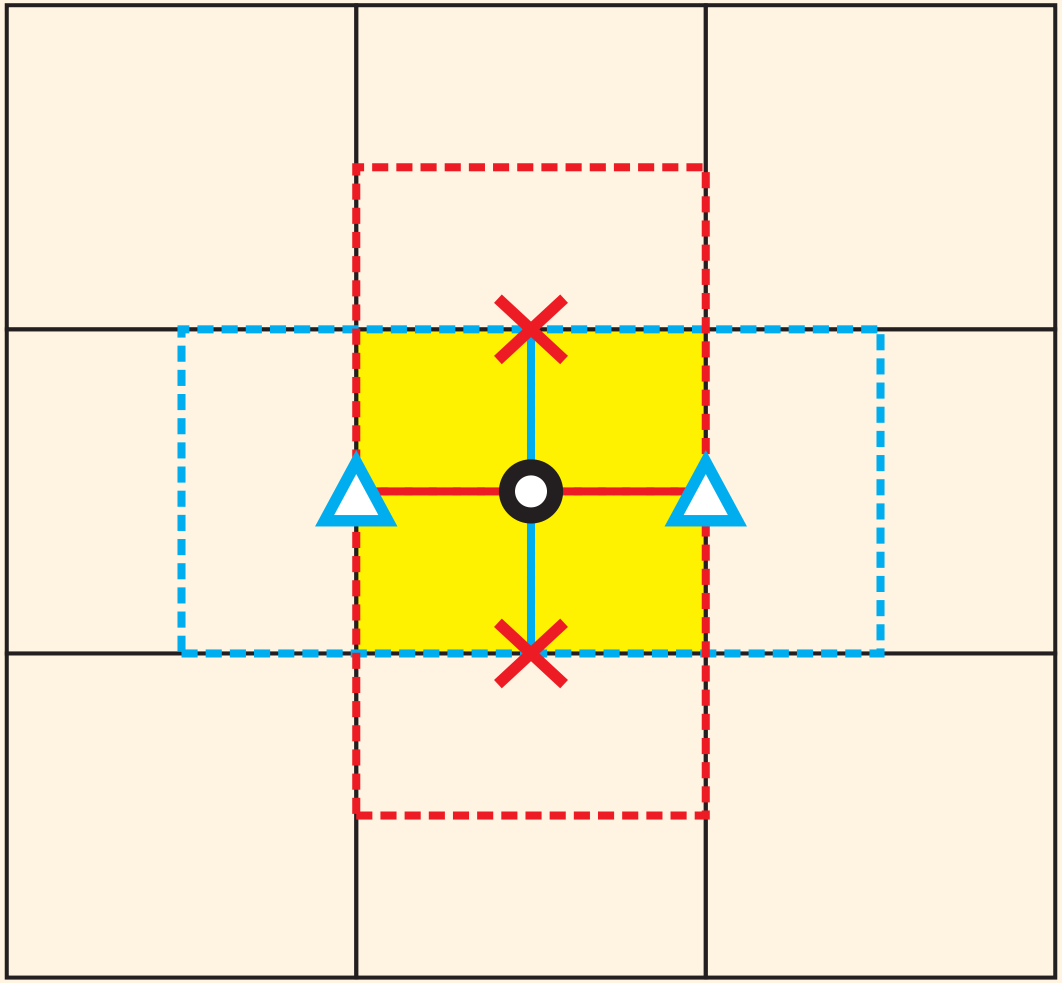


- Staggered grid $\rightarrow u$ control volume.



Introduction

- Staggered grid $\rightarrow v$ control volume.



Introduction

- **Staggered grid arrangement:**
  - Variables stored at different locations
  - No interpolations required
  - Drawback → Complex in unstructured and/or 3D grids
- **Collocated grid arrangement:**
  - Variables stored at cell centroid
  - Structured and unstructured grid
  - Drawback → Possibility of checkerboard

Introduction

- **Staggered grid arrangement:**
  - Variables stored at different locations
  - No interpolations required
  - Drawback → Complex in unstructured and/or 3D grids
- **Collocated grid arrangement:**
  - Variables stored at cell centroid
  - Structured and unstructured grid
  - Drawback → Possibility of checkerboard

Introduction

- Staggered grid arrangement:
  - Variables stored at different locations
  - No interpolations required
  - Drawback → Complex in unstructured and/or 3D grids
- Collocated grid arrangement:
  - Variables stored at cell centroid
  - Structured and unstructured grids
  - Drawback → Possibility of checkerboard
  - Special interpolation is required

- Introduction
- Formulation
- Numerical Examples

Incompressible Navier Stokes

- Incompressible Navier-Stokes:

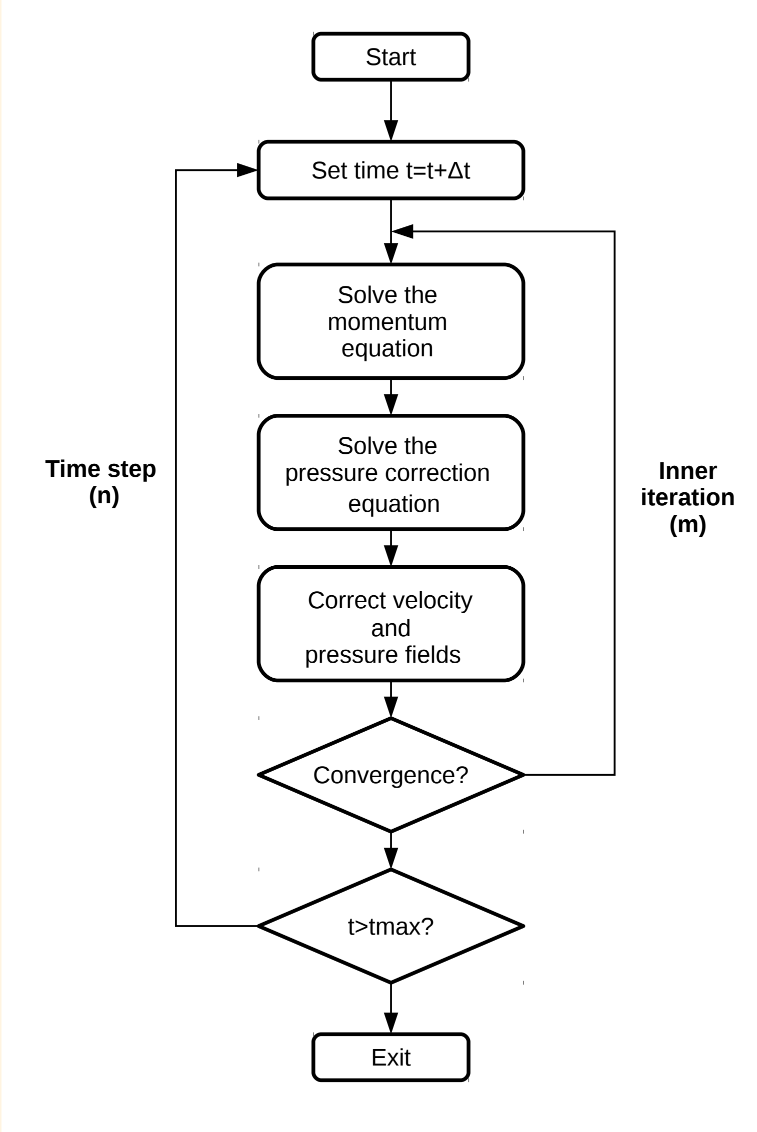
$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{U} \cdot (\nabla \boldsymbol{U}) = -\nabla p + \frac{1}{Re}(\Delta \boldsymbol{U})$$

$$\nabla \cdot \boldsymbol{U} = 0$$

where $\boldsymbol{U} = (u, v)^T$ is the velocity field, $p(x, y, t)$ is the pressure variable and Re denotes the Reynolds number.

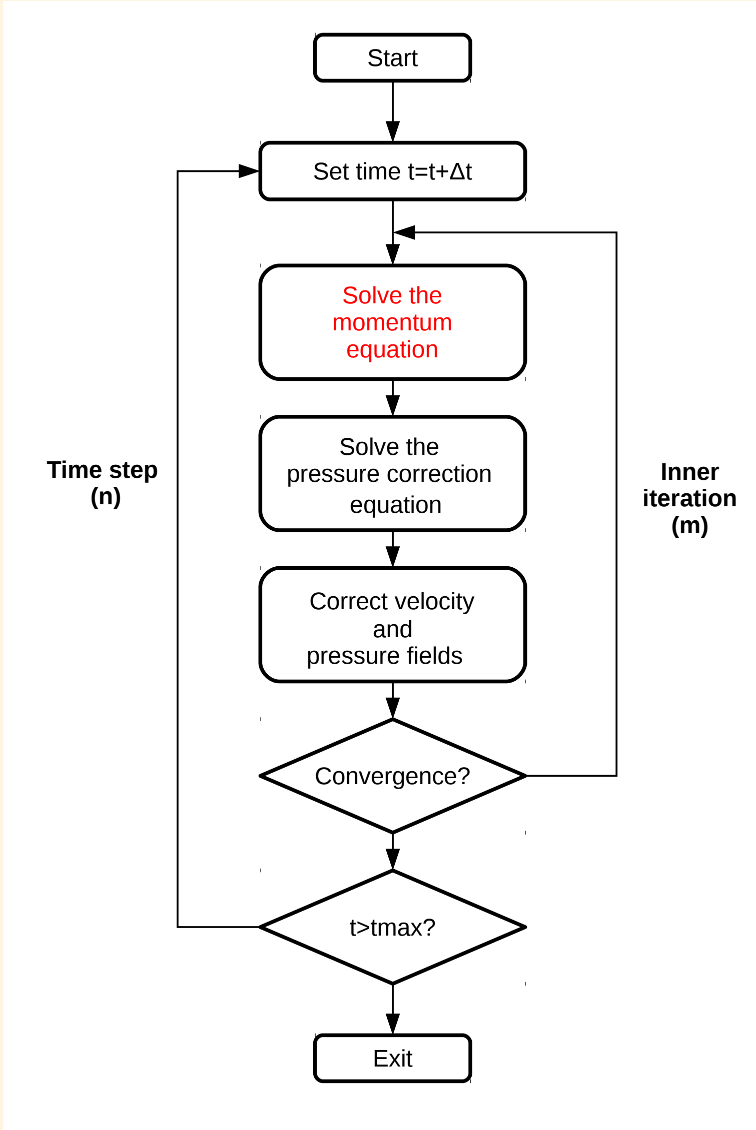
- Resolution procedure:
  - A collocated Semi-Implicit Method for Pressure Linked Equations (SIMPLE).
  - Momentum Interpolation Method to avoid checkerboard oscillations.

SIMPLE



Start
Set time t=t+Δt
Solve the momentum equation
Solve the pressure correction equation
Correct velocity and pressure fields
Convergence?
t>tmax?
Exit
Time step (n)
Inner iteration (m)

SIMPLE


Start
Set time t=t+Δt
Solve the momentum equation
Solve the pressure correction equation
Correct velocity and pressure fields
Convergence?
t>tmax?
Exit
Time step (n)
Inner iteration (m)

SIMPLE

- Momentum equation
  - Cell centered finite volume scheme

$$\int_{\Omega_I} \frac{\partial \boldsymbol{U}}{\partial t} d\Omega + \int_{\Omega_I} \boldsymbol{U} \cdot (\nabla \boldsymbol{U}) d\Omega = -\int_{\Omega_I} \nabla p d\Omega + \frac{1}{Re}\left(\int_{\Omega_I} (\Delta \boldsymbol{U}) d\Omega\right)$$

  - Discretized momentum equation

$$V_I \frac{3\boldsymbol{U}_I^{m+1,n+1} - 4\boldsymbol{U}_I^{n} + \boldsymbol{U}_I^{n-1}}{2\Delta t} + \sum_{j=1}^{N_f}\sum_{ig=1}^{N_G}\left[H_j^{m,n+1}\boldsymbol{U}_j^{m+1,n+1}\right]_{ig}\mathcal{W}_{ig} =$$

$$= -\sum_{j=1}^{N_f}\sum_{ig=1}^{N_G}\left[p_j^{m,n+1}\cdot\hat{\boldsymbol{n}}_j\right]_{ig}\mathcal{W}_{ig} + \frac{1}{Re}\sum_{j=1}^{N_f}\sum_{ig=1}^{N_G}\left[\nabla\boldsymbol{U}_j^{m+1,n+1}\cdot\hat{\boldsymbol{n}}_j\right]_{ig}\mathcal{W}_{ig}$$

SIMPLE

- Momentum equation
  - Cell centered finite volume scheme

$$\int_{\Omega_I} \frac{\partial \boldsymbol{U}}{\partial t} d\Omega + \int_{\Omega_I} \boldsymbol{U} \cdot (\nabla \boldsymbol{U}) d\Omega = -\int_{\Omega_I} \nabla p d\Omega + \frac{1}{Re} \left(\int_{\Omega_I} (\Delta \boldsymbol{U}) d\Omega \right)$$

  - Discretized momentum equation

$$V_I \frac{3\boldsymbol{U}_I^{m+1,n+1} - 4\boldsymbol{U}_I^{n} + \boldsymbol{U}_I^{n-1}}{2\Delta t} + \sum_{j=1}^{N_f} \sum_{ig=1}^{N_G} \left[H_j^{m,n+1} \boldsymbol{U}_j^{m+1,n+1} \right]_{ig} \mathcal{W}_{ig} =$$

$$= -\sum_{j=1}^{N_f} \sum_{ig=1}^{N_G} \left[p_j^{m,n+1} \cdot \hat{\boldsymbol{n}}_j \right]_{ig} \mathcal{W}_{ig} + \frac{1}{Re} \sum_{j=1}^{N_f} \sum_{ig=1}^{N_G} \left[\nabla \boldsymbol{U}_j^{m+1,n+1} \cdot \hat{\boldsymbol{n}}_j \right]_{ig} \mathcal{W}_{ig}$$

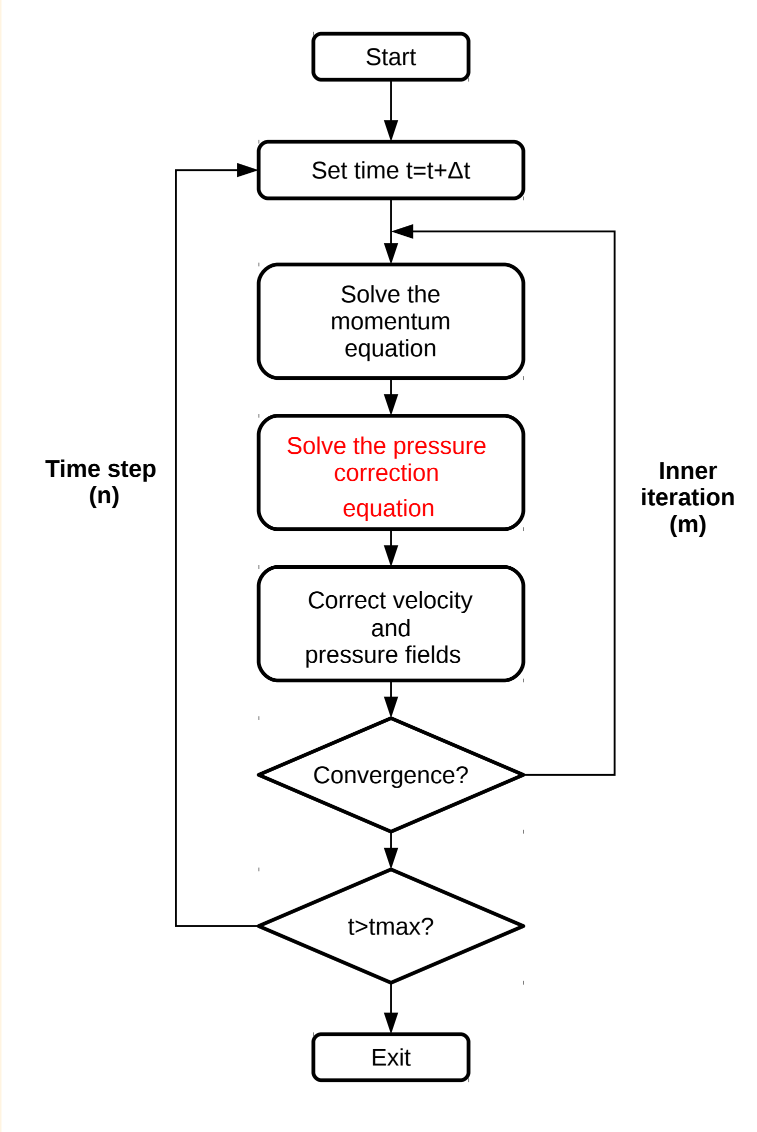
SIMPLE

- Higher-order approximations are made using MLS:
  - Pressure term $$p_j = \sum_{k=1}^{n\boldsymbol{x}} N_k^g(\boldsymbol{x}_j) p_k$$
  - Diffusive term $$\nabla \boldsymbol{U}_j = \sum_{l=1}^{n\boldsymbol{x}} \nabla N_l^g(\boldsymbol{x}_j) \boldsymbol{U}_l$$
  - Convective term $\Rightarrow$ Deferred correction approach

$$\boldsymbol{U}_j = \left(\boldsymbol{U}_j^{LO}\right)^{m+1,n+1} + \left(\boldsymbol{U}_j^{HO} - \boldsymbol{U}_j^{LO}\right)^{m,n+1}$$

$$\boldsymbol{U}_j^{LO} = \begin{cases} \boldsymbol{U}_I & , H_j \geq 0 \\ \boldsymbol{U}_N & , H_j < 0 \end{cases} \qquad \boldsymbol{U}_j^{HO} = \sum_{k=1}^{n\boldsymbol{x}} N_k^g(\boldsymbol{x}_j) \boldsymbol{U}_k$$

SIMPLE


Start
Set time t=t+Δt
Solve the momentum equation
Solve the pressure correction equation
Correct velocity and pressure fields
Convergence?
t>tmax?
Exit
Time step (n)
Inner iteration (m)

SIMPLE

- Pressure correction equation

$$\int_{\Omega_I} \nabla \cdot \boldsymbol{U} d\Omega = \int_{\Gamma_I} \boldsymbol{U} \cdot \boldsymbol{n}_j d\Gamma = \sum_{j=1}^{N_f} \sum_{ig=1}^{N_G} \left[\hat{\boldsymbol{U}}_j \cdot \hat{\boldsymbol{n}}_j \right]_{ig} \mathcal{W}_{ig}$$

- In order to avoid checkerboard oscillations ⇒ Momentum Interpolation Method (MIM)

$$\hat{\boldsymbol{U}}_j = \boldsymbol{U}_j^* + \left(\frac{V_I}{a_I} \right)_j \left[\left(\overline{\nabla p_I} \right)_j - \nabla p_j \right]$$

- The MIM was proposed by Rhie and Chow in 1983.

- Pressure correction equation
  - Checkerboard oscillations⇒Momentum Interpolation Method

$$\hat{\boldsymbol{U}}_j = \boldsymbol{U}_j^* + \left(\frac{V_I}{a_I}\right)_j \left[\left(\overline{\nabla p_I}\right)_j - \nabla p_j\right]$$

- ▷ These terms are usually obtained at integration point j using linear interpolation.
- ▷ We propose to use higher-order approximations using MLS

$$\boldsymbol{U}_j^* = \sum_{k=1}^{n\boldsymbol{x}} N_k^g(\boldsymbol{x}_j)\boldsymbol{U}_k^* \qquad \left(\frac{V_I}{a_I}\right)_j = \sum_{k=1}^{n\boldsymbol{x}} N_k^g(\boldsymbol{x}_j)\left(\frac{V_I}{a_I}\right)_k$$

$$\left(\overline{\nabla p_I}\right)_j = \sum_{k=1}^{n\boldsymbol{x}} N_k^g(\boldsymbol{x}_j)\nabla p_k \qquad \nabla p_j = \sum_{l=1}^{n\boldsymbol{x}} \nabla N_l^g(\boldsymbol{x}_j) p_l$$

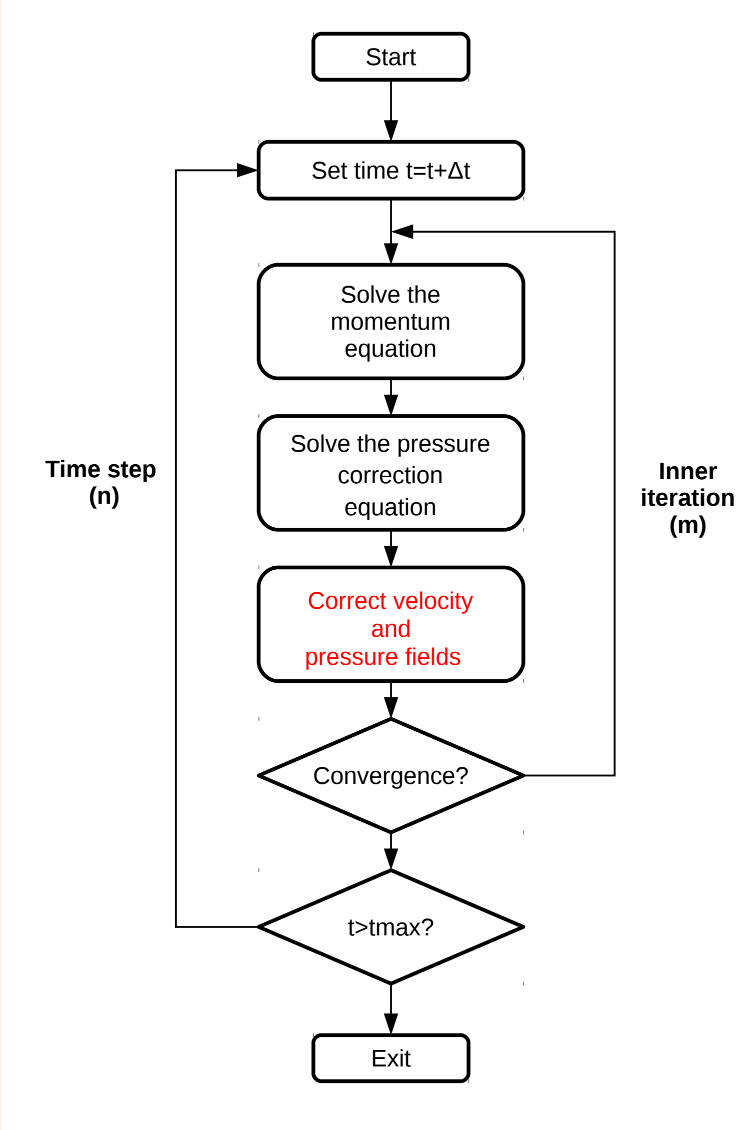
SIMPLE

- Pressure correction equation
  - A pressure correction equation is solved in order to impose the continuity "constraint".

$$\sum_{j=1}^{N_f}\sum_{ig=1}^{N_G}\left[\hat{\boldsymbol{U}}_j\cdot\hat{\boldsymbol{n}}_j\right]_{ig}\mathcal{W}_{ig}-\sum_{j=1}^{N_f}\sum_{ig=1}^{N_G}\left[\left(\frac{V_I}{a_I}\right)_j\left(\nabla p^{'}\right)_j\cdot\hat{\boldsymbol{n}}_j\right]_{ig}\mathcal{W}_{ig}=0$$

  - The pressure correction, $p^{'}$, is the unknown.
  - Approximations at integration point are obtained with MLS.

SIMPLE



Start
Set time t=t+Δt
Solve the momentum equation
Solve the pressure correction equation
Correct velocity and pressure fields
Convergence?
t>tmax?
Exit
Time step (n)
Inner iteration (m)

SIMPLE

- Correct velocity and pressure fields at cell centroids as

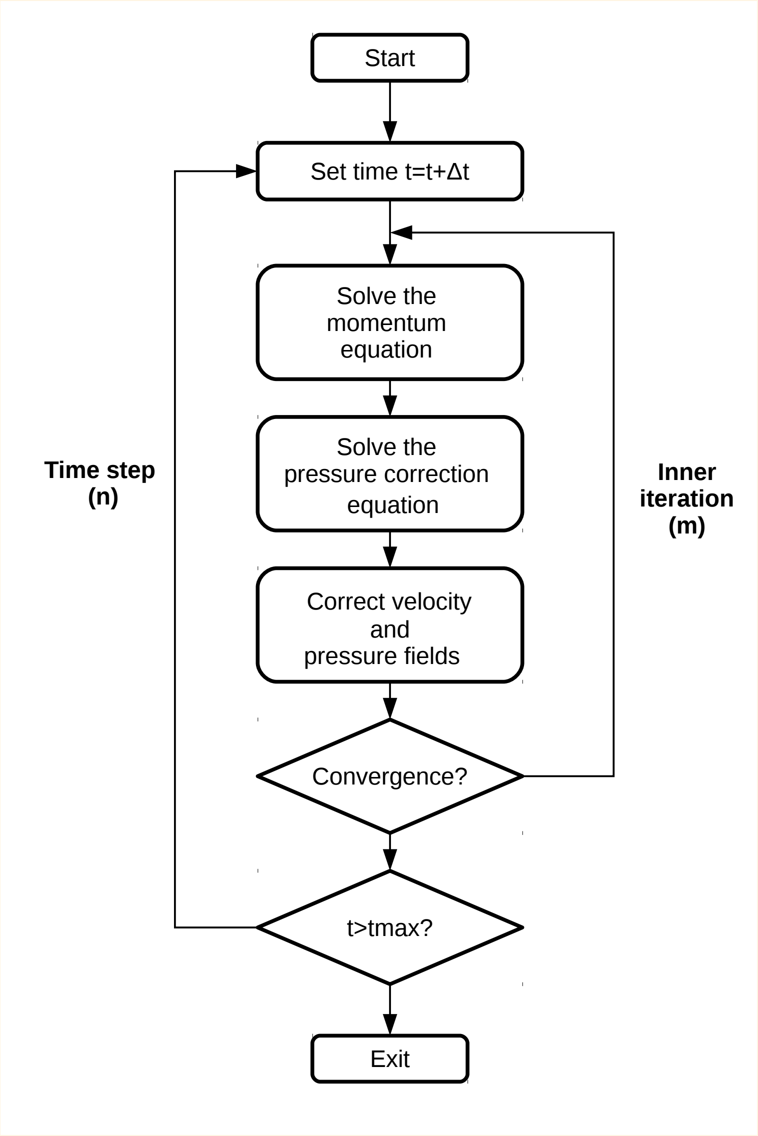
$$\boldsymbol{U}^{m+1,n+1} = \boldsymbol{U}^{*} + \boldsymbol{U}^{'} = \boldsymbol{U}^{*} - \frac{V_I}{a_I}\left(\nabla p^{'}\right)_I$$

$$p^{m+1,n+1} = p^{m,n+1} + \left(p^{'}\right)^{m+1,n+1}$$

  - The value $\left(\nabla p^{'}\right)_I$ is approximated at cell centroid using MLS

$$\nabla p^{'}_I = \sum_{l=1}^{n\boldsymbol{x}} \nabla N_l^g(\boldsymbol{x}_j) p^{'}_l$$

SIMPLE


Start
Set time t=t+Δt
Solve the momentum equation
Solve the pressure correction equation
Correct velocity and pressure fields
Convergence?
t>tmax?
Exit
Time step (n)
Inner iteration (m)

Numerical Examples

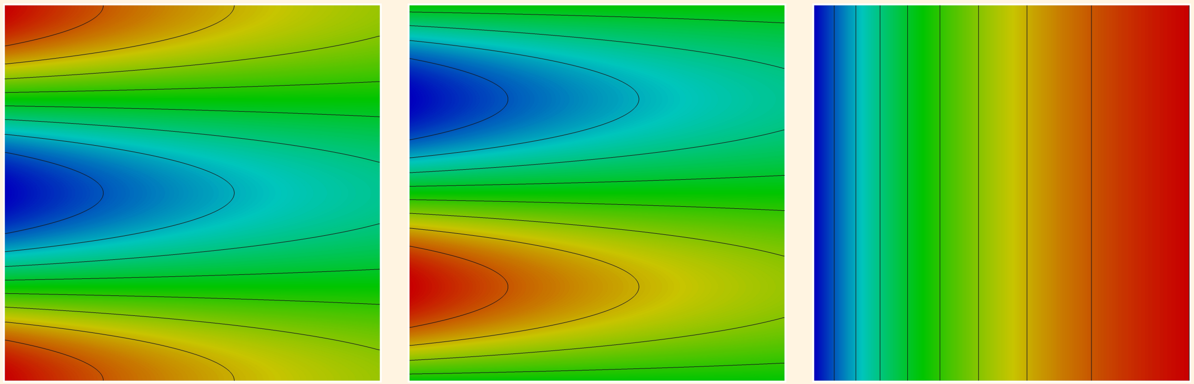
- Kovasznay Flow.

$$u(x,y) = 1 - e^{\alpha x} \cos(2\pi y)$$

$$v(x,y) = \frac{\alpha}{2\pi} e^{\alpha x} \sin(2\pi y) \qquad \alpha = \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}$$

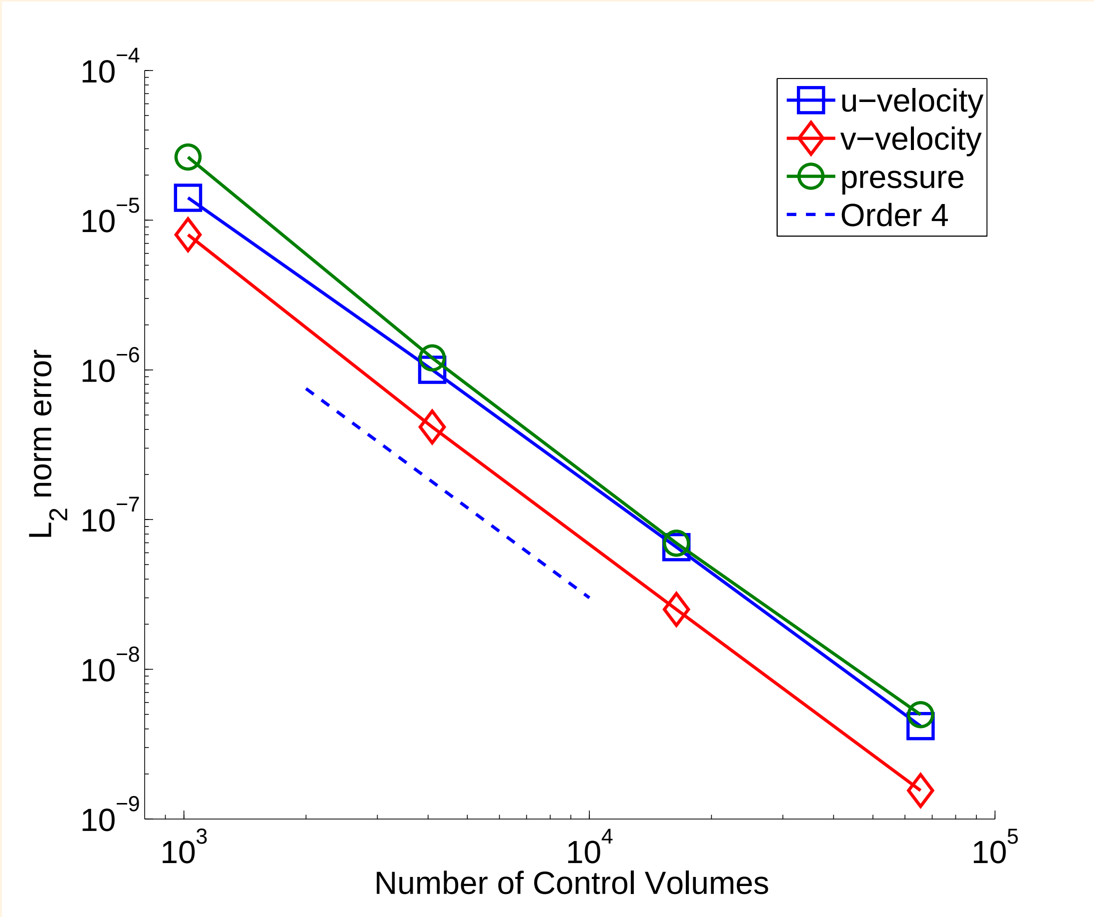
$$p(x,y) = \frac{1}{2}\left(1 - e^{2\alpha x}\right)$$

  - Domain $\Omega = [-0.5, 0.5] \times [0.5, 0.5]$. Re=40



u v p

- Kovasznay Flow.



- The formal order of accuracy is recovered

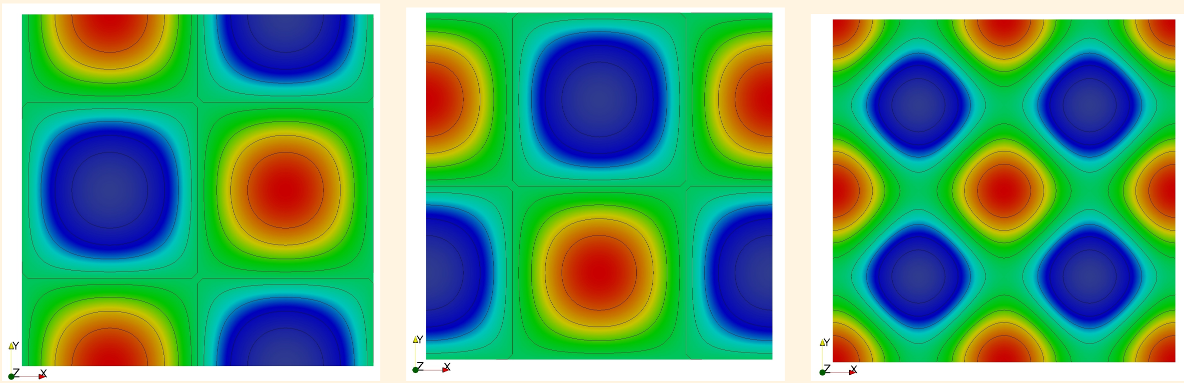
2D Taylor-Green Flow.

$$u(x,y,t) = e^{\frac{-2t}{Re}} \cos(y) \sin(x)$$

$$v(x,y,t) = -e^{\frac{-2t}{Re}} \cos(x) \sin(y)$$

$$p(x,y,t) = \frac{e^{\frac{-4t}{Re}}}{4} \left(\cos(2x) + \cos(2y)\right)$$

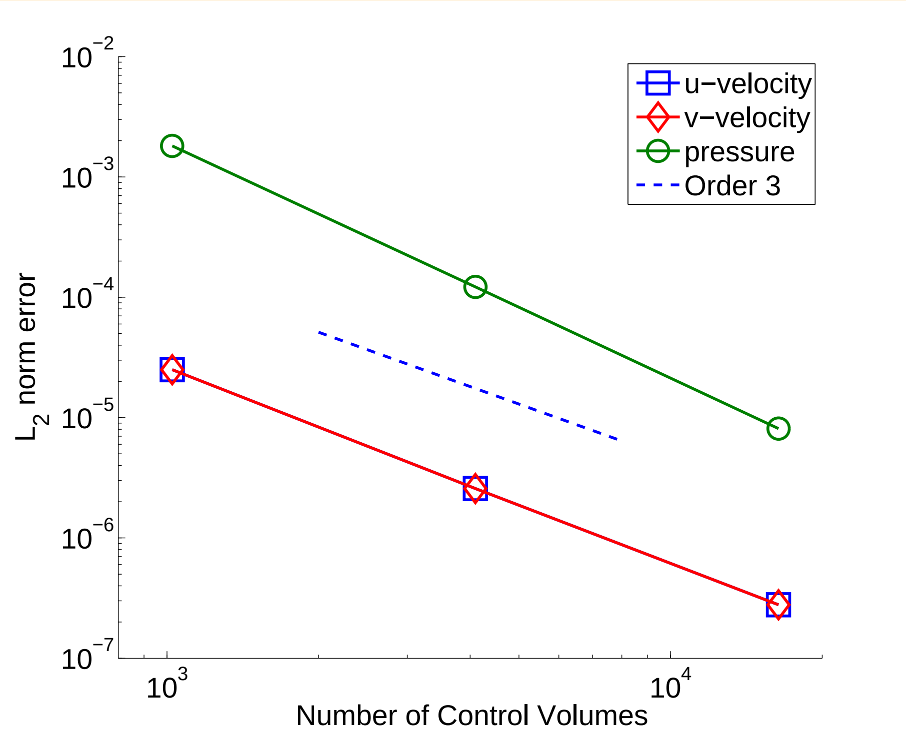
- Domain $\Omega = [0, 2\pi] \times [0, 2\pi]$. Re=100

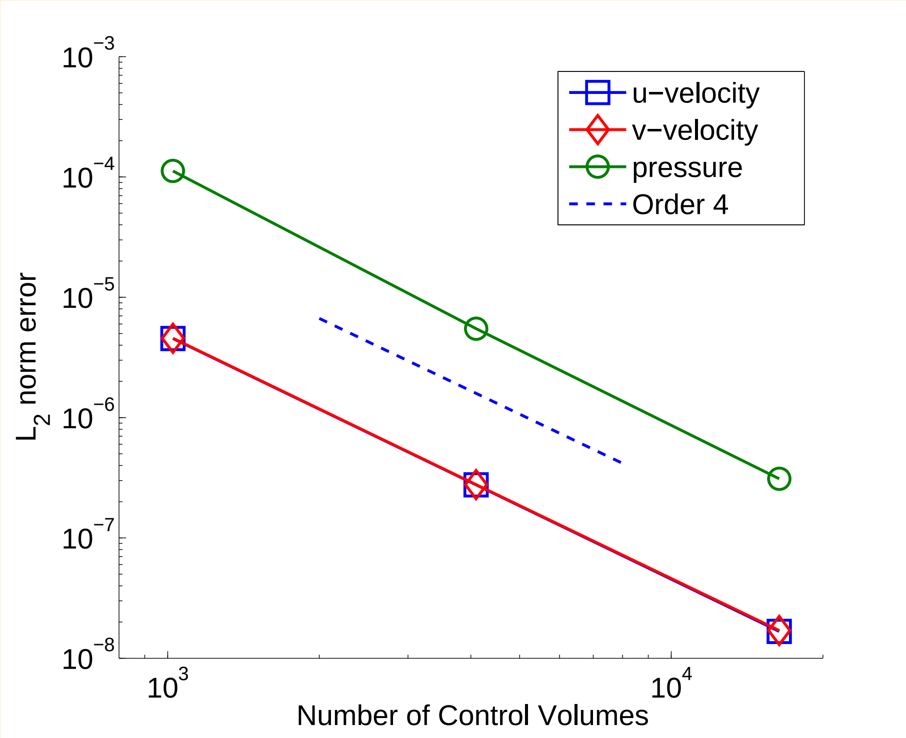


u v p

Numerical Examples

▶ 2D Taylor-Green Flow.

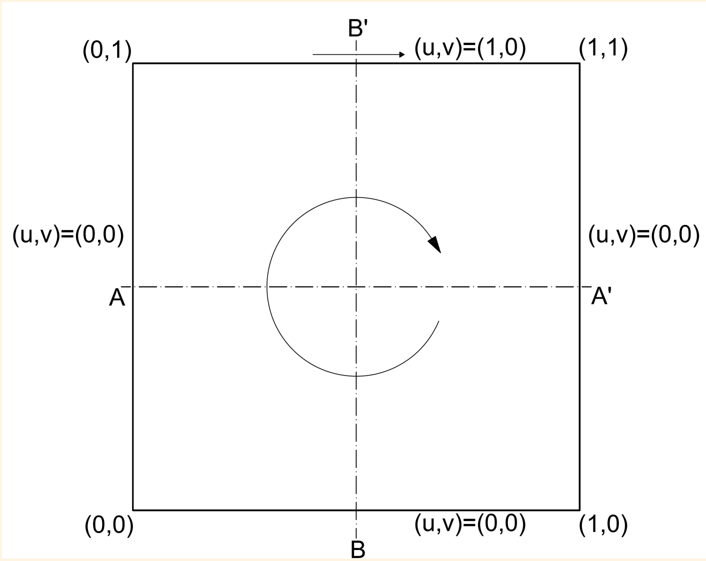


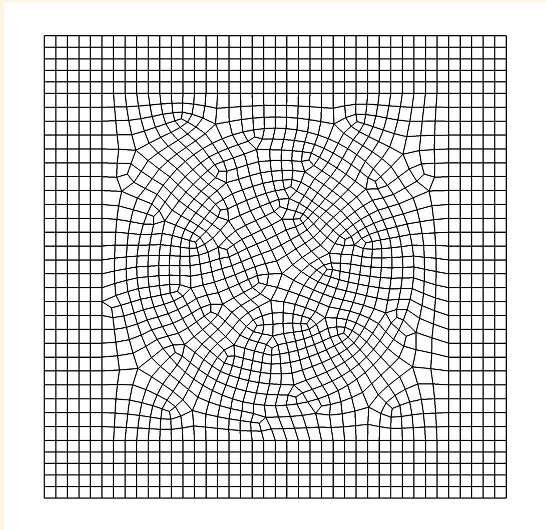


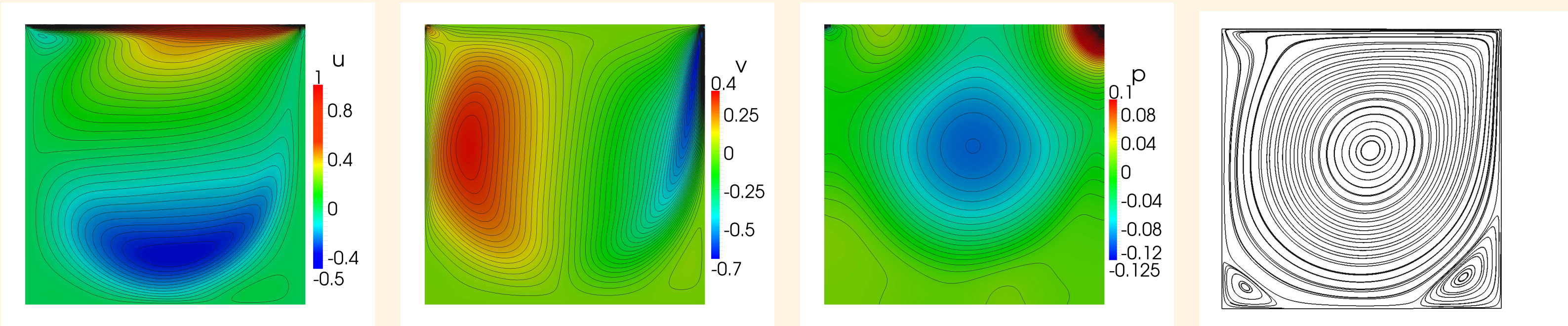
- The formal order of accuracy is recovered

Numerical Examples

- Cavity Flow



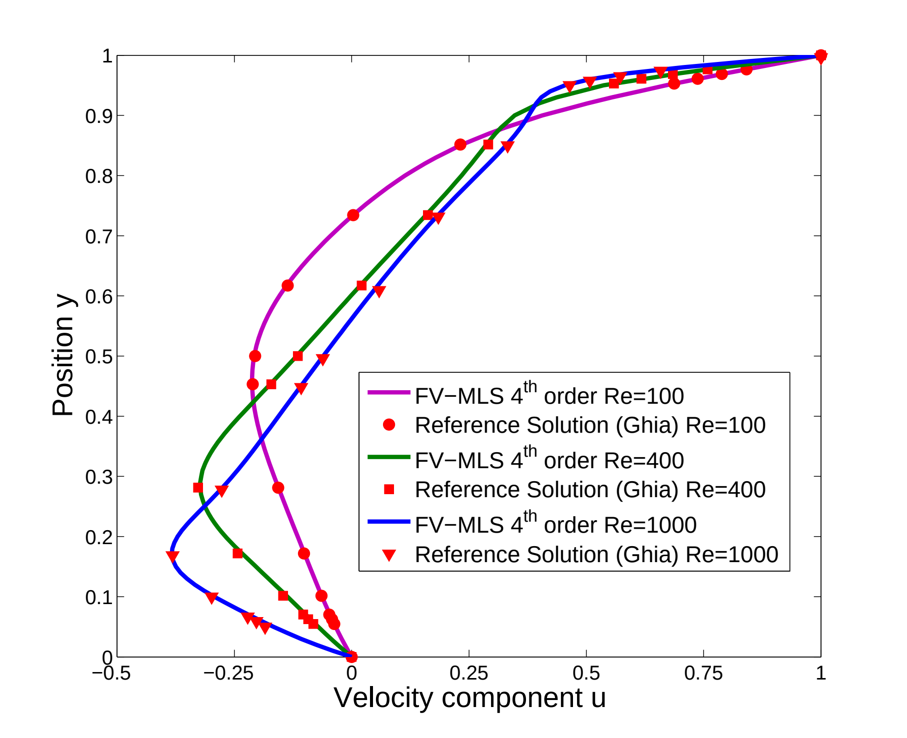


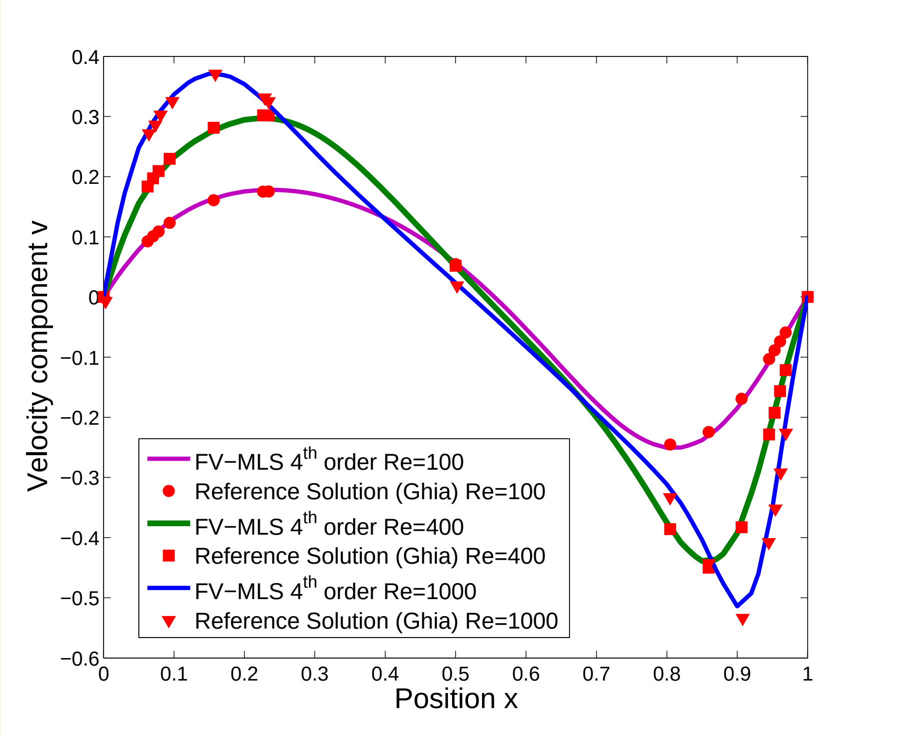


Re=1000

Numerical Examples

- Cavity Flow. 1635 cells.

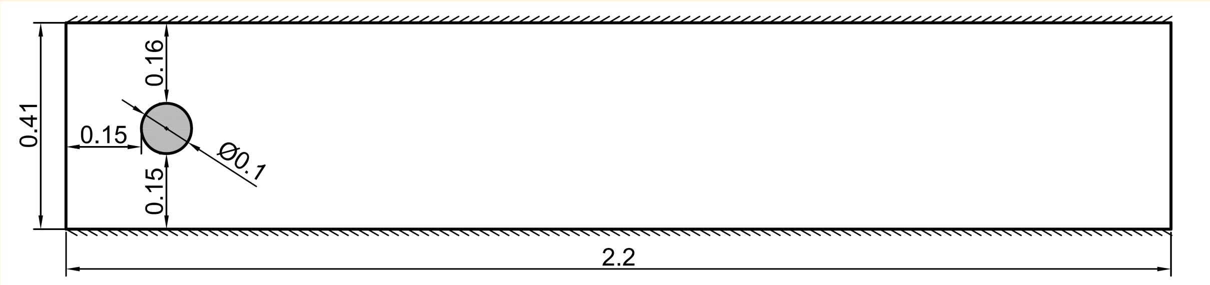




- Excellent agreement with the reference solution for different Reynolds number. The reference solution is obtained on a 128x128 structured mesh (16384 cells).

Numerical Examples

- Laminar Flow around a cylinder
  - Benchmark proposed by Schäfer and Turek.



  - Parabolic velocity profile at inlet

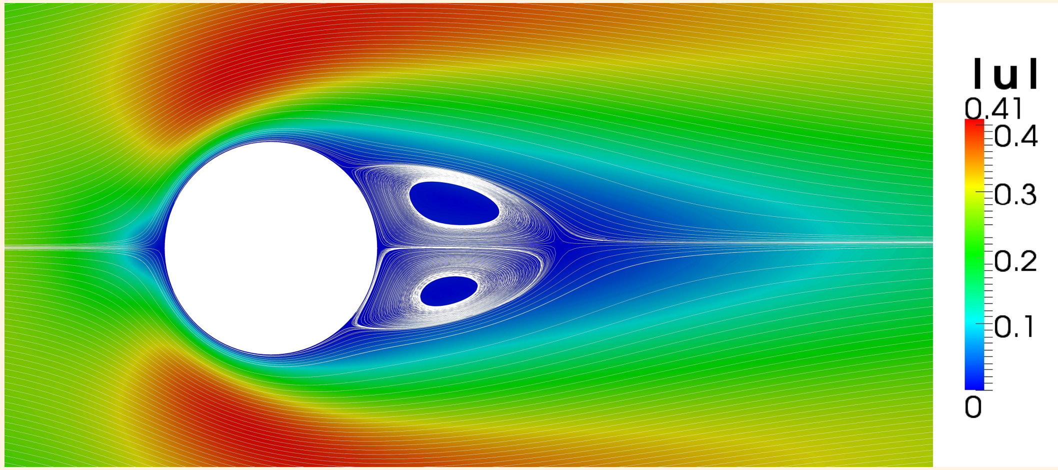
$$u(0,y) = \frac{4U_m y(H-y)}{H^2}, v(0,y) = 0$$

  - Two test cases
    - Reynolds 20
    - Reynolds 100

Reference Solution: Schäfer, M., Turek, S., *Benchmark Computations of Laminar Flow Around a Cylinder*, Notes on Numerical Fluid Mechanics, Volume 52 , pp. 547-566, 1996.

Laminar Flow around a cylinder

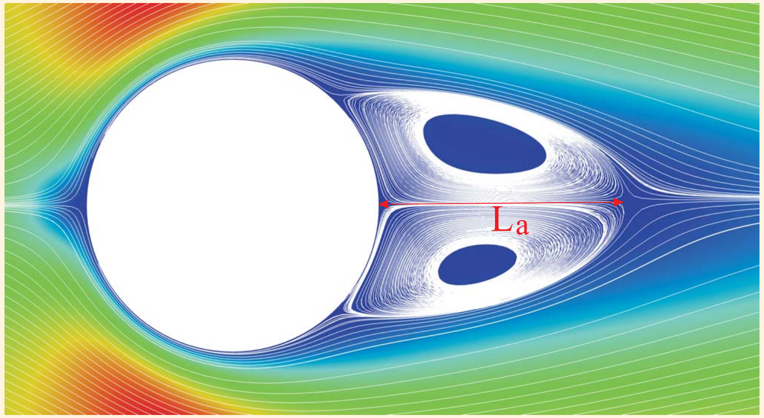
- Reynolds 20



Mesh	Order	C_D	C_L	L_a	Δp
Mesh A	2	5.5869	0.0087	0.0881	0.1149
(4968 cells)	3	5.5919	0.0108	0.0851	0.1161
Mesh B	2	5.5817	0.0113	0.0851	0.1168
(19079 cells)	3	5.5859	0.0107	0.0845	0.1174
Upper bound	—	5.5900	0.0110	0.0852	0.1176
Lower bound	—	5.5700	0.0104	0.0842	0.1172

Laminar Flow around a cylinder

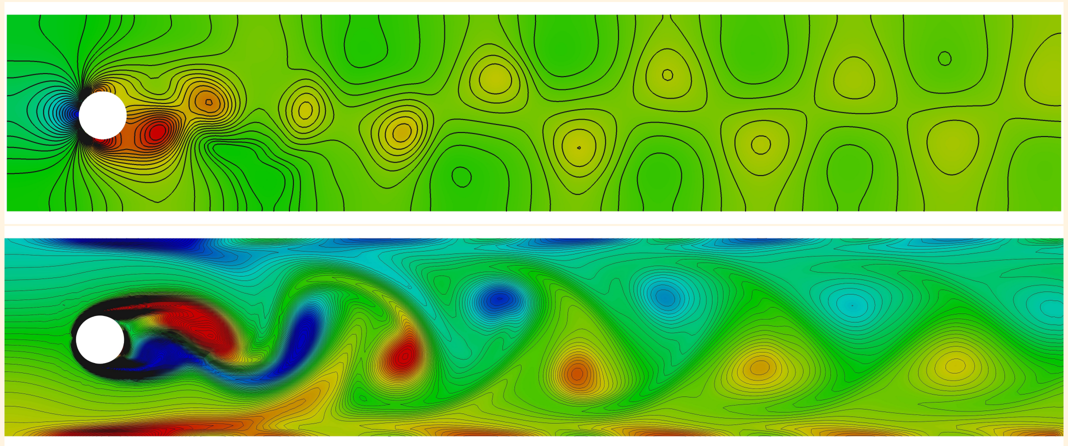
- Reynolds 20



Mesh	Order	C_D	C_L	L_a	Δp
Mesh A	2	5.5869	0.0087	0.0881	0.1149
(4968 cells)	3	5.5919	0.0108	0.0851	0.1161
Mesh B	2	5.5817	0.0113	0.0851	0.1168
(19079 cells)	3	5.5859	0.0107	0.0845	0.1174
Upper bound	–	5.5900	0.0110	0.0852	0.1176
Lower bound	–	5.5700	0.0104	0.0842	0.1172

Laminar Flow around a cylinder

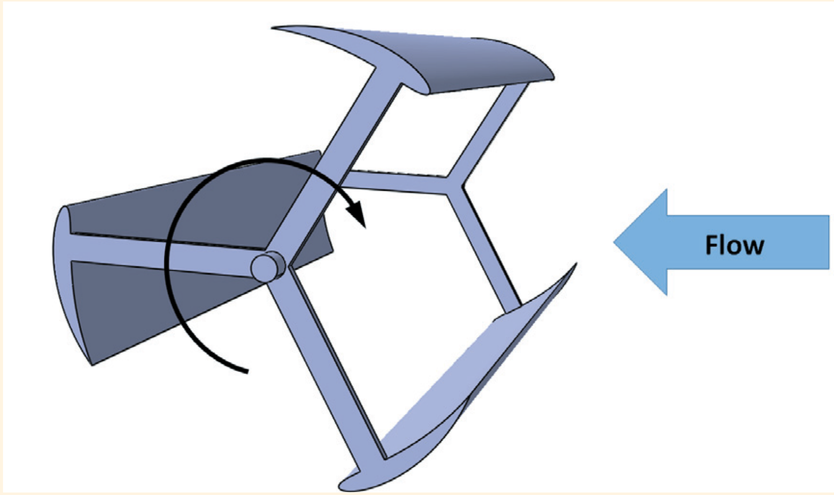
- Reynolds 100



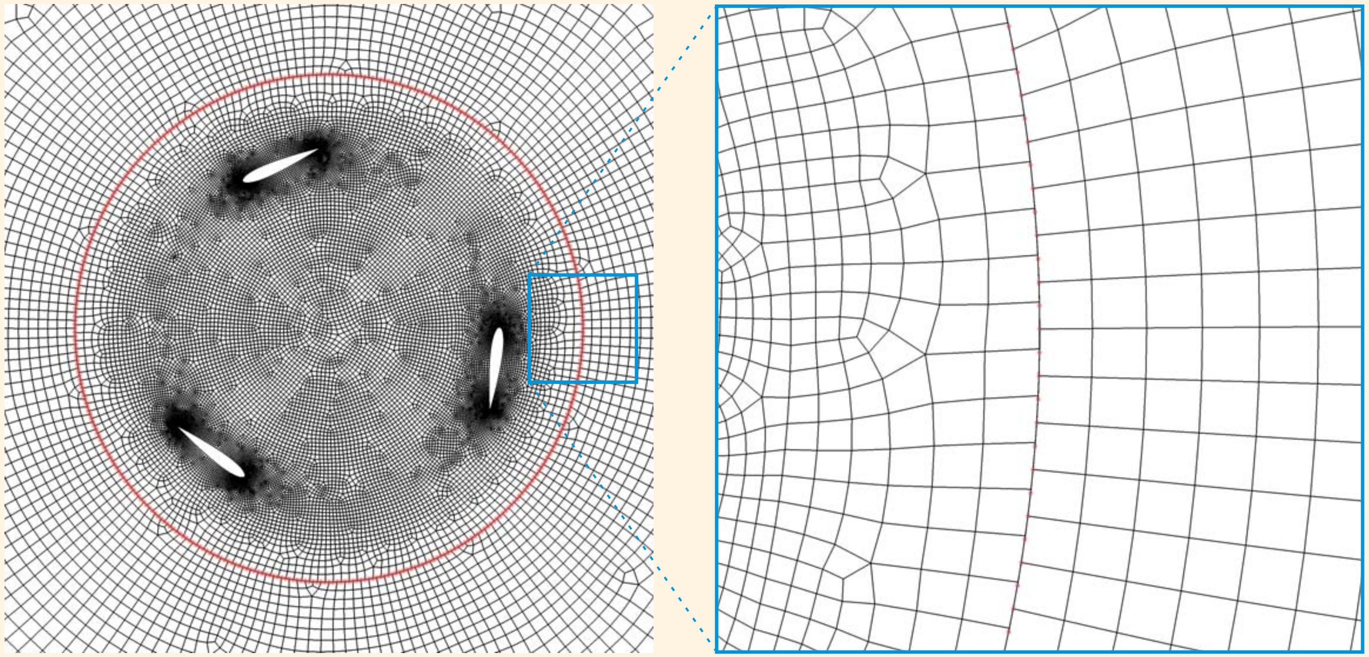
Mesh	Order	C_{Dmax}	C_{Lmax}	St	Δp
Mesh A (4968 cells)	2	3.2741	1.2246	0.2825	2.3548
	3	3.2986	1.0451	0.2924	2.3962
Mesh B (19079 cells)	2	3.2702	1.0662	0.2952	2.4731
	3	3.2380	0.9985	0.3008	2.4858
Upper bound	–	3.2400	1.0100	0.3050	2.5000
Lower bound	–	3.2200	0.9900	0.2950	2.4600

- Introduction
- The FV-MLS method
- A high-order formulation for incompressible flows
- High-order Fluid-Structure-Interaction techniques
- Conclusions

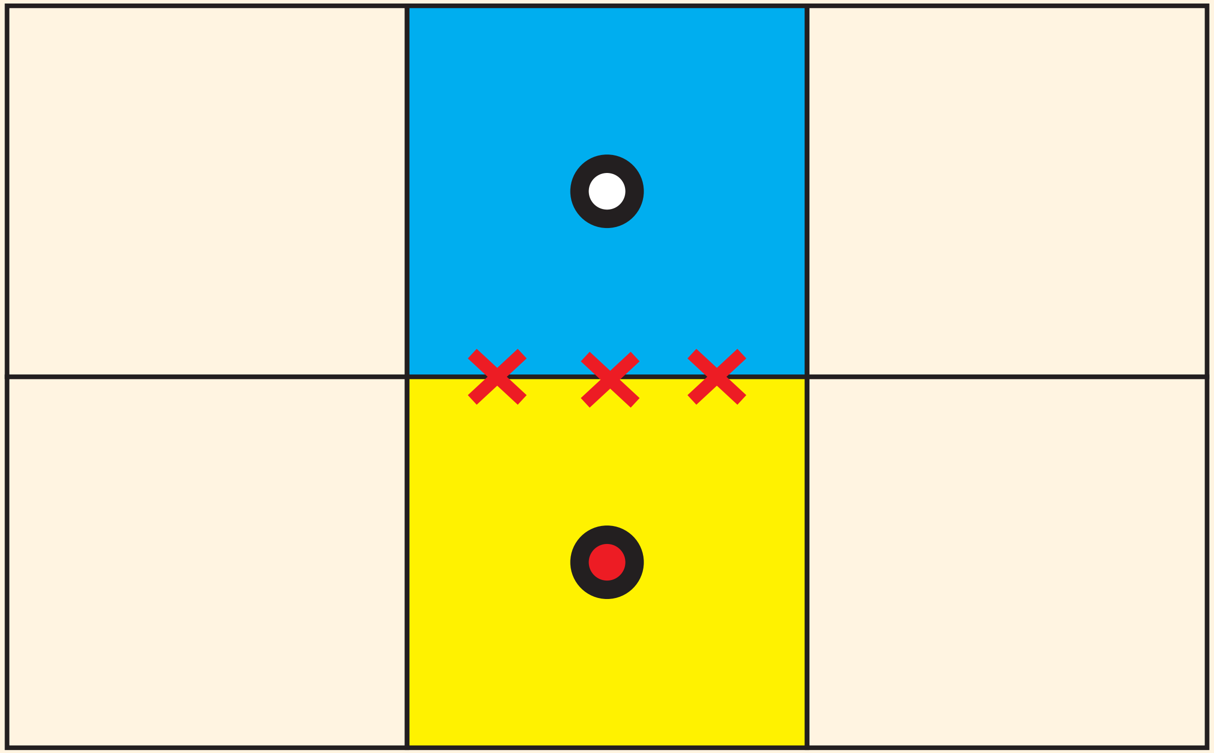
- Incompressible flow around a cross-flow turbine.



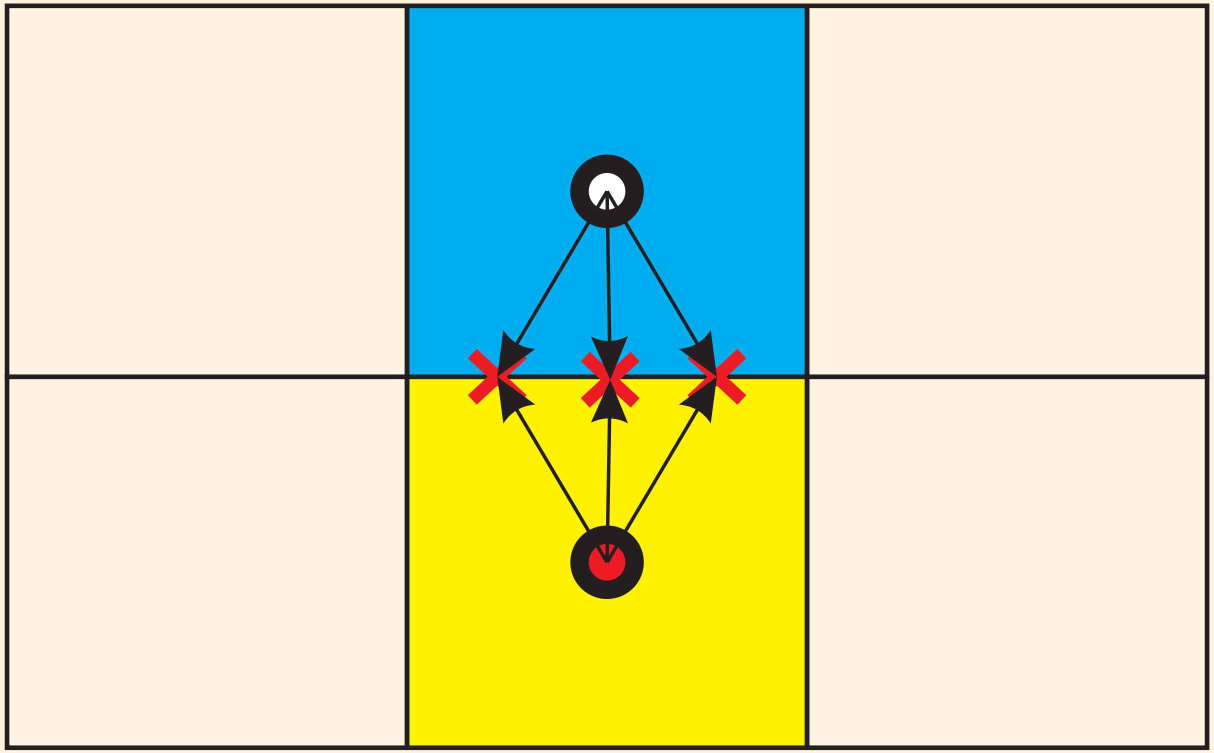
- Incompressible flow around a cross-flow turbine.



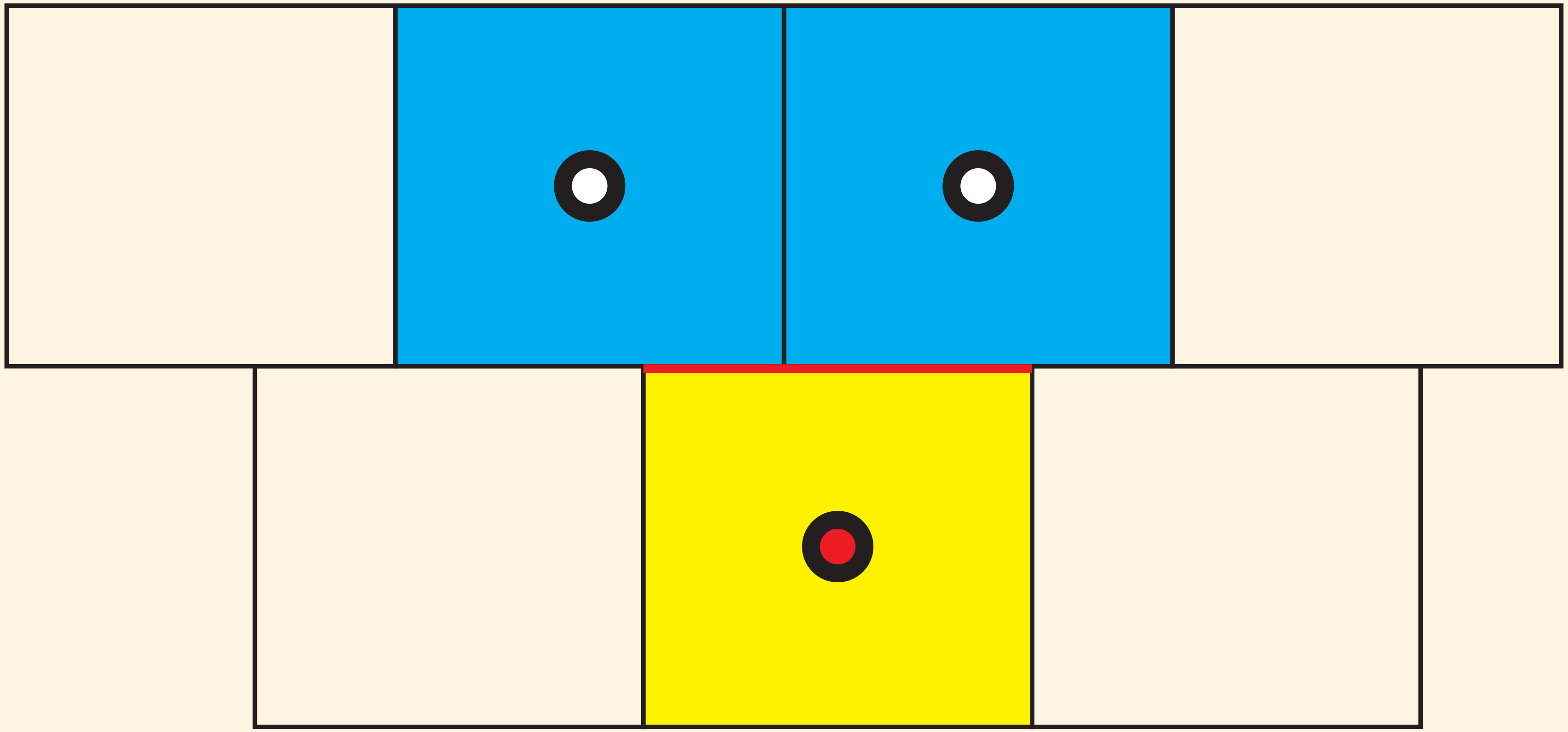
- High-order Sliding Mesh Techniques



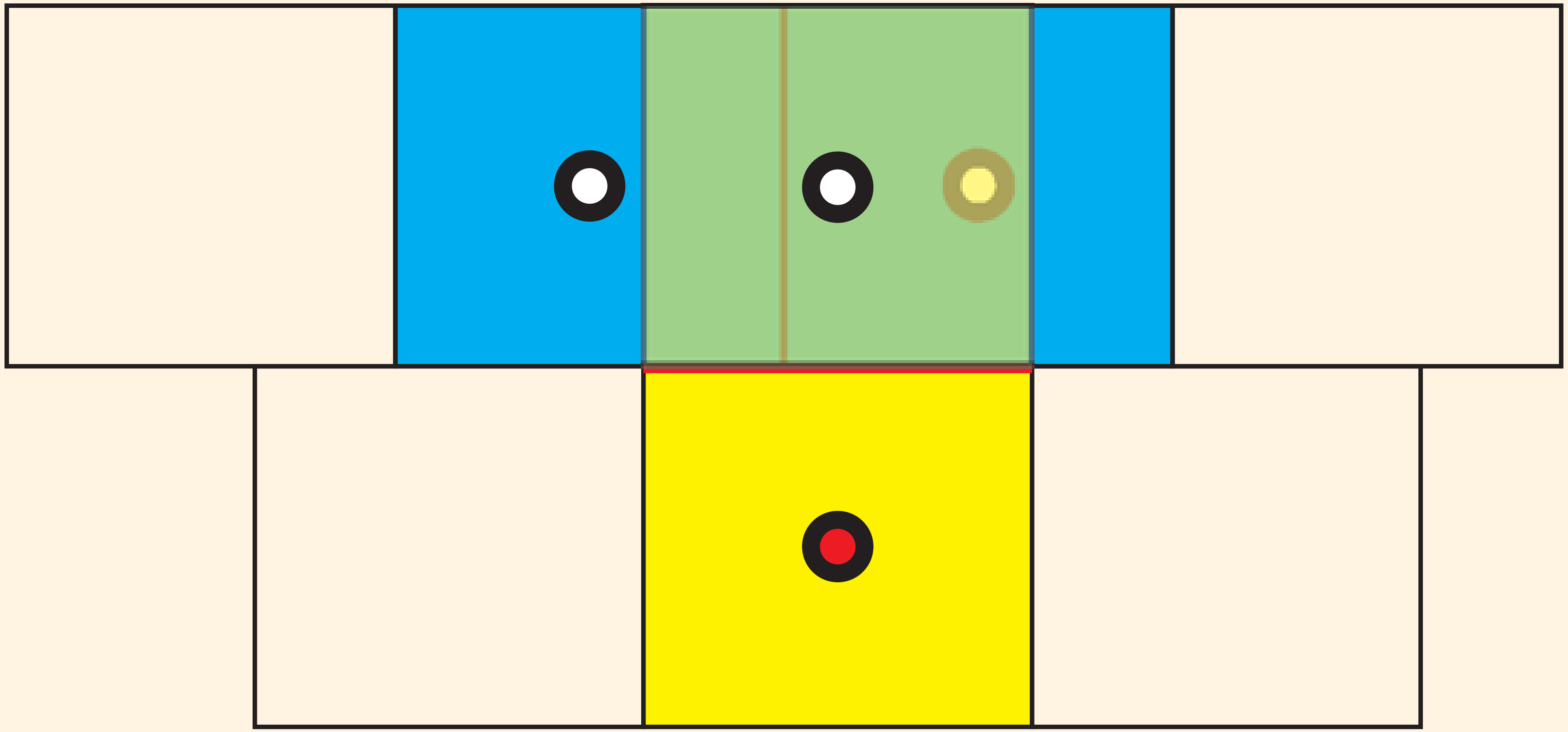
- High-order Sliding Mesh Techniques



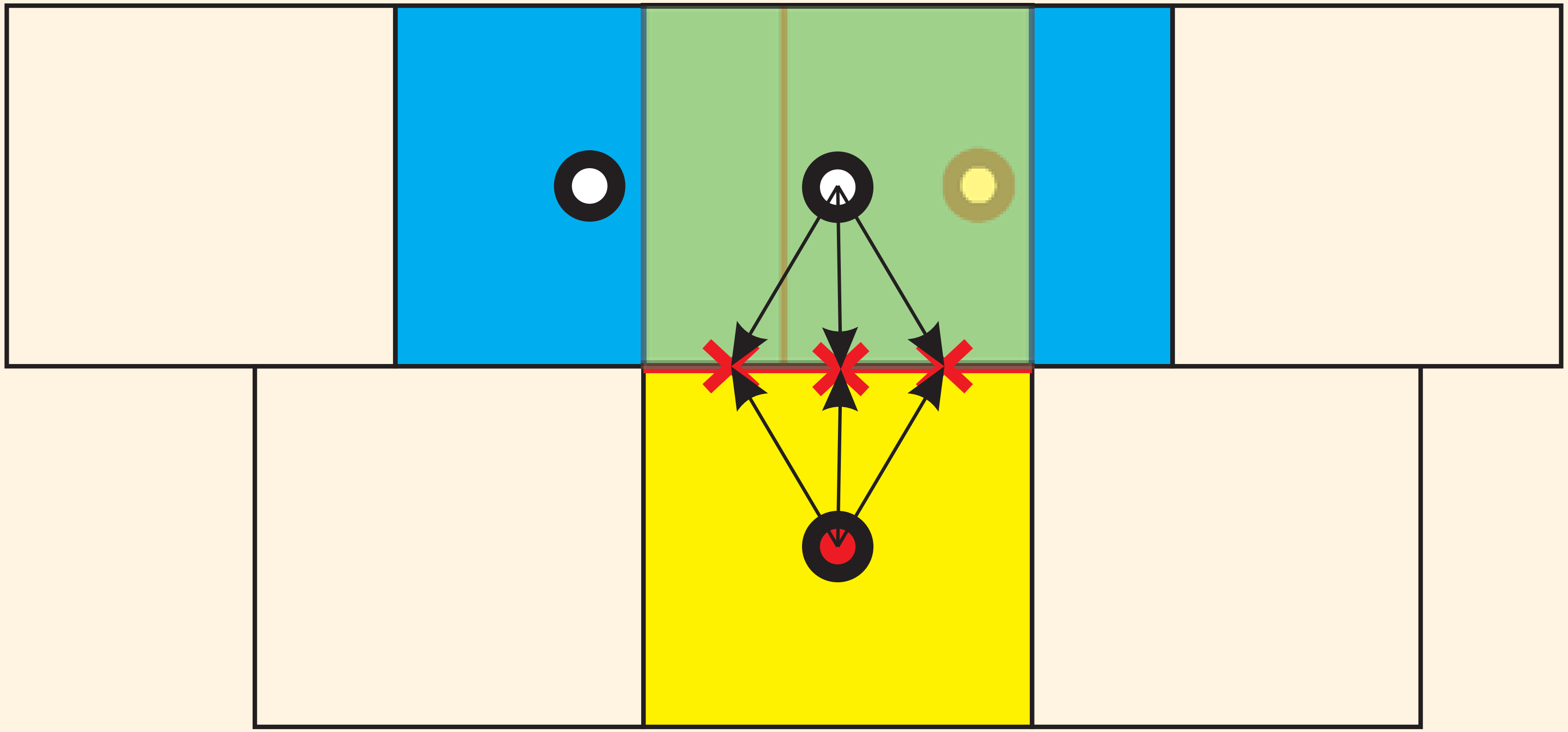
- High-order Sliding Mesh Techniques



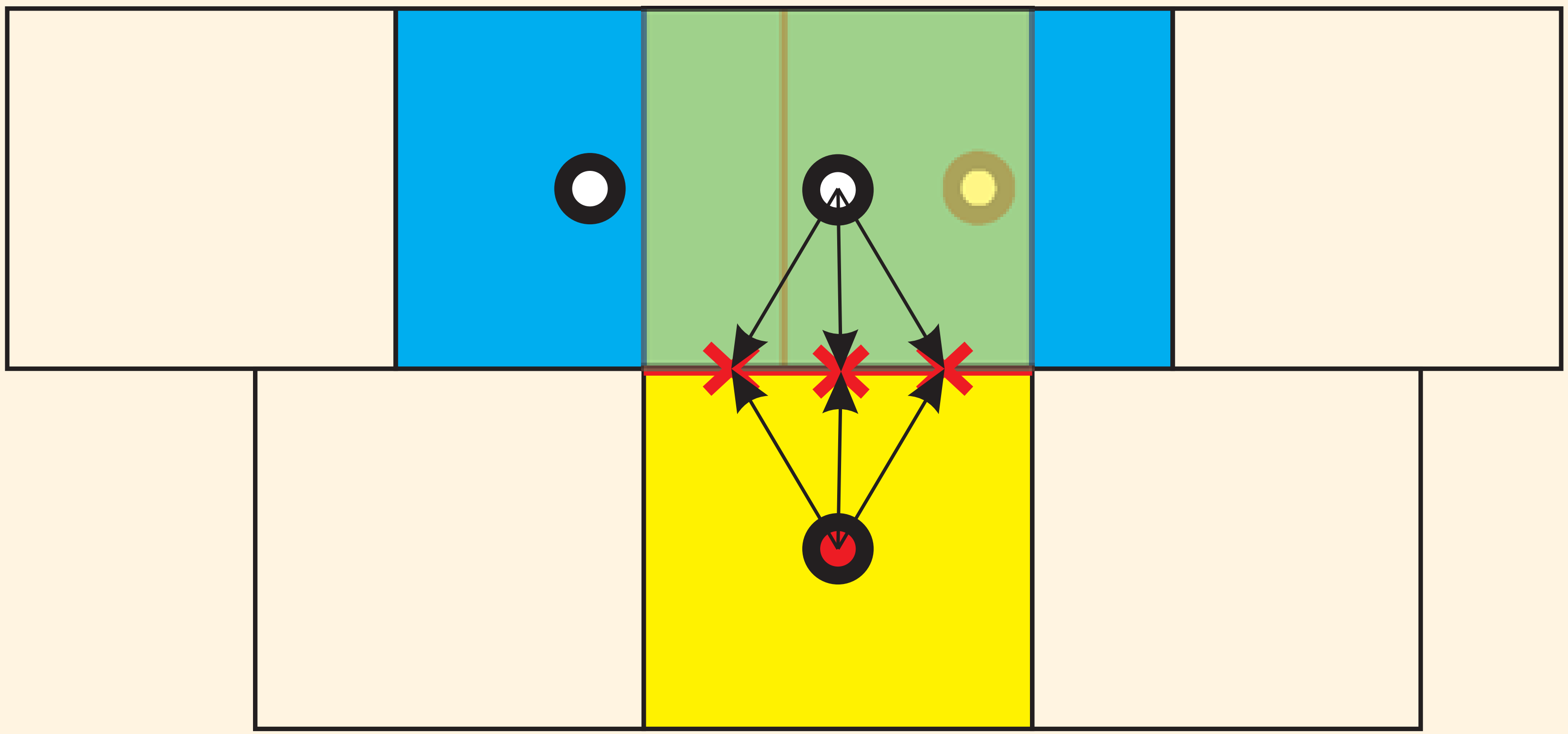
- High-order Sliding Mesh Techniques



- High-order Sliding Mesh Techniques



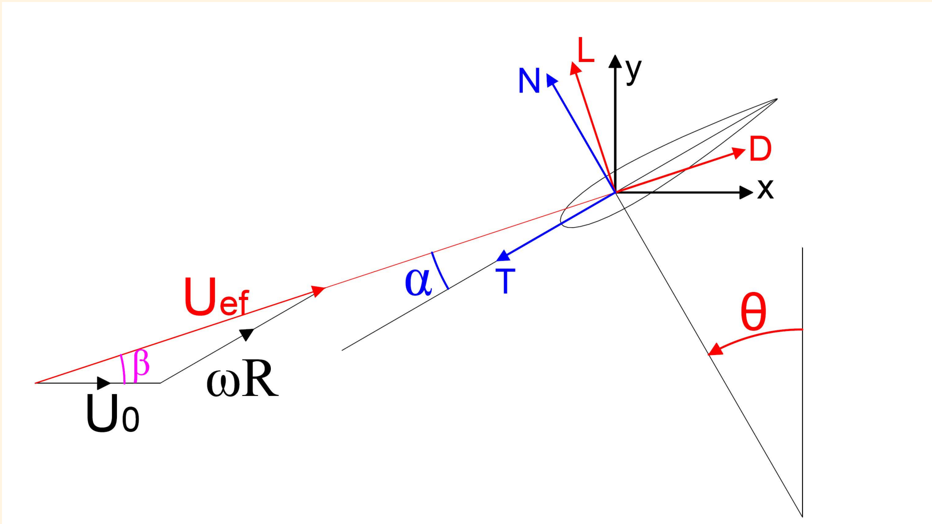
- High-order Sliding Mesh Techniques



It will be presented tomorrow

Numerical Examples

- Incompressible flow around a cross-flow turbine.



$$\vec{f} = \left\{ \begin{array}{c} f_x \\ f_y \end{array} \right\} = \oint (p\vec{n} - \nu(\nabla\vec{U} \cdot \vec{n}))d\Gamma$$

$$f_N = f_y \cos\theta - f_x \sin\theta \quad f_T = -f_x \cos\theta - f_y \sin\theta$$

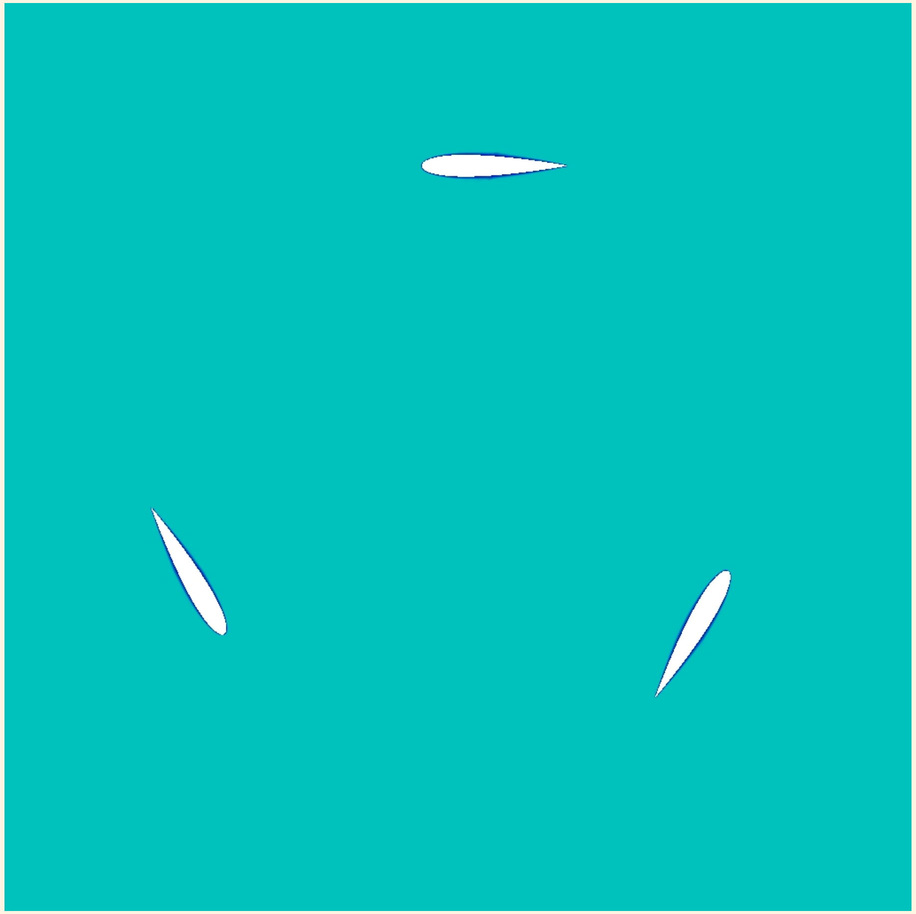
Fluid-Structure Interaction (FSI)

- Flow driven approach $\rightarrow$ ω given by the fluid

$$\omega^{n+1} = \omega^{n} + \frac{(T - M)\ \Delta t}{J}$$

- T $\rightarrow$ Torque
- M $\rightarrow$ Loading Moment
- Δt $\rightarrow$ Time step
- J $\rightarrow$ Mass moment of inertia

- Fluid-Structure Interaction (FSI)



Conclusions

- Introduction
- The FV-MLS method
- A high-order formulation for incompressible flows
- High-order Fluid-Structure-Interaction techniques
- Conclusions

Conclusions

- We have a proposed a new higher-order accurate FV formulation for the numerical solution of incompressible fluid flows on unstructured meshes.
- We have modified the usual linear formulation of MIM to introduce higher-order approximations using MLS.
- The proposed methodology obtains excellent results.
- This methodology can be easily included in existing finite volume codes which represents an additional advantage.

HIGHER-ORDER FV-MLS METHOD FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

Luis Ramírez

email: lramirez@udc.es

Thank you

—

Acknowledgments

- This work has been partially supported by:
  - The *Ministerio de Educación y Ciencia* of the Spanish Government,
  - *Dirección Xeral de I+D* of the *Consellería de Innovación, Industria e Comercio* of the *Xunta de Galicia*,
  - the *Universidade da Coruña (UDC)*, and
  - the *Group of Numerical Methods in Engineering - GMNI*

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

UNIÓN EUROPEA
FONDO EUROPEO DE DESENVOLVEMENTO REXIONAL
"Unha maneira de facer Europa"

XUNTA DE GALICIA
CONSELLERÍA DE ECONOMÍA E INDUSTRIA

Some FV-MLS references

- L. Cueto-Felgueroso, I. Colominas, X. Nogueira, F. Navarrina, and M. Casteleiro, *Finite-volume solvers and moving least-squares approximations for the compressible Navier-Stokes equations on unstructured grids*, CMAME, 2007
- X. Nogueira, I. Colominas, L. Cueto-Felgueroso, and S. Khelladi, *On the simulation of wave propagation with a higher-order finite volume scheme based on reproducing kernel methods*, CMAME, 2010
- X. Nogueira, L. Cueto-Felgueroso, I. Colominas, F. Navarrina, and M. Casteleiro, *A new shock-capturing technique based on moving least squares for higher-order numerical schemes on unstructured grids*, CMAME, 2010
- X. Nogueira, L. Cueto-Felgueroso, I. Colominas, H.Gómez,*Implicit Large Eddy Simulation of non-wall-bounded turbulent flows based on the multiscale properties of a high-order finite volume method*, CMAME, 2010
- X. Nogueira, S. Khelladi, I. Colominas, L. Cueto-Felgueroso, J. París, and H. Gómez, *High-resolution finite volume methods on unstructured grids for turbulence and aeroacoustics*, ARCME, 2011
- S. Khelladi, X. Nogueira, F. Bakir, and I. Colominas, *Toward a higher-order unsteady finite volume solver based on reproducing kernel particle method*, CMAME, 2011.
- J.C. Chassaing, S. Khelladi, and X.Nogueira, *Accuracy assesment of a high-order moving least squares finite volume method for compressible flows*, C&F, 2013
- L. Ramirez, X. Nogueira, S. Khelladi, J.C. Chassaing, and I. Colominas, *A new higher-order finite volume method based on moving least squares for the resolution of the incompressible Navier-Stokes equations on unstructured grids*, CMAME, 2014