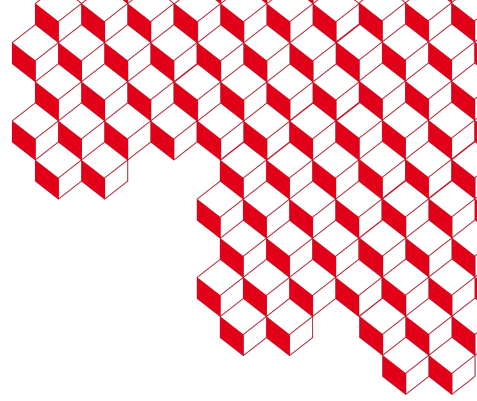


PolyMAC: A Comparison

Staggered Finite Volume Methods on General Meshes

GdR MaNu, 24 Octobre 2023

P.-L. Bacq (pierre-loic.bacq@cea.fr), A. Gerschenfeld, M. Ndjinga
UNIVERSITÉ PARIS-SACLAY, CEA SACLAY

Outline

Outline

Motivation

- Development of codes for nuclear safety
- Challenging: conservation mass at the discrete level, treatment of spurious modes, precision at low Mach numbers,...
- MAC scheme does it all! Unfortunately, only on Cartesian meshes $\Rightarrow$ limited in practice.

Objective: develop a generalization of the MAC scheme on the general meshes - called PolyMAC

PolyMAC

- PolyMAC: **family** of numerical schemes for multiphase compressible flows on **general** meshes (hexahedra, prisms, tetrahedra,...)
- Currently, three versions implemented in TRUST platform $=>$ PolyMAC I, II and III.
- Simplification: incompressible NS equations.

PolyMAC

- PolyMAC: **family** of numerical schemes for multiphase compressible flows on **general** meshes (hexahedra, prisms, tetrahedra,...)
- Currently, three versions implemented in TRUST platform $=>$ PolyMAC I, II and III.
- Simplification: incompressible NS equations.

Objectives:

- Description of the 3 schemes
- Definition of a benchmark (2D and 3D)
- Comparison: order of convergence, numerical computability
- Define guidelines for the user

General principle

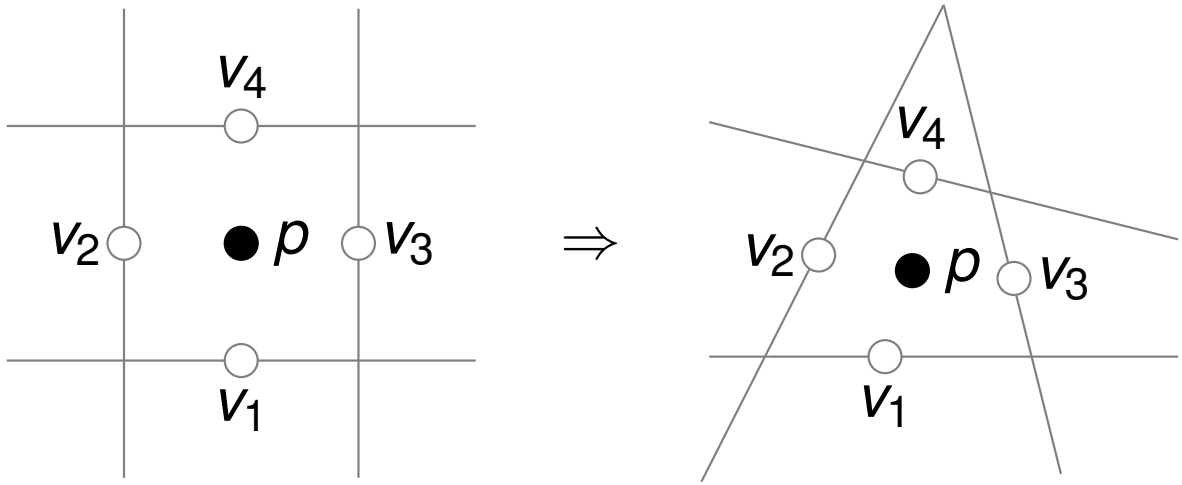
PolyMAC: Finite Volume (FV) **generalization** of the MAC scheme to polyhedral meshes. The main unknowns are then:

- normal component of the **velocity** at the **faces**
- **pressure** at the **cells**

General principle

PolyMAC: Finite Volume (FV) **generalization** of the MAC scheme to polyhedral meshes. The main unknowns are then:

- normal component of the **velocity** at the **faces**
- **pressure** at the **cells**



Incompressible Navier-Stokes equations

Find $\vec{u}$ and p such that

$$\begin{aligned} \partial_t \vec{u} + (\vec{u} \cdot \nabla)\vec{u} - \nu \Delta \vec{u} + \nabla p &= \vec{f} \quad \text{in } \Omega\,, \\ \nabla \cdot \vec{u} &= 0 \quad \text{in } \Omega\,, \end{aligned} \tag{1}$$

- $\vec{u}$: fluid velocity
- p: pressure
- $\nu > 0$: viscosity
- $\Omega \subset \mathbb{R}^d$: unit square in 2D (unit cube in 3D)

Outline

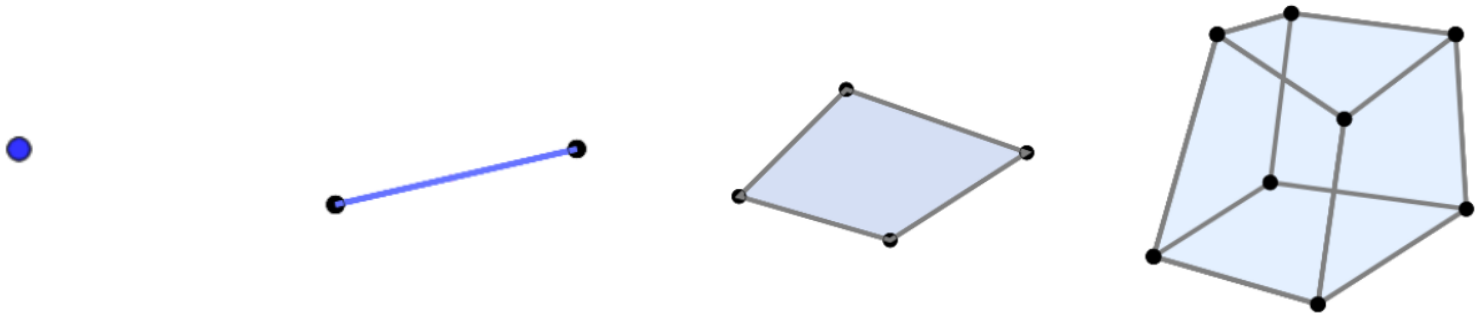
PolyMAC I

Reformulation of Equation (1) with the vorticity:

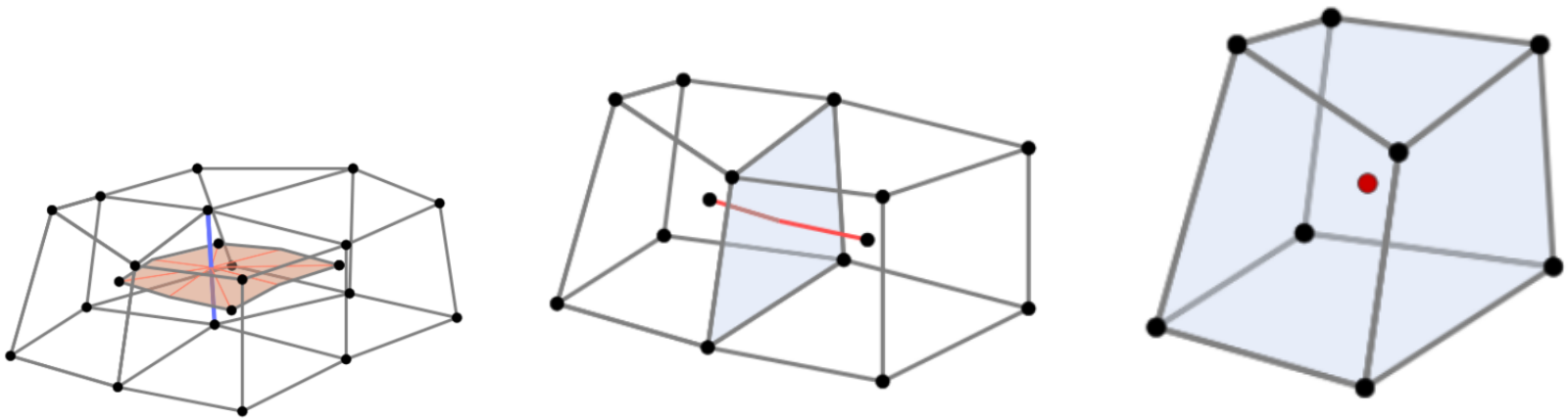
$$\begin{aligned} \text{Momentum:} \quad & \partial_t \vec{u} + \nabla \cdot (\vec{u} \otimes \vec{u}) + \nu \nabla \times \vec{\omega} + \nabla p &&= \vec{f}\,, \\ \text{Vorticity:} \quad & \nabla \times \vec{u} - \vec{\omega} &&= 0\,, \\ \text{Divergence:} \quad & \nabla \cdot \vec{u} &&= 0\,. \end{aligned} \tag{2}$$

- Mimetic finite volumes (Bonelle (2014), Beltman (2018))
- Stability (star-shaped mesh) and discrete conservation law
- Introduction of a dual mesh

PolyMAC Control Volumes



(a) Primal Vertex s (b) Primal Edge a (c) Primal Face f (d) Primal Cell e



(a) Dual Edge $\overline{a}$ (b) Dual Face $\overline{f}$ (c) Dual Cell $\overline{e}$

Mimetic methods for PolyMAC

Use of **mimetic** methods:

- Control volumes chosen so that differential operators have **exact** representation based on integral theorems such as

$$\int_e \nabla \cdot \vec{v} \;=\; \int_{\partial e} \vec{v} \cdot \vec{n}_{\partial e}$$

In terms of discrete unknowns, if the divergence is discretized at the cells and the velocity at the faces:

$$\left[\nabla \cdot \vec{v}\right]_e \;=\; \frac{1}{|e|} \sum_{f \mathrm{v} e} |f| \, [v]_f \;=\; (D\,[v]_F)_e$$

Mimetic methods for PolyMAC

Use of **mimetic** methods:

- Control volumes chosen so that differential operators have **exact** representation based on integral theorems such as

$$\int_e \nabla \cdot \vec{v} \;=\; \int_{\partial e} \vec{v} \cdot \vec{n}_{\partial e}$$

  In terms of discrete unknowns, if the divergence is discretized at the cells and the velocity at the faces:

$$\left[\nabla \cdot \vec{v}\right]_e \;=\; \frac{1}{|e|} \sum_{f \mathrm{v} e} |f| \, [v]_f \;=\; (D\,[v]_F)_e$$

- **Approximation** introduced to link primal to dual mesh
  - the primal edges A to the dual edges $\overline{A}$:

$$[\omega]_{\overline{A}} \approx (M^{(1)}\,[\omega]_A)$$

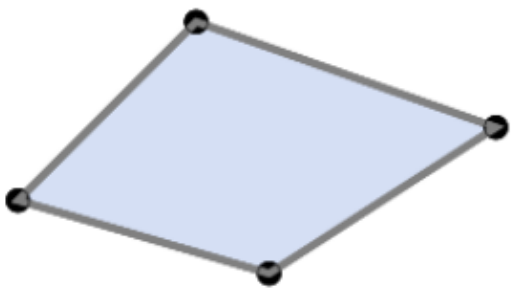
  - the primal faces F to the dual faces $\overline{F}$:

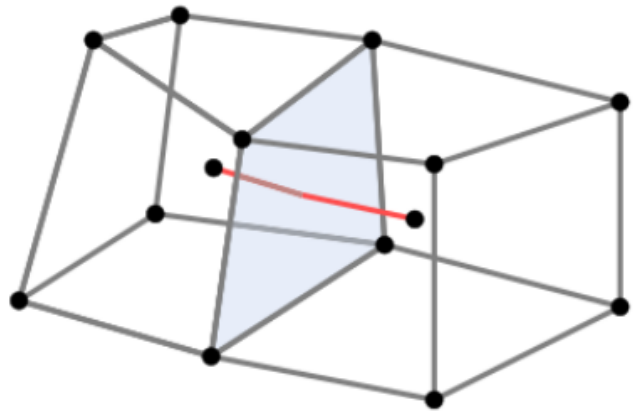
$$[v]_{\overline{F}} \approx (M^{(2)}\,[v]_F)\,.$$

Choice of Control Volumes

For PolyMAC I, the control volumes are

- Velocity at the faces f and momentum at the dual faces $\overline{f}$
  - Time derivative $\left[\partial_t \vec{v}\right]_{\overline{F}} = M^{(2)} \partial_t \left[v\right]_F$
  - Vorticity curl $\left[\nu \nabla \times \vec{\omega}\right]_{\overline{F}} = M^{(2)} R^F \left[\omega\right]_A$
  - Pressure gradient $\left[\nabla p\right]_{\overline{F}} = G \left[p\right]_{\overline{E}}$

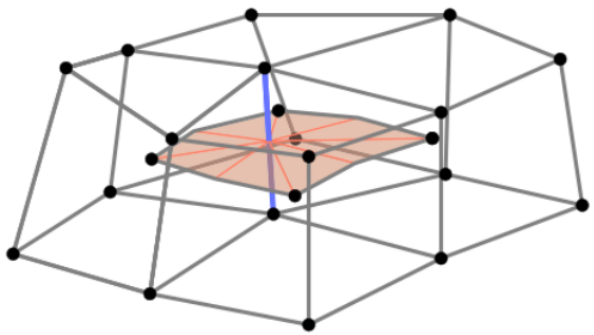




Choice of Control Volumes

For PolyMAC I, the control volumes are

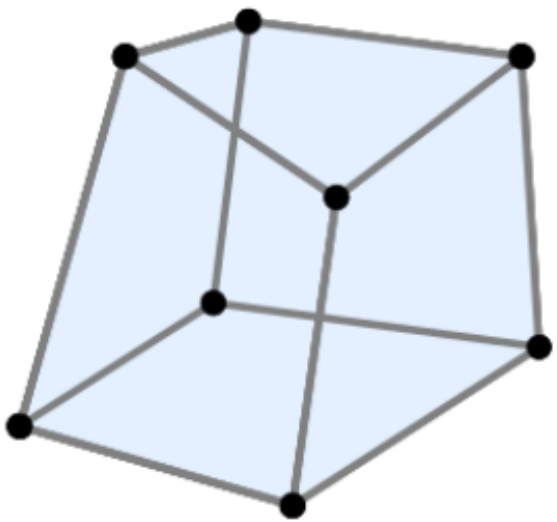
- Velocity at the faces f and momentum at the dual faces $\overline{f}$
- Vorticity at the edges a and its definition at the dual edges $\overline{a}$
  - Velocity curl $\left[\nabla \times \vec{v}\right]_{\overline{A}} = R^A M^{(2)} [v]_F$
  - Vorticity $[\omega]_{\overline{A}} = M^{(1)} [\omega]_A$

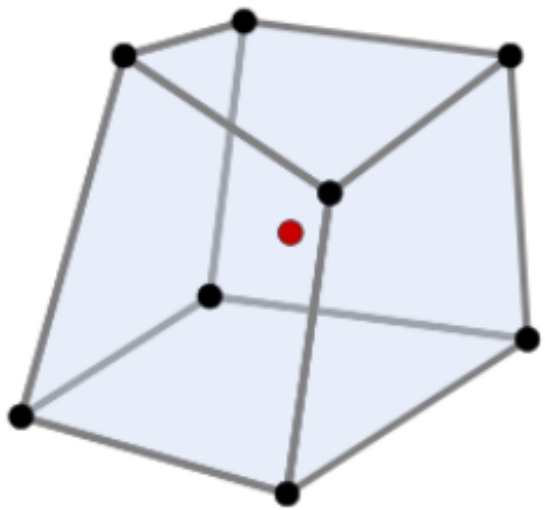


Choice of Control Volumes

For PolyMAC I, the control volumes are

- Velocity at the faces f and momentum at the dual faces $\overline{f}$
- Vorticity at the edges a and its definition at the dual edges $\overline{a}$
- Pressure at the dual cells $\overline{e}$ and continuity at the cells e
  - Divergence $\left[\nabla \cdot \vec{v}\right]_E = D\,[v]_F$





PolyMAC I: Linear System

Resulting linear system:

$$\begin{pmatrix} \frac{M^{(2)}}{\Delta t} + C^F([v]_F^t) & M^{(2)} R^F & G \\ R^A M^{(2)} & -\frac{1}{\nu} M^{(1)} & 0 \\ -D & 0 & 0 \end{pmatrix} \begin{pmatrix} [v]_F^{t+\Delta t} \\ \nu\,[\omega]_A^{t+\Delta t} \\ [p]_E^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} \frac{M^{(2)}}{\Delta t} \mathbf{u}_f^t \\ 0 \\ 0 \end{pmatrix}, \tag{3}$$

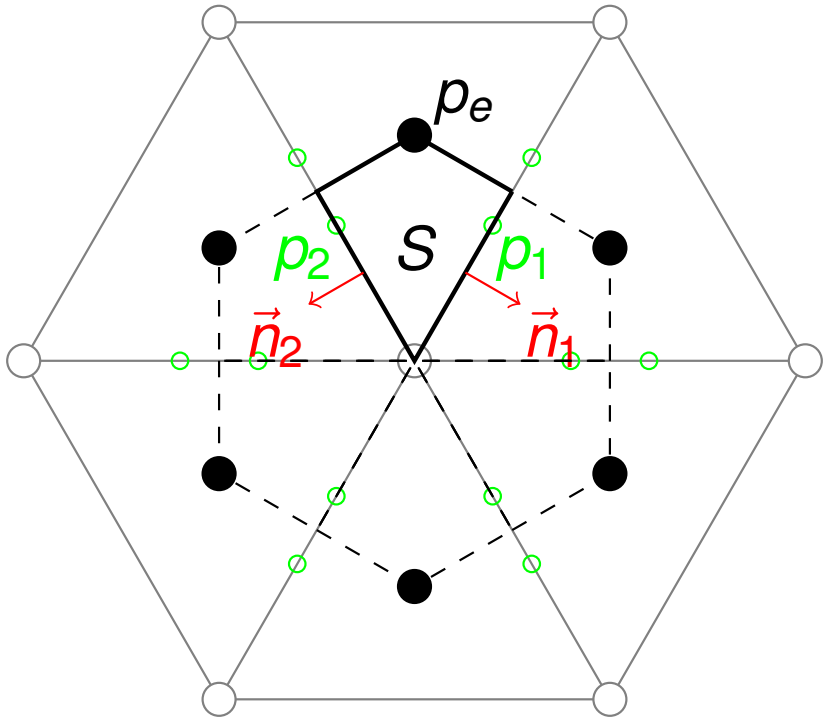
- $C([v]_F^t)$: convection $\Rightarrow$ NS linearised at each time step
- Without convection, symmetric matrix
- **Saddle-point** system
- $M^{(2)}$: not diagonally dominant on deformed meshes

PolyMAC II

PolyMAC II: simplification of the top-left block and avoid saddle-point bottleneck.

- Discretization of the momentum equation at the faces f and the velocity at the faces f $[v]_f$
- Discretization of the continuity equation at the cells e and the pressure at the dual cells $\overline{e}$ $[p]_{\overline{e}}$
- Convection and diffusion terms: complicated
  $\Rightarrow$ auxilary variables: velocity at the elements $\left[\vec{v}\right]_e$.
- Gradient: not an exact operator anymore
  $\Rightarrow$ MPFA for the gradients (Aavatsmark 2002)

MPFA schemes



Approached the gradient in the cell e

$$\nabla p_e \leftarrow G(p) = \frac{1}{|S|}\left((p_1 - p_e)\,\vec{n}_1^T + (p_2 - p_e)\,\vec{n}_2^T\right), \qquad (4)$$

Imposes continuity of the flux

$$\nabla p_e \cdot \vec{n}_2 + \nabla p_{e'} \cdot \vec{n}_2 = 0, \qquad (5)$$

Discretization of the equations

Mass equation:

- As for PolyMAC I: $\left[\nabla \cdot \vec{v}\right]_E = \left(D\left[v\right]_F\right)$

Momentum equation:

- Time derivative: $\left[\partial_t \vec{v}\right]_f = \left(M_f \partial_t \left[v\right]_F\right)_f$
- Pressure gradient: use of the MPFA method $\left[\nabla p\right]_f = \left(G^{MPFA}\left[p\right]_{\overline{E}}\right)_f$
- Diffusion term:
  1. Velocity at the cells $\left[\vec{v}\right]_e$ reconstructed from $\left[v\right]_f$
  2. MPFA used to compute $\left[\nu\nabla\vec{v}\right]_f$ and integral theorems yield

$$\left[\nabla \cdot (\nu\nabla\vec{v})\right]_e = \frac{1}{|e|}\sum_{f \vee e} |f| \left[\nu\nabla\vec{v}\right]_f$$

  3. Interpolate the term to the faces:

$$\left[\nabla \cdot (\nu\nabla\vec{v})\right]_f = \mu\left[\nabla \cdot (\nu\nabla\vec{v})\right]_e + (1-\mu)\left[\nabla \cdot (\nu\nabla\vec{v})\right]_{e'}$$

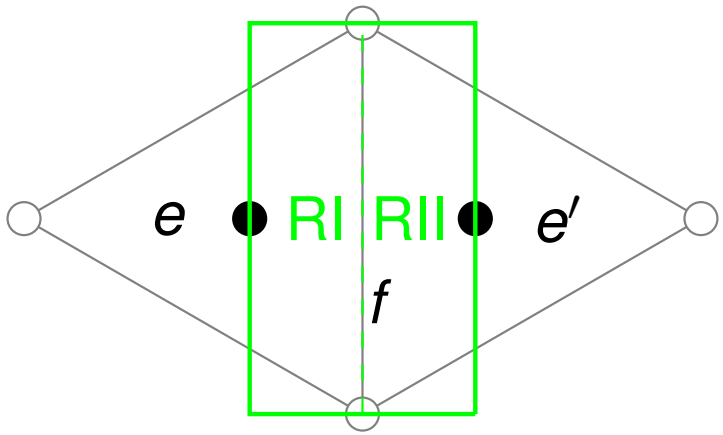
Discretization of the equations (cont'd)

- Convection term:
  1. Velocity at the cells $\left[\vec{v}\right]_e$ reconstructed from $[v]_f$
  2. Apply a convection term to the velocity

$$\left[\nabla \cdot (\vec{v} \otimes \vec{v})\right]_e = \frac{1}{|e|} \sum_{f \text{ v } e} |f| \, [v]_f \, [v]_{cv} \, ,$$

  where $[v]_{cv}$ depends on the chosen convection scheme: upwind or centered.
  3. Finally, the convection term is interpolated at the faces from the neighbouring cell values. The left element will contribute for the ratio $\mu = RII/(RI + RII)$ while the right element will contribute for the ratio $1 - \mu$:

$$\left[\nabla \cdot (\vec{v} \otimes \vec{v})\right]_f = \mu \left[\nabla \cdot (\vec{v} \otimes \vec{v})\right]_e \cdot \vec{n}_f + (1-\mu) \left[\nabla \cdot (\vec{v} \otimes \vec{v})\right]_{e'} \cdot \vec{n}_f \, , \quad (6)$$



Linear system for PolyMAC II

Resulting linear system:

$$\begin{pmatrix} \frac{I}{\Delta t} & C_f(\mathbf{u}_c^t) + D_f(\mathbf{u}_c^t) & G_f^{MPFA} \\ 0 & \frac{I}{\Delta t} + C_c(\mathbf{u}_c^t) + D_c(\mathbf{u}_c^t) & G_c^{MPFA} \\ G^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_f^{t+\Delta t} \\ \mathbf{u}_c^{t+\Delta t} \\ \mathbf{p}_c^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} \frac{I}{\Delta t}\mathbf{u}_f^t \\ \frac{I}{\Delta t}\mathbf{u}_c^t \\ 0 \end{pmatrix} \tag{7}$$

- Saddle point manageable (I instead of M_u)
- High numerical cost on some meshes (tetrahedra,...)
- Not symmetric in the Stokes case

PolyMAC III

Best of both worlds: **simple top left block**, numerical cost **under control**.

- Same formulation as PolyMAC I:

$$\begin{aligned} \text{Momentum:} \quad & \partial_t \vec{u} + \nabla \cdot (\vec{u} \otimes \vec{u}) + \nu \nabla \times \vec{\omega} + \nabla p = \vec{f}\,, \\ \text{Vorticity:} \quad & \nabla \times \vec{u} - \vec{\omega} = 0\,, \\ \text{Divergence:} \quad & \nabla \cdot \vec{u} = 0\,. \end{aligned} \tag{8}$$

- Except momentum at the faces:
  - Velocity at the faces f and momentum at the faces f
  - Vorticity at the edges a and its definition at the dual edges $\overline{a}$
  - Pressure at the dual cells $\overline{e}$ and continuity at the cells e
- Convection similar to PolyMAC II
- Pressure gradient with HFV: auxilary pressure at the faces

Resulting linear system

$$\begin{pmatrix} \frac{I}{\Delta t} + C^F([v]_F^t) & R^F & G_e^{HFV} & G_f^{HFV} \\ R^A M^{(2)} & -\frac{1}{\nu} M^{(1)} & 0 & 0 \\ -D & 0 & 0 & 0 \\ 0 & 0 & P_e & P_f \end{pmatrix} \begin{pmatrix} [v]_F^{t+\Delta t} \\ \nu\,[\omega]_A^{t+\Delta t} \\ [p]_E^{t+\Delta t} \\ [p]_F^{t+\Delta t} \end{pmatrix} \tag{9}$$

$$= \begin{pmatrix} \frac{I}{\Delta t}\,[v]_F^t \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{10}$$

- Saddle point manageable (I instead of M_u)
- Numerical cost reasonable on all meshes
- Not symmetric for Stokes problems

Outline

2D and 3D test problems

In 2D: Bercovier-Engelman test for a right-hand-side of the form:

$$\vec{f} = \begin{pmatrix} f_1(x,y) + (y - \frac{1}{2}) \\ -f_1(y,x) + (x - \frac{1}{2}) \end{pmatrix}, \tag{11}$$

where $f_1(x,y) =$
$256[x^2(x-1)^2(12y-6) + y(y-1)(2y-1)(12x^2 - 12x + 2)]$

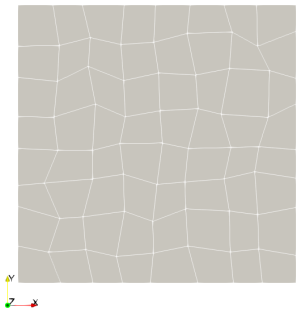
In 3D: rotating Navier-Stokes problem ($\nu = 10^{-2}$, $\mathcal{R}e = 100$)

$$\vec{u} = \begin{pmatrix} y - z \\ z - x \\ x - y \end{pmatrix} \tag{12}$$

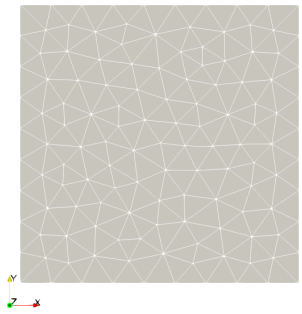
$$p = (x^2 + y^2 + z^2) - xy - xz - yz - \frac{1}{4}$$

(Extensively used in FVCA benchmarks)

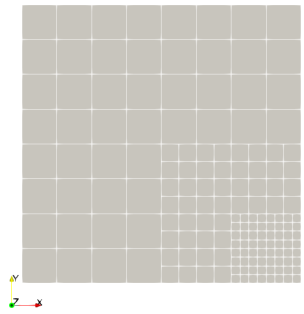
Meshes



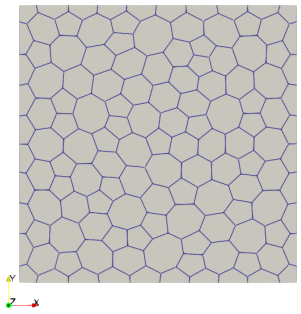
(a) Quadrangles



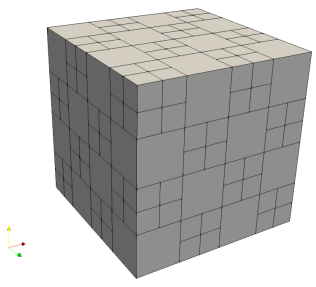
(b) Triangles



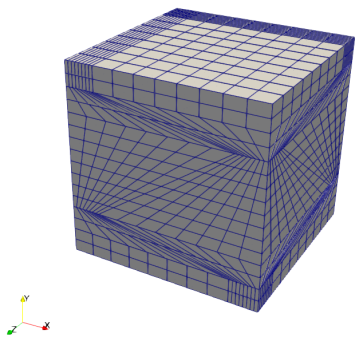
(c) Locally Refined



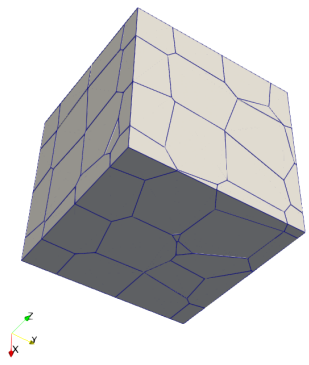
(d) Polygons



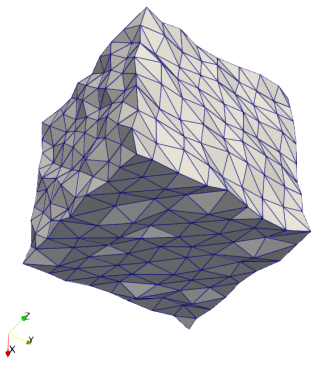
(e) Checkerboard



(f) Kershaw



(g) Voronoi



(h) Random

Figure 3: Representation of some meshes from the benchmark.

Outline

Convergence (in velocity)

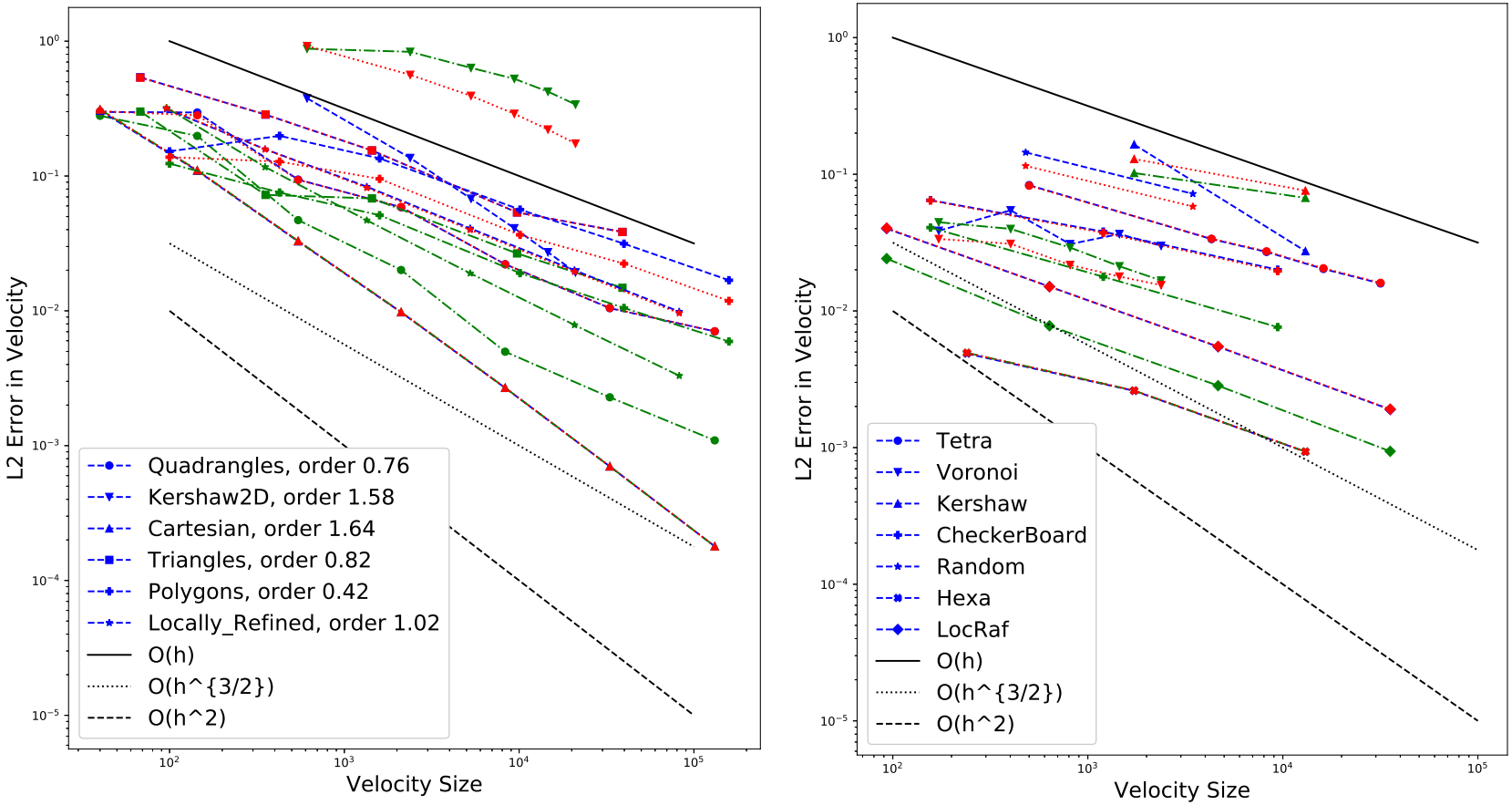


Figure 4: Left: 2D meshes; right: 3D meshes. PolyMAC I is represented in blue, PolyMAC II in green and PolyMAC III in red.

Convergence: 2D

	Velocity		
PolyMAC	I	II	III
`Cartesian`	1.64	**1.66**	**1.66**
`Triangles`	0.82	**1.32**	0.82
`Quadrangles`	0.76	**1.06**	0.78
`Polygons`	0.42	**0.74**	0.50
`Locally Refined 2D`	1.02	**1.50**	1.04
`Kershaw 2D`	**1.58**	0.44	0.86

Table 1: Estimated order of convergence for all three PolyMAC versions on the 2D meshes.

Convergence: 3D

	Velocity		
PolyMAC	I	II	III
Hexahedras	1.08	**1.11**	**1.11**
Locally Refined 3D	1.53	**1.71**	1.53
Kershaw 3D	**2.67**	0.63	0.78
Checker Board	0.84	**1.23**	0.84
Voronoi	0.36	**1.02**	0.90
Tetrahedras	**1.23**	/	1.20
Random	**1.05**	/	**1.05**

Table 2: Estimated order of convergence for all three PolyMAC versions on the 3D meshes.

Sparsity: 2D

	Sparsity Ratio		
PolyMAC	I	II	III
`Cartesian`	9.93	**10.14**	4.65
`Triangles`	8.92	**27.23**	7.95
`Quadrangles`	11.91	**20.48**	9.96
`Polygons`	18.13	**18.42**	14.12
`Locally Refined 2D`	9.95	**10.16**	4.68
`Kershaw 2D`	11.67	**17.58**	9.42

Table 3: Nnz per row on the 2D meshes.

Sparsity: 3D

PolyMAC	Sparsity Ratio I	II	III
Hexahedras	**19.61**	12.86	6.94
Locally Refined 3D	**23.24**	13.93	8.34
Kershaw 3D	42.48	**47.68**	31.52
Checker Board	**91.05**	29.64	54.27
Voronoi	**148.76**	53.58	117.15
Tetrahedras	24.35	**KO**	20.29
Random	72.65	**KO**	63.53

Table 4: Nnz per row on the 3D meshes.

Outline

Comparisons

PolyMAC	Accuracy	Computational cost	Stability
I	+	-	++
II	++	- -	(-)
III	+	++	(-)

(-) Sometimes divergent

- PolyMAC II most **accurate** BUT **heavy**
- PolyMAC III **as accurate** as PolyMAC I but **much sparser** $\Rightarrow$ Default option
- PolyMAC I **always** convergent BUT **saddle point** $\Rightarrow$ Back-up option

P.-L. Bacq, A. Gerschenfeld, M. Ndjinga, PolyMAC: Staggered Finite Volume Methods on General Meshes for Incompressible Navier-Stokes Problems, *FVCA X*, 2023.

Outline

Towards industrial cases

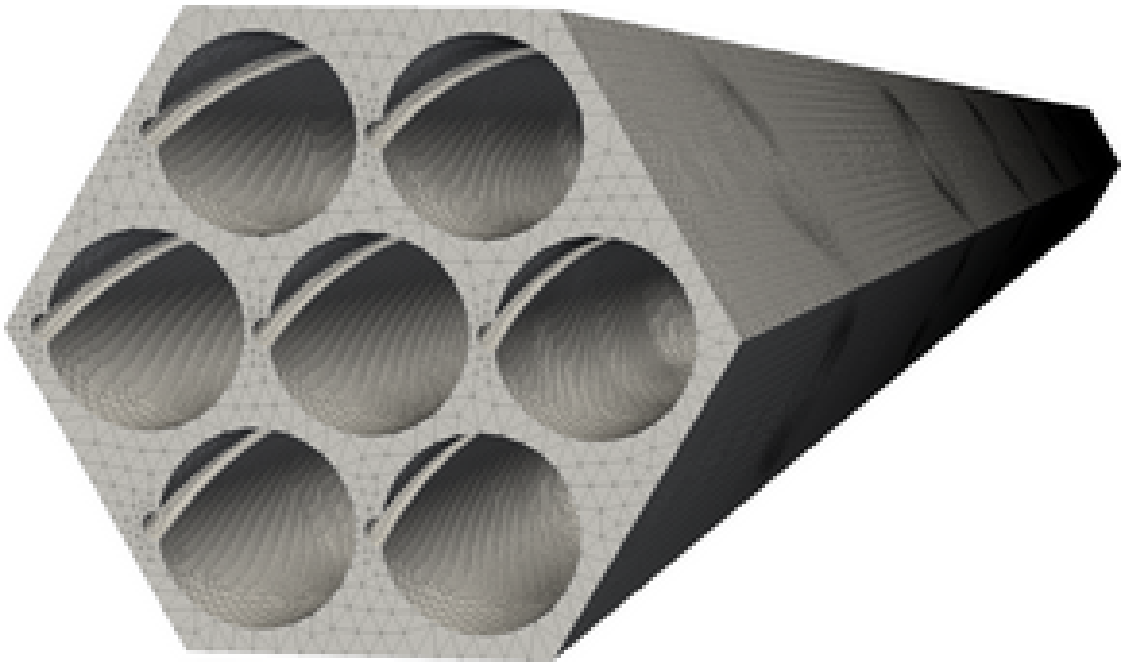


Figure 5: Sodium fast reactor `Assembly` mesh

- More complex than FVCA meshes
- Spacer $=>$ poor cells
- No satisfactory method without efficient linear solver

PolyMAC I system

$$\begin{pmatrix} \frac{M_{\mathbf{u}}}{\Delta t} + C(\mathbf{u}_f^t) & R & G \\ R^T & -\frac{1}{\nu} M_\omega & 0 \\ G^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_f^{t+\Delta t} \\ \nu \omega_e^{t+\Delta t} \\ \mathbf{p}_c^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} \frac{M_{\mathbf{u}}}{\Delta t} \mathbf{u}_f^t \\ 0 \\ 0 \end{pmatrix}$$

- Double saddle point system
- Hard to solve with iterative methods
- Direct solvers restrict the size of the problem

Prediction-correction method

Currently, the solving step is implemented as a correction-prediction method:

1. Prediction: Momentum equation on its own to find first approximation of $\vec{u}$
2. Correction: Poisson equation on pressure to enforce $\nabla \cdot \vec{u} = 0$

Linear system for the prediction:

$$\begin{pmatrix} \frac{M_U}{\Delta t} + F & R_U \\ R_O & M_P \end{pmatrix} \begin{pmatrix} \mathbf{u}^\star \\ \nu\boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} \frac{M_U}{\Delta t}\mathbf{u}^t - B^T\mathbf{p} \\ \mathbf{p} \end{pmatrix} \tag{13}$$

Solved by GMRES preconditioned by a block ILU method *i.e.* both the velocity block and the Schur complement are preconditioned by ILU.

Prediction-correction method

Currently, the solving step is implemented as a correction-prediction method:

1. Prediction: Momentum equation on its own to find first approximation of $\vec{u}$
2. Correction: Poisson equation on pressure to enforce $\nabla \cdot \vec{u} = 0$

Linear system for the correction:

$$\begin{pmatrix} M_U & B^T \\ -B & 0 \end{pmatrix} \begin{pmatrix} \delta \mathbf{u} \\ \Delta t \delta \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -B\mathbf{v}^\star \end{pmatrix} \tag{13}$$

Solved by a **direct solver** $\Rightarrow$ bottleneck of the approach.

Classical Approach for Saddle-Point problems

Block preconditioner for a Krylov subspace method:

$$\begin{pmatrix} M_U & \\ & S \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \tilde{M}_U & \\ & \tilde{S} \end{pmatrix} \tag{14}$$

- $\tilde{M}_U$: approximation of M_U
- $\tilde{S}$: approximation of $S = BM_U^{-1}B^T$ (typically: $BD_U^{-1}B^T$).

Classical Approach for Saddle-Point problems

Block preconditioner for a Krylov subspace method:

$$\begin{pmatrix} M_U & \\ & S \end{pmatrix} \Rightarrow \begin{pmatrix} \tilde{M}_U & \\ & \tilde{S} \end{pmatrix} \tag{14}$$

- $\tilde{M}_U$: approximation of M_U
- $\tilde{S}$: approximation of $S = BM_U^{-1}B^T$ (typically: $BD_U^{-1}B^T$).

Size	Iterations		
	Cartesian	Kershaw 2D	Assembly
1	8	66	NO
2	8	82	NO
3	8	93	NO
4	8	110	/
5	10	140	/
6	10	131	/

An algebraic transformation

Algebraic transformation of the system to give it a more suitable structure:

- Change of variables:

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} I & -D_U^{-1} B^T \\ & I \end{pmatrix} \begin{pmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{p}} \end{pmatrix},$$

where D_U is the diagonal of M_U. The transformed system becomes:

$$\begin{pmatrix} M_U & (I - M_U D_U^{-1}) B^T \\ -B & B D_U^{-1} B^T \end{pmatrix}$$

Advantages:

- More weight on the diagonal blocks (easier for iterative methods)
- Pressure block has a Laplacian-like structure: $\widehat{C} = B D_U^{-1} B^T$

Preconditioning the transformed system

Stokes-like system if reorder the transformed system ($\mathbf{u} \leftrightarrow \mathbf{p}$):

$$\begin{pmatrix} BD_U^{-1}B^T & -B \\ (I - M_U D_U^{-1})B^T & M_U \end{pmatrix}$$

State-of-the-art block preconditioner:

- $\widehat{C} = BD_U^{-1}B^T$: similar to a discrete Laplacian $\Rightarrow$ AMG preconditioner
- M_U: mass matrix *spectrally equivalent* to Schur complement $\Rightarrow 2M_U$

$$\begin{pmatrix} \tilde{\widehat{C}} & \\ & 2\tilde{M}_U \end{pmatrix}$$

Numerical Results

Size	Cartesian		Kershaw 2D		Assembly	
	Class.	Transf.	Class.	Transf.	Class.	Transf.
1	8	**5**	66	**35**	NO	**88**
2	8	**4**	82	**42**	NO	**49**
3	8	**4**	93	**49**	NO	**25**
4	8	**4**	110	**54**	/	/
5	10	**4**	140	**56**	/	/
6	10	**4**	131	**55**	/	/

Outline

Conclusions

- General FV scheme for Navier-Stokes equations
- Implemented for compressible multi-phasic flows
- Key issue identified: iterative solver
- Robust iterative solver for Correction step $\Rightarrow$ **full iterative process**
- Work in progress: Iterative solver for the full 3×3 system
- Implemented in PETSc

Thank you! Any questions?