# Global Jacobian Mortar Algorithms for Multiphase Flow in Porous Media

Ben Ganis*

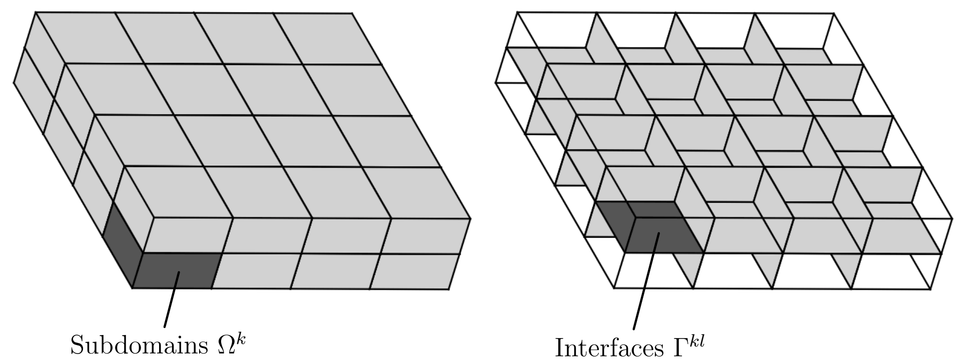
Collaborators: Kundan Kumar*, Gergina Pencheva*, Mary F. Wheeler*, Ivan Yotov**

*Center for Subsurface Modeling, ICES, UT Austin
**University of Pittsburgh

ICES Seminar-Babuska Forum Series
November 7, 2014

# Multiscale Mortar Mixed FEM

- Mortar finite elements are a domain decomposition technique to couple unknowns across:
  - Multiple Scales
  - Multiple Physics
  - Multiple Numerics
  - Multiple Processors




- Note that Domain Decomposition is not the same as “Data Decomposition”.

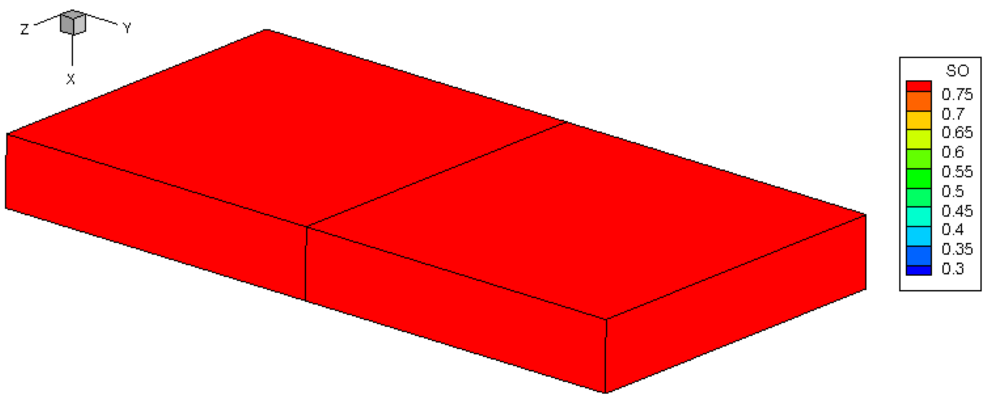
- The “Global Jacobian” algorithms developed in this research seek to have the best of both worlds.

# Models used with mortars

- Mortars have been used with:
  - 1,2,3 phase flows in porous media
  - Linear elastic solid mechanics
  - Porescale network models
  - CG, Mixed, DG methods
  - Bricks, prisms, tetrahedra

- <u>Example</u>:

  Saturation field in two phase flow, with two subdomains.




- Prior to this research, the solution algorithm for nonlinear problems relied on two Newton loops with a forward difference approximation.



Center for Subsurface Modeling

# Selected References on Mortars


Single Phase Mortar Theory

- Glowinski, R., and Wheeler, M.F. 1988. Domain decomposition and mixed finite element methods for elliptic problems. In *1st international symposium on domain decomposition methods for PDEs*.
- Arbogast, T., Cowsar, L.C., Wheeler, M.F. and Yotov, I. 2000. Mixed finite element methods on nonmatching multiblock grids. *SIAM Journal on Numerical Analysis* **37** (4): 1295– 1315.
- Arbogast, T., Pencheva, G., Wheeler, M.F., and Yotov, I. 2007. A multiscale mortar mixed finite element method. *Multiscale Modeling & Simulation* **6** (1): 319–346.

Forward Difference (FD) Algorithms for Nonlinear problems

- Peszynska, M., Wheeler, M.F., and Yotov, I. 2002. Mortar upscaling for multiphase flow in porous media. *Computational Geosciences* **6** (1): 73–100.
- Yotov, I. 2001. A multilevel Newton–Krylov interface solver for multiphysics couplings of flow in porous media. *Numerical Linear Algebra and Applications*, **8** (8): 551–570.

Global Jacobian (GJ) Algorithms for Nonlinear problems

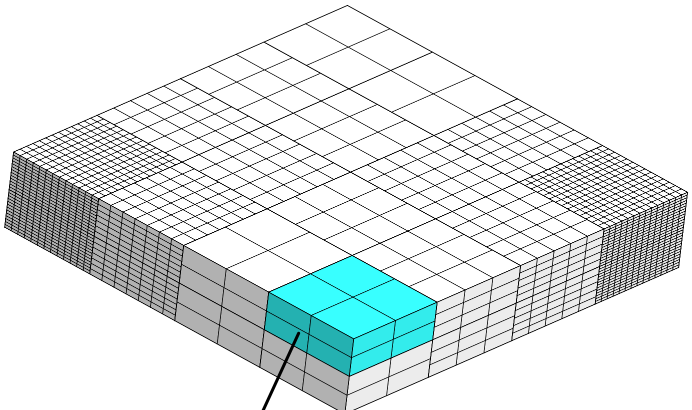
- Ganis, B., Juntunen, M., Pencheva, G., Wheeler, M.F., and Yotov, I. 2014. A global Jacobian method for mortar discretizations of nonlinear porous media flows. *SIAM Journal on Scientific Computation* **36** (2): A522–A542.
- Ganis, B., Kumar, K., Pencheva, G., Wheeler, M.F., and Yotov, I. 2014. A global Jacobian method for mortar discretizations of a fully-implicit two-phase flow model. *Multiscale Modeling & Simulation* **12** (4): 1401–1423.
- Ganis, B., Kumar, K., Pencheva, G., Wheeler, M.F., Yotov, I. A multiscale mortar method and two-stage preconditioner for multiphase flow using a global Jacobian approach. SPE 172990-MS.

# Outline

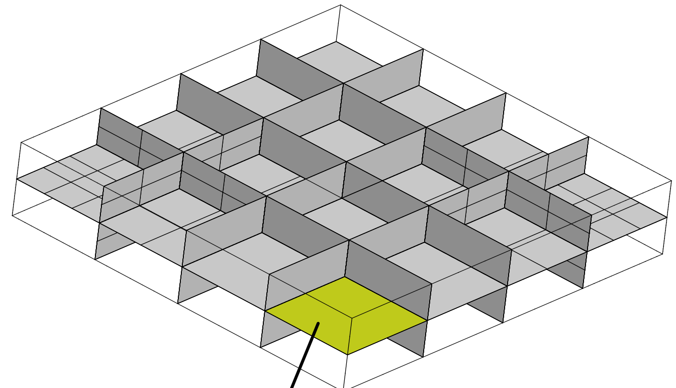
1. Multiscale, Multiphase Problem Setting
2. Fully-implicit two-phase model for flow in porous media
3. Global Jacobian algorithms
   - Schur complements
   - Interface unknowns
   - Upwinding scheme
4. Numerical results
   - Strongly Heterogeneous Case
   - Two Rock Type Case
   - Non-matching Geometry Case
5. Two-Stage Preconditioner and Parallel Results

# Problem Setting

- Non-overlapping domain decomposition on spatial domain



Use mixed finite elements on structured subdomain grids



Use high-order mortars (Lagrange multipliers) on non-matching interfaces

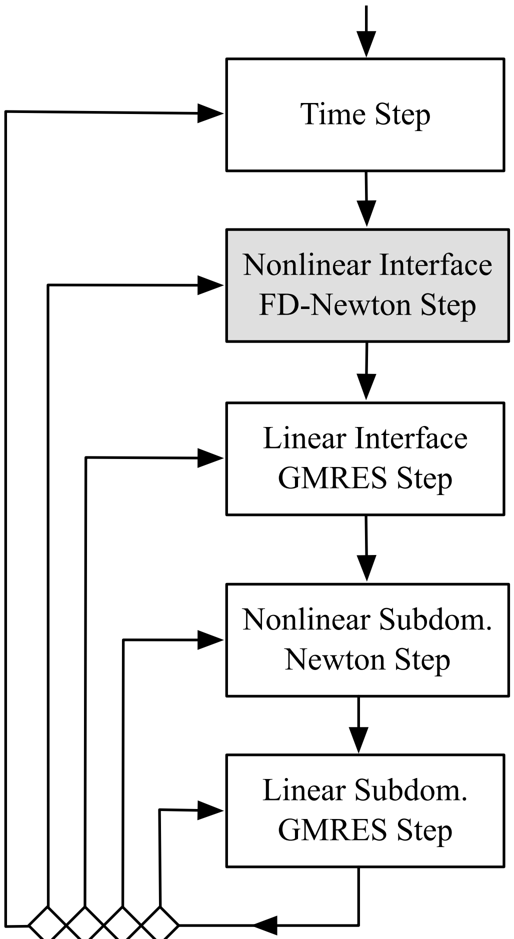
- **Application**: Multiphase flow in porous media
- **Goal**: Develop simple algorithms with parallel scalability
- **Key Idea**: Global linearization
- **Capillarity, gravity, and compressibility.**

# Algorithms for nonlinear mortar problems

- This algorithm uses local linearizations for subdomain and mortar unknowns separately.
  - Two nested Newton-Krylov loops
  - Outer loop formes a numerical Jacobian with a forward difference
  - Requires delicate choice of four tolerances and difference parameter
  - Challenging to precondition outer GMRES
  + Allows multiple physics and multiple time steps

**Prior Method**

**FD Method**



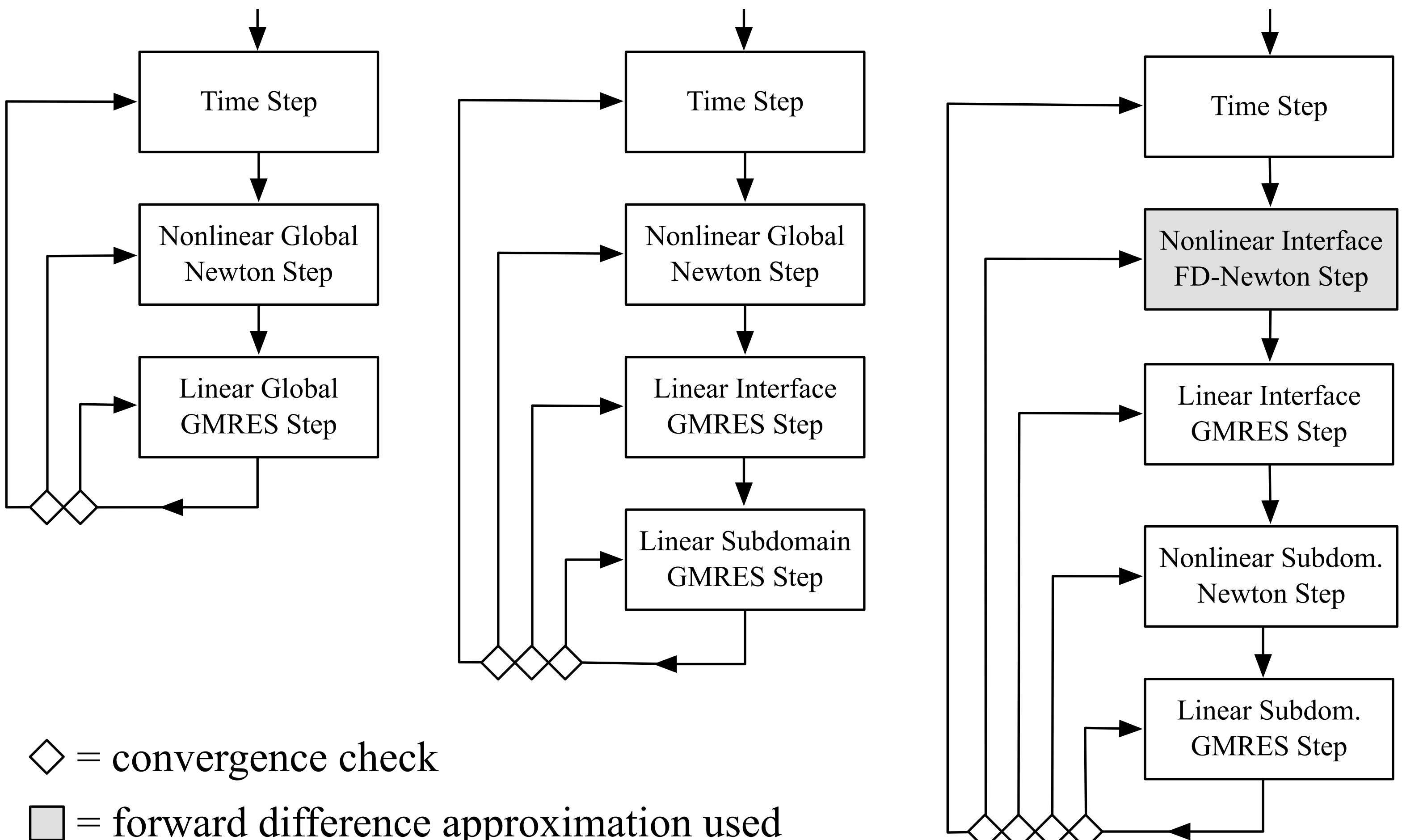

◇ = convergence check

□ = forward difference approximation used

# Algorithms for nonlinear mortar problems

**New Methods** | **Prior Method**

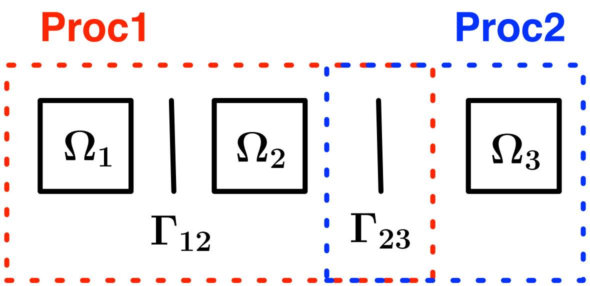
**GJ Method** | **GJS Method** | **FD Method**

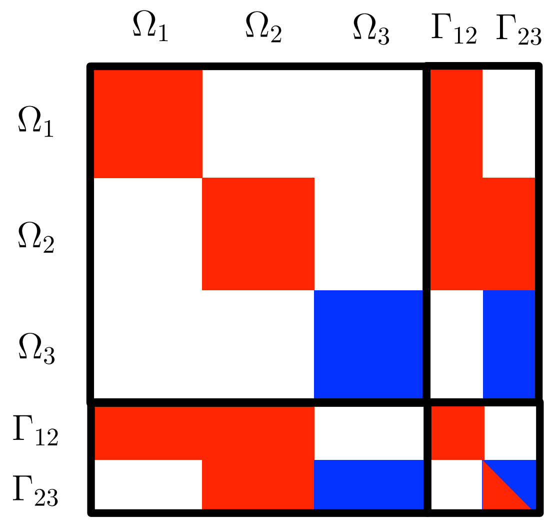



◇ = convergence check

▢ = forward difference approximation used

# Novelty of this work



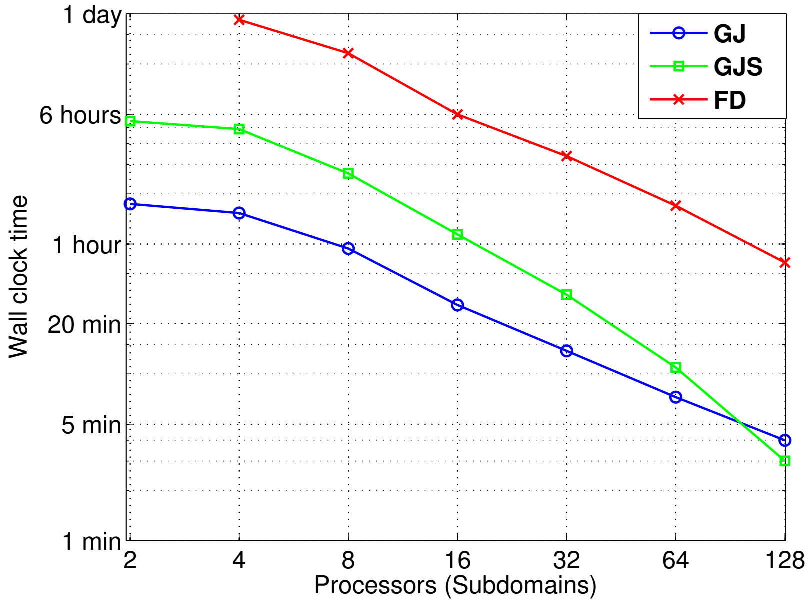

$$= \begin{bmatrix} J_{\Theta\Theta} & J_{\Theta\Lambda} \\ J_{\Lambda\Theta} & J_{\Lambda\Lambda} \end{bmatrix}$$

Global linearization:

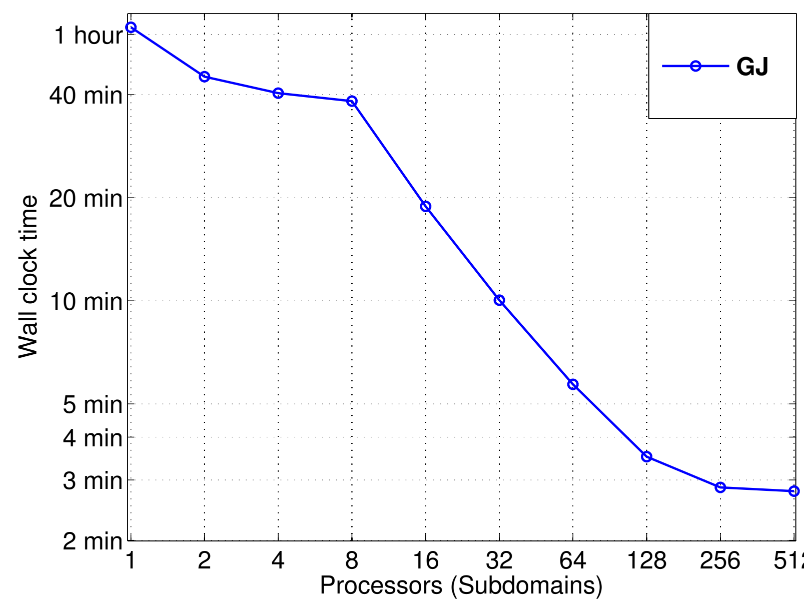
- Augment linear systems to reuse codes.
- Utilize existing preconditioners for multiscale models.
- Simplify algorithms by having fewer nested iterations.
- Demonstrate parallel scaling with strong nonlinearities.
- Improve saturation with careful mobility upwinding.

# Parallel scaling, nonlinear single phase

Homogeneous,
No Preconditioning




Heterogeneous,
AMG+ILU Preconditioner




Strong scaling, $O(10^6)$ elements

[2] B. Ganis, M. Juntunen, G. Pencheva, M.F. Wheeler, I. Yotov. A global Jacobian method for mortar discretizations of nonlinear porous media flows. *SIAM Journal on Scientific Computation*, Vol. 36, No. 2, (2014) pp. A522-A542.

# Two Phase Model

Mass Balance: $$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha) + \nabla \cdot \boldsymbol{u}_\alpha = q_\alpha \quad \text{in } \Omega^k \times (0,T]$$

Auxiliary Velocity: $$\widetilde{\boldsymbol{u}}_\alpha = -K(\nabla p_\alpha - \rho_\alpha \boldsymbol{g}) \quad \text{in } \Omega^k \times (0,T]$$

Darcy Law: $$\boldsymbol{u}_\alpha = \frac{k_{r\alpha}\rho_\alpha}{\mu_\alpha} \widetilde{\boldsymbol{u}}_\alpha \quad \text{in } \Omega^k \times (0,T]$$

Initial condition: $$p_\alpha = p_{\alpha,0} \quad \text{at } \Omega \times \{t=0\},$$

Boundary condition: $$\boldsymbol{u} \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega \times (0,T]$$

Lagrange multiplier: $$p_\alpha = p_\alpha^\Gamma(\lambda_1, \lambda_2) \quad \text{on } \Gamma \times (0,T],$$

Flux continuity: $$\boldsymbol{u}_\alpha^k \cdot \boldsymbol{n}^k + \boldsymbol{u}_\alpha^l \cdot \boldsymbol{n}^l = 0 \quad \text{on } \Gamma^{kl} \times (0,T]$$

Saturation constraint: $$s_\mathrm{w} + s_\mathrm{o} = 1$$

Capillary pressure: $$p_\mathrm{c}(s_\mathrm{w}) = p_\mathrm{o} - p_\mathrm{w}$$

Slightly compressible density: $$\rho_\alpha(p_\alpha) = \rho_\alpha^{\mathrm{ref}} \mathrm{e}^{c_\alpha p_\alpha}.$$

# Finite element discretization

Primary Unknowns: $(p_{\mathrm{o}}, n_{\mathrm{o}})$

Phase Velocities: $(\widetilde{\boldsymbol{u}}_{\mathrm{o}}, \widetilde{\boldsymbol{u}}_{w}, \boldsymbol{u}_{\mathrm{o}}, \boldsymbol{u}_{\mathrm{w}})$

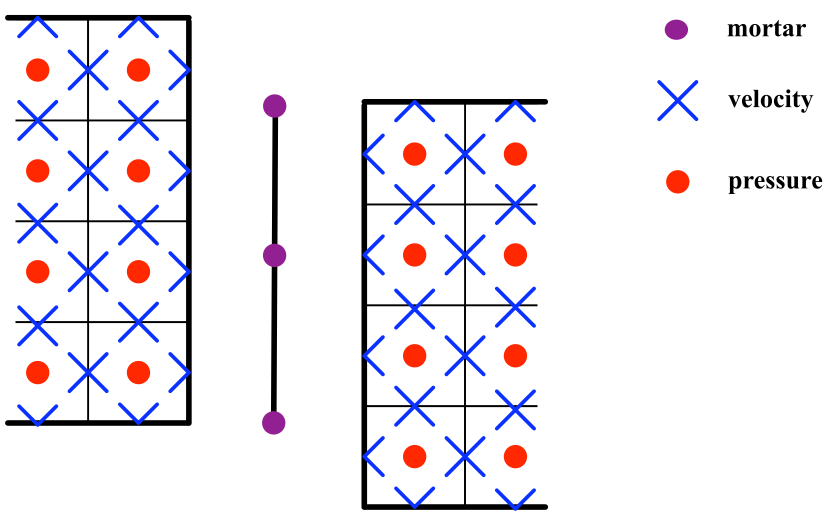
Lagrange Multipliers: $(\lambda_1, \lambda_2)$

Lowest Order Raviart-Thomas (RT0) mixed finite elements with mortars

Velocity, Pressure, Mortar Spaces:

$$\mathrm{V}_h = \bigoplus_{k=1}^{N_\Omega} \mathrm{V}_h^k, \quad \mathrm{W}_h = \bigoplus_{k=1}^{N_\Omega} \mathrm{W}_h^k,$$

$$\mathrm{M}_H = \bigoplus_{k=1}^{N_\Omega} \mathrm{M}_H^{kl}.$$




Time discretization:

$$0 = t^0 < t^1 < \cdots < t^{N_T} = T, \text{ with } \delta t^n = t^n - t^{n-1}.$$

# Fully discrete system

Expanded multiscale mortar method for fully-implicit two-phase flow:

Phase concentration: $n_\alpha = \rho_\alpha s_\alpha$

**Phase mobility:** $m_\alpha = \dfrac{k_{r\alpha}\rho_\alpha}{\mu_\alpha}$

$$A_\alpha^k = \int_{\Omega^k} \boldsymbol{u}_\alpha^k \cdot \boldsymbol{v}\, dx - \int_{\Omega^k} m_\alpha \tilde{\boldsymbol{u}}_\alpha^k \cdot \boldsymbol{v}\, dx = 0,$$

$$D_\alpha^k = \int_{\Omega^k} K^{-1}\tilde{\boldsymbol{u}}_\alpha^k \cdot \boldsymbol{v}\, dx - \int_{\Omega^k} p_\alpha^k \nabla \cdot \boldsymbol{v}\, dx - \int_{\Omega^k} \rho_\alpha \boldsymbol{g} \cdot \boldsymbol{v}\, dx + \sum_{l=1, l\neq k}^{N_\Omega} \int_{\Gamma^{kl}} p_\alpha^\Gamma \boldsymbol{v} \cdot \boldsymbol{n}\, d\sigma = 0,$$

$$B_\alpha^k = \int_{\Omega^k} \frac{\phi n_\alpha^k - \phi n_\alpha^{n-1}}{\delta t} w\, dx + \int_{\Omega^k} \nabla \cdot \boldsymbol{u}_\alpha^k w\, dx - \int_{\Omega^k} q_\alpha w\, dx = 0,$$

$$H_\alpha = \int_{\Gamma^{kl}} \left(\boldsymbol{u}_\alpha^k \cdot \boldsymbol{n}_k + \boldsymbol{u}_\alpha^l \cdot \boldsymbol{n}_l\right) \mu\, d\sigma = 0.$$

**Flux continuity equation**

# Forming Residual Equations

- Express 8 unknowns as **linear combinations** of finite element basis functions, insert into discrete form.

$$p_o^k = \sum_{i=1}^{N_p^k} P_{o,i}^k w_i^k$$

$$A_\alpha^k = \int_{\Omega^k} \boldsymbol{u}_\alpha^k \cdot \boldsymbol{v}\, dx - \int_{\Omega^k} m_\alpha \widetilde{\boldsymbol{u}}_\alpha^k \cdot \boldsymbol{v}\, dx = 0,$$

$$D_\alpha^k = \int_{\Omega^k} K^{-1} \widetilde{\boldsymbol{u}}_\alpha^k \cdot \boldsymbol{v}\, dx - \int_{\Omega^k} p_\alpha^k \nabla \cdot \boldsymbol{v}\, dx - \int_{\Omega^k} \rho_\alpha \boldsymbol{g} \cdot \boldsymbol{v}\, dx + \sum_{l=1, l\neq k}^{N_\Omega} \int_{\Gamma^{kl}} p_\alpha^\Gamma \boldsymbol{v} \cdot \boldsymbol{n}\, d\sigma = 0,$$

$$B_\alpha^k = \int_{\Omega^k} \frac{\phi n_\alpha^k - \phi n_\alpha^{n-1}}{\delta t} w\, dx + \int_{\Omega^k} \nabla \cdot \boldsymbol{u}_\alpha^k w\, dx - \int_{\Omega^k} q_\alpha w\, dx = 0,$$

$$H_\alpha = \int_{\Gamma^{kl}} \left( \boldsymbol{u}_\alpha^k \cdot \boldsymbol{n}_k + \boldsymbol{u}_\alpha^l \cdot \boldsymbol{n}_l \right) \mu\, d\sigma = 0.$$

- Obtain a nonlinear system for the global coefficient vectors:

$$\widetilde{U}_o, \widetilde{U}_w, U_o, U_w \in \mathbb{R}^{N_{\boldsymbol{u}}} \qquad P_o, N_o \in \mathbb{R}^{N_p} \qquad \Lambda_1, \Lambda_2 \in \mathbb{R}^{N_\lambda}$$

$$N_{\boldsymbol{u}} = \sum_{i=1}^{N_\Omega} N_{\boldsymbol{u}}^k \qquad N_p = \sum_{i=1}^{N_\Omega} N_p^k \qquad N_\lambda = \sum_{1 \leq k < l \leq N_\Omega} N_\lambda^{kl}$$

# Global nonlinear system

- Express all variables in terms of primary unknowns
- Nonlinear system of 8 equations in 8 unknowns

$$
\begin{aligned}
A_{\mathrm{o}}(\widetilde{U}_{\mathrm{o}}, U_{\mathrm{o}}, P_{\mathrm{o}}, N_{\mathrm{o}}) &= 0 \\
A_{\mathrm{w}}(\widetilde{U}_{\mathrm{w}}, U_{\mathrm{w}}, P_{\mathrm{o}}, N_{\mathrm{o}}) &= 0
\end{aligned}
\quad \Bigg\} \quad \text{Aux. Velocity}
$$

$$
\begin{aligned}
D_{\mathrm{o}}(\widetilde{U}_{\mathrm{o}}, P_{\mathrm{o}}, \Lambda_1, \Lambda_2) &= 0 \\
D_{\mathrm{w}}(\widetilde{U}_{\mathrm{w}}, P_{\mathrm{o}}, N_{\mathrm{o}}, \Lambda_1, \Lambda_2) &= 0
\end{aligned}
\quad \Bigg\} \quad \text{Darcy Velocity}
$$

$$
\begin{aligned}
B_{\mathrm{o}}(U_o, N_{\mathrm{o}}) &= 0 \\
B_{\mathrm{w}}(U_w, P_{\mathrm{o}}, N_{\mathrm{o}}) &= 0
\end{aligned}
\quad \Bigg\} \quad \text{Mass Balance}
$$

$$
\begin{aligned}
H_{\mathrm{o}}(U_{\mathrm{o}}) &= 0 \\
H_{\mathrm{w}}(U_{\mathrm{w}}) &= 0
\end{aligned}
\quad \Bigg\} \quad \text{Flux Continuity}
$$

# Forming Jacobian entries

- Compute partial derivatives of each residual equation with respect to each type of unknown.

$$(A_1^k)_{ji} = \frac{\partial A_{o,j}^k}{\partial \widetilde{U}_{o,i}} = -\left(m_o \boldsymbol{v}_i, \boldsymbol{v}_j\right)_k ,$$

$$(A_2^k)_{ji} = \frac{\partial A_{o,j}^k}{\partial U_{o,i}} = \left(\boldsymbol{v}_i, \boldsymbol{v}_j\right)_k ,$$

$$(\widehat{A}_3^k)_{ji} = \frac{\partial A_{o,j}^k}{\partial P_{o,i}} = -\left(\left(\frac{c_o n_o}{\mu_o} k_{ro}' + \frac{c_o \rho_o}{\mu_o} k_{ro}\right) w_i \widetilde{\boldsymbol{u}}_o, \boldsymbol{v}_j\right)_k , \quad \ldots$$

- Drop slightly compressible terms. $(\widehat{A}_3^k)_{ji} \approx 0$

- Group matrices together by subdomain and interface.

$$A_1 = \begin{pmatrix} A_1^1 & & \\ & \ddots & \\ & & A_1^{N_\Omega} \end{pmatrix}, \quad C_3 = \begin{pmatrix} C_3^{12} \\ \vdots \\ C_3^{(N_\Omega - 1)N_\Omega} \end{pmatrix}$$

# Global Newton step

The 8x8 fully implicit two phase global Jacobian system:

$$\begin{bmatrix} A_1 & 0 & A_2 & 0 & 0 & A_4 & 0 & 0 \\ 0 & B_1 & 0 & B_2 & 0 & B_4 & 0 & 0 \\ C_1 & 0 & 0 & 0 & C_2 & 0 & C_3 & C_4 \\ 0 & D_1 & 0 & 0 & D_2 & D_3 & D_4 & D_5 \\ 0 & 0 & E_1 & 0 & 0 & E_2 & 0 & 0 \\ 0 & 0 & 0 & F_1 & F_2 & F_3 & 0 & 0 \\ 0 & 0 & L_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_2 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\widetilde{U}_{\mathrm{o}} \\ \delta\widetilde{U}_{\mathrm{w}} \\ \delta U_{\mathrm{o}} \\ \delta U_{\mathrm{w}} \\ \delta P_{\mathrm{o}} \\ \delta N_{\mathrm{o}} \\ \delta\Lambda_1 \\ \delta\Lambda_2 \end{bmatrix} = - \begin{bmatrix} A_{\mathrm{o}} \\ A_{\mathrm{w}} \\ D_{\mathrm{o}} \\ D_{\mathrm{w}} \\ B_{\mathrm{o}} \\ B_{\mathrm{w}} \\ H_{\mathrm{o}} \\ H_{\mathrm{w}} \end{bmatrix}$$

# Velocity elimination

- We first eliminate the 4 velocities to form 1st Schur complement:

$$\begin{bmatrix} J_{\Theta\Theta} & J_{\Theta\Lambda} \\ J_{\Lambda\Theta} & J_{\Lambda\Lambda} \end{bmatrix} \begin{bmatrix} \delta\Theta \\ \delta\Lambda \end{bmatrix} = \begin{bmatrix} R_\Theta \\ R_\Lambda \end{bmatrix}$$

Subdomain unknowns $\delta\Theta = \begin{bmatrix} \delta P_o \\ \delta N_o \end{bmatrix}$

Mortar unknowns $\delta\Lambda = \begin{bmatrix} \delta \Lambda_1 \\ \delta \Lambda_2 \end{bmatrix}$

$$J_{\Theta\Theta} = \begin{bmatrix} J_{P_o P_o} & J_{P_o N_o} \\ J_{N_o P_o} & J_{N_o N_o} \end{bmatrix} \qquad J_{\Theta\Lambda} = \begin{bmatrix} J_{P_o \Lambda_1} & J_{P_o \Lambda_2} \\ J_{N_o \Lambda_1} & J_{N_o \Lambda_2} \end{bmatrix}$$

$$J_{\Lambda\Theta} = \begin{bmatrix} J_{\Lambda_1 P_o} & J_{\Lambda_1 N_o} \\ J_{\Lambda_2 P_o} & J_{\Lambda_2 N_o} \end{bmatrix} \qquad J_{\Lambda\Lambda} = \begin{bmatrix} J_{\Lambda_1 \Lambda_1} & J_{\Lambda_1 \Lambda_2} \\ J_{\Lambda_2 \Lambda_1} & J_{\Lambda_2 \Lambda_2} \end{bmatrix}$$

# 3 Schur complements

- Starting from the saddle point system, we can form 3 different algorithms with different character by taking Schur complements:

1. Can eliminate **velocities** to form **(Θ,Λ)–Schur complement**

$$\begin{bmatrix} J_{\Theta\Theta} & J_{\Theta\Lambda} \\ J_{\Lambda\Theta} & J_{\Lambda\Lambda} \end{bmatrix} \begin{bmatrix} \delta\Theta \\ \delta\Lambda \end{bmatrix} = \begin{bmatrix} R_\Theta \\ R_\Lambda \end{bmatrix}$$

"GJ method"

2. Can eliminate **subdomain unknowns** to form **Λ–Schur complement**

$$\left(J_{\Lambda\Lambda} - J_{\Lambda\Theta} J_{\Theta\Theta}^{-1} J_{\Theta\Lambda}\right) \delta\Lambda = R_\Lambda - J_{\Lambda\Theta} J_{\Theta\Theta}^{-1} R_\Theta$$

"GJS method"

Here, the action of $J_{\Theta\Theta}^{-1}$ requires solving linear subdomain problems.

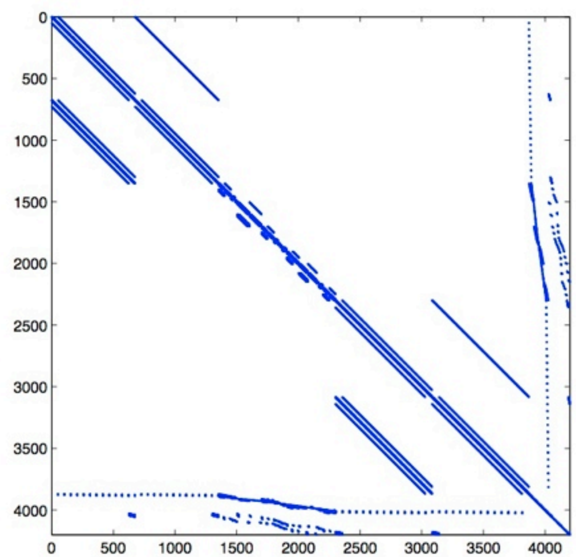
3. Can eliminate **mortar unknowns** to form **Θ–Schur complement**

$$\left(J_{\Theta\Theta} - J_{\Theta\Lambda} J_{\Lambda\Lambda}^{-1} J_{\Lambda\Theta}\right) \delta\Theta = R_\Theta - J_{\Theta\Lambda} J_{\Lambda\Lambda}^{-1} R_\Lambda$$

Here, the matrix $J_{\Lambda\Lambda}^{-1}$ can be computed with Sparse LU or mass lumping.

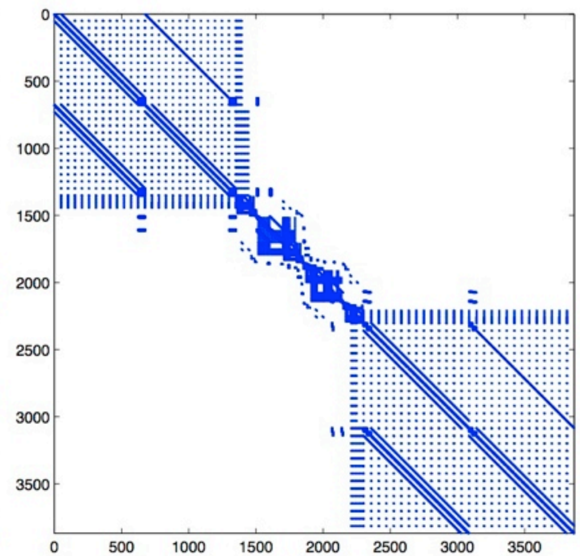
# Sparsity Pattern of GJ Matrices

Unknowns $(\delta P_O, \delta N_O, \delta\Lambda_1, \delta\Lambda_2)$



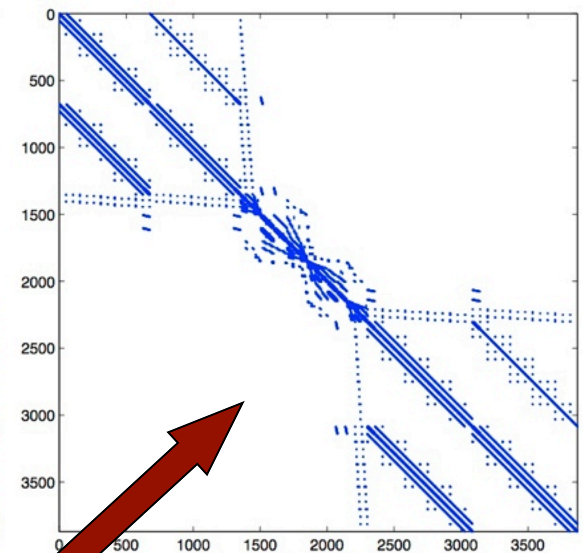

nnz=41505

Unknowns $(\delta P_O, \delta N_O)$ without mass lumping




nnz=63642

Unknowns $(\delta P_O, \delta N_O)$ with mass lumping




nnz=44075

We will precondition this system in this work.

# Choice of interface unknowns

- Flexibility in choosing physical meaning of Lagrange multipliers.
- Changes entries and condition number of GJ matrix.

- (Choice $\lambda_1 = p_{\mathrm{o}}^{\Gamma}, \lambda_2 = p_{\mathrm{w}}^{\Gamma}$).

$$(C_3^{kl})_{ji} = \left\langle \eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}, \quad (C_4^{kl})_{ji} = 0,$$

$$(D_4^{kl})_{ji} = 0, \qquad (D_5^{kl})_{ji} = \left\langle \eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}.$$

- (Choice $\lambda_1 = p_{\mathrm{o}}^{\Gamma}, \lambda_2 = p_c^{\Gamma}$). With this choice, $p_{\mathrm{w}}^{\Gamma} = \lambda_1 - \lambda_2$.

$$(C_3^{kl})_{ji} = \left\langle \eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}, \quad (C_4^{kl})_{ji} = 0,$$

$$(D_4^{kl})_{ji} = \left\langle \eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}, \quad (D_5^{kl})_{ji} = \left\langle -\eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}.$$

# Choice of interface unknowns

- (Choice $\lambda_1 = p_{\rm o}^\Gamma, \lambda_2 = n_{\rm o}^\Gamma$). Using $\rho_{\rm o}$, we have $s_{\rm w} = 1 - \lambda_2/\rho_{\rm o}$, hence

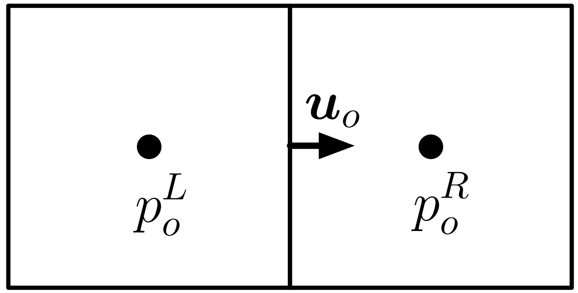
$$p_{\rm w} = \lambda_1 - p_{\rm c}\left(1 - \frac{\lambda_2}{\lambda_1}\right).$$

$$(C_3^{kl})_{ji} = \left\langle \eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}, \qquad (C_4^{kl})_{ji} = 0,$$

$$(D_4^{kl})_{ji} = \left\langle \left(1 - c_{\rm o}\frac{p_{\rm c}'\lambda_2}{\rho_{\rm o}}\right)\eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}, \quad (D_5^{kl})_{ji} = \left\langle \frac{p_{\rm c}'}{\rho_{\rm o}}\eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}.$$

$$(D_4^{kl})_{ji} \approx \left\langle \eta_j^{kl}, \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k \right\rangle_{kl}.$$

# Upwinding on a single domain

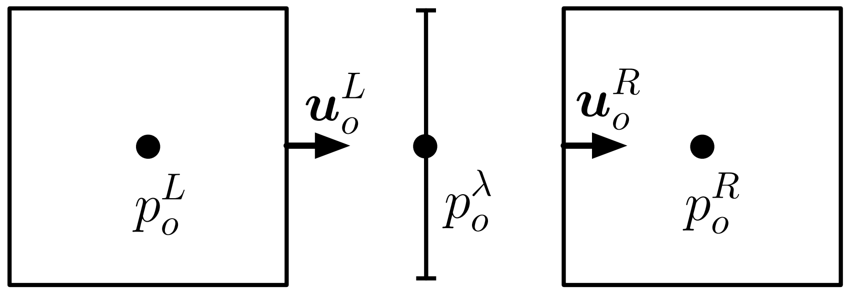


$$\triangle p_o \approx p_o^R - p_o^L$$

$$m_o^{up} = \begin{cases} m_o^L, & \text{if} \quad \triangle p_o < 0 \\ m_o^R, & \text{if} \quad \triangle p_o > 0 \end{cases}$$

$$\int_\Omega m_o \, \boldsymbol{u}_o \cdot \boldsymbol{u}_o dx \underset{TM}{\approx} m_o^{up} \times \left( \frac{h_x^L}{2\, h_y\, h_z} + \frac{h_x^R}{2\, h_y\, h_z} \right)$$

# Upwinding "through a mortar"



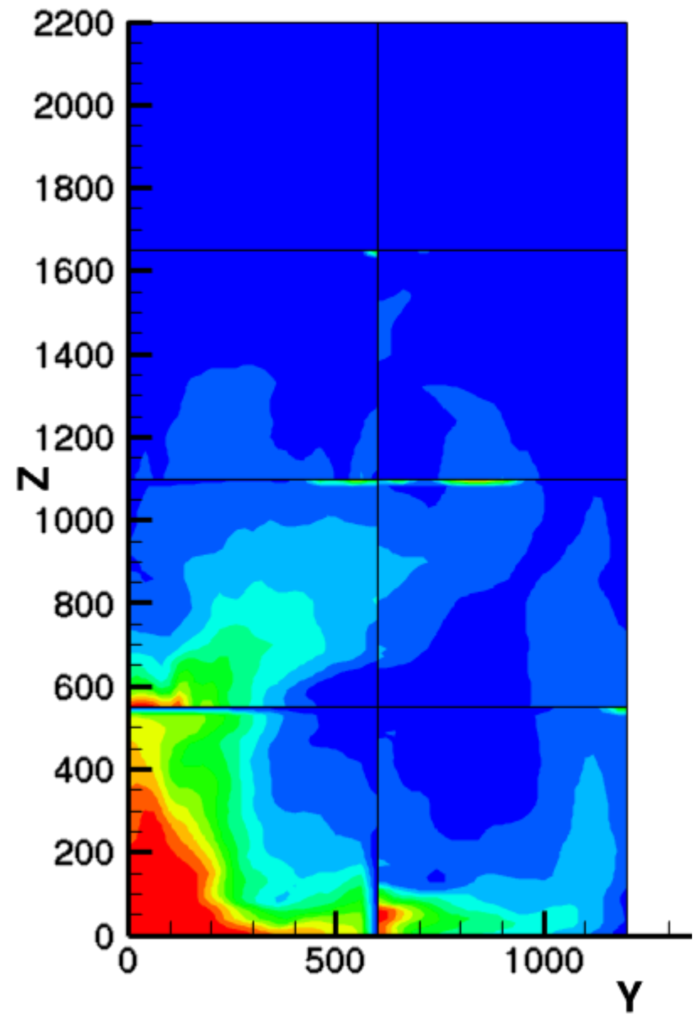

$$\triangle p_o^L \approx p_o^\lambda - p_o^L \qquad \triangle p_o^R \approx p_o^R - p_o^\lambda$$

$$m_o^{up,L} = \begin{cases} m_o^L, & \text{if } \triangle p_o^L < 0 \\ m_o^\lambda, & \text{if } \triangle p_o^L > 0 \end{cases} \qquad m_o^{up,R} = \begin{cases} m_o^\lambda, & \text{if } \triangle p_o^R < 0 \\ m_o^R, & \text{if } \triangle p_o^R > 0 \end{cases}$$

$$\int_{E^L} m_o\, \boldsymbol{u}_o^L \cdot \boldsymbol{u}_o^L dx \underset{TM}{\approx} m_o^{up,L} \times \left( \frac{h_x^L}{2\, h_y\, h_z} \right) \qquad \int_{E^R} m_o\, \boldsymbol{u}_o^R \cdot \boldsymbol{u}_o^R dx \underset{TM}{\approx} m_o^{up,R} \times \left( \frac{h_x^R}{2\, h_y\, h_z} \right)$$

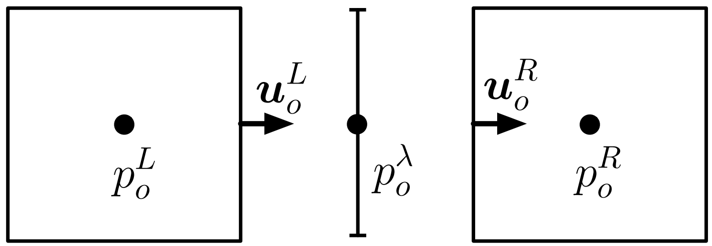
# What can go wrong?

- Excessive time step cuts
- Singular linear systems
- Loss of nonlinear convergence
- Loss of mass conservation
  - No guarantee that $p^L < p^\lambda < p^R$ or $p^L > p^\lambda > p^R$
  - May create artificial sources/sinks on interfaces




Center for Subsurface Modeling

# Upwinding “block-to-block”




This technique was used in enhanced velocity method and IMPES models.
**It is new for the fully-implicit model.**

$$\triangle p_o \approx p_o^R - p_o^L \quad \text{by directly projecting} \quad \Omega^L|_\Gamma \longleftrightarrow \Omega^R|_\Gamma$$

$$m_o^{up} = \begin{cases} m_o^L, & \text{if } \triangle p_o < 0 \\ m_o^R, & \text{if } \triangle p_o > 0 \end{cases}$$

$$\int_{E^R} m_o \, \boldsymbol{u}_o^R \cdot \boldsymbol{u}_o^R dx \underset{TM}{\approx} m_o^{up} \times \left( \frac{h_x^R}{2\, h_y \, h_z} \right)$$

$$\int_{E^L} m_o \, \boldsymbol{u}_o^L \cdot \boldsymbol{u}_o^L dx \underset{TM}{\approx} m_o^{up} \times \left( \frac{h_x^L}{2\, h_y \, h_z} \right)$$

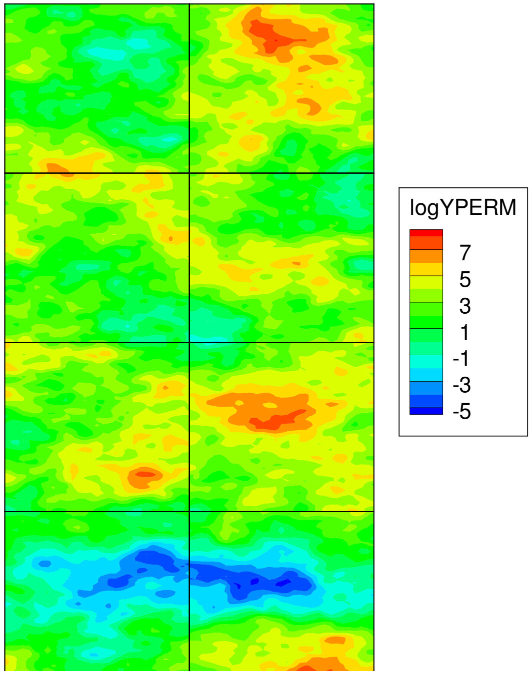
**Important consequences:**

- No saturation info. is needed on interfaces.
- No longer need $Pc^{-1}$ with extra “interface Newton”.
- Sw is allowed to be discontinuous even when using a continuous mortar.

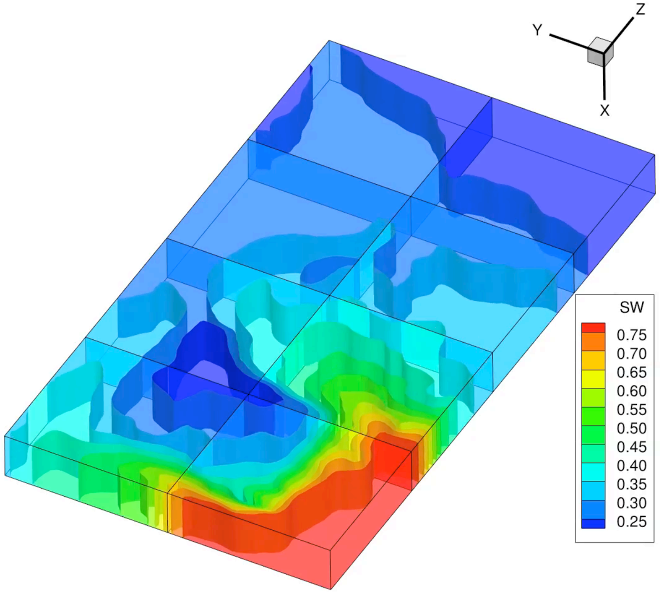
# Heterogeneous Case

- Challenging SPE10 industrial benchmark case, layer 1
- 8 subdomains, matching P0 mortars
- Two-phase flow with gravity, compressibility, capillary pressure

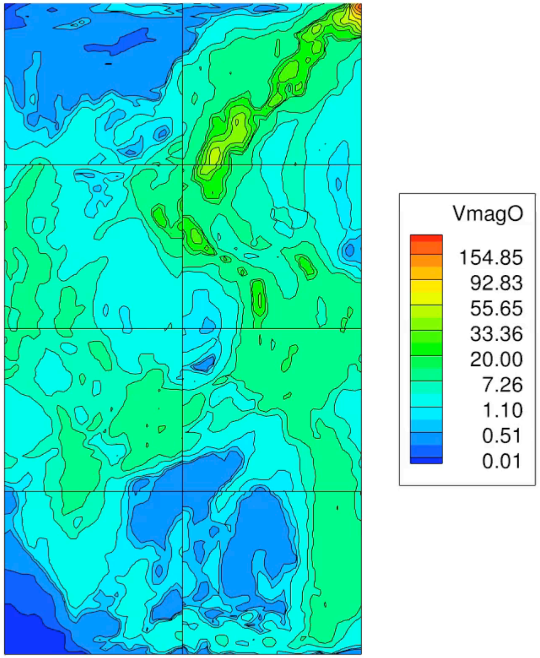
Log Permeability



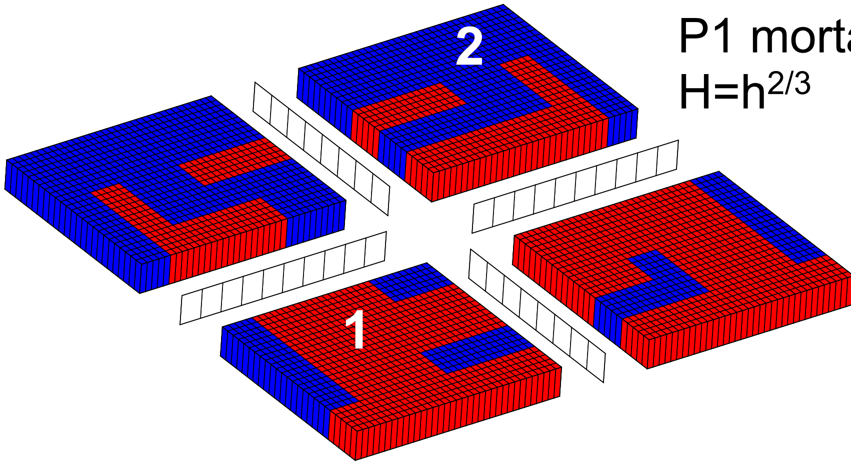

Water Saturation




Oil Velocity Magnitude



# Two Rock Type Example




P1 mortar
$H=h^{2/3}$

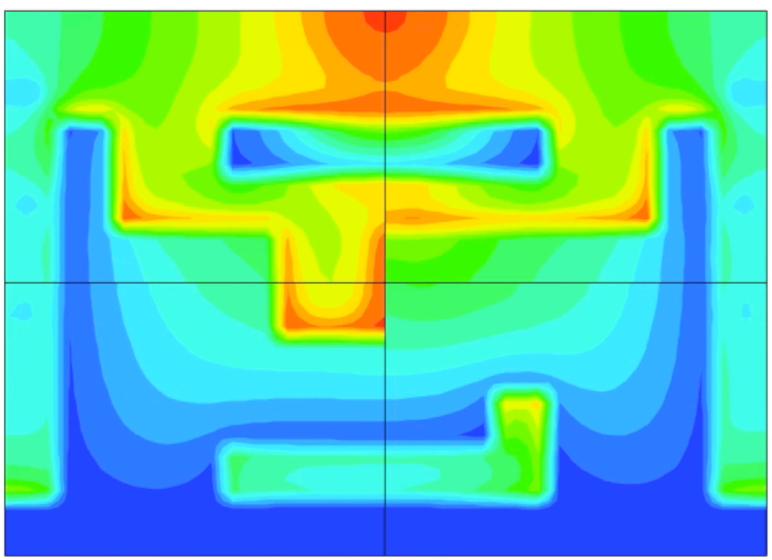
## Two Rock Types

| | $p_d$ | $\lambda$ | $K$ | $\phi$ |
|---|---|---|---|---|
| **rock type 1** | 135 psi | 2.49 | 504 md | 0.2 |
| **rock type 2** | 37.7 psi | 3.86 | 52.6 md | 0.2 |

## Capillary Pressure

$$p_c(s_w) = \begin{cases} p_d\, s_{c1}^{-1/\lambda}, & \text{if } 0 \le s_e < s_{c1} \\ p_d\, s_e^{-1/\lambda}, & \text{if } s_{c1} \le s_e \le s_{c2} \\ p_d\, s_{c2}^{-1/\lambda}\, \frac{1-s_e}{1-s_{c2}}, & \text{if } s_{c2} < s_e \le 1 \end{cases}$$

## Water Saturation



## Effective Saturation

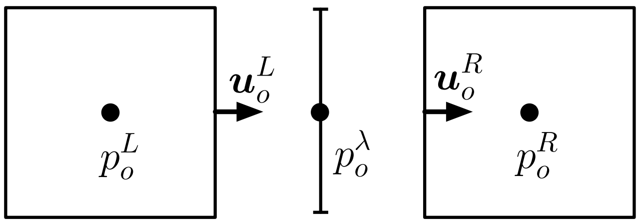
$$s_e = \frac{s_w - s_{rw}}{1 - s_{rw} - s_{ro}}$$

## Relative Permeability

$$k_{rw} = 0.9\, s_e^2$$

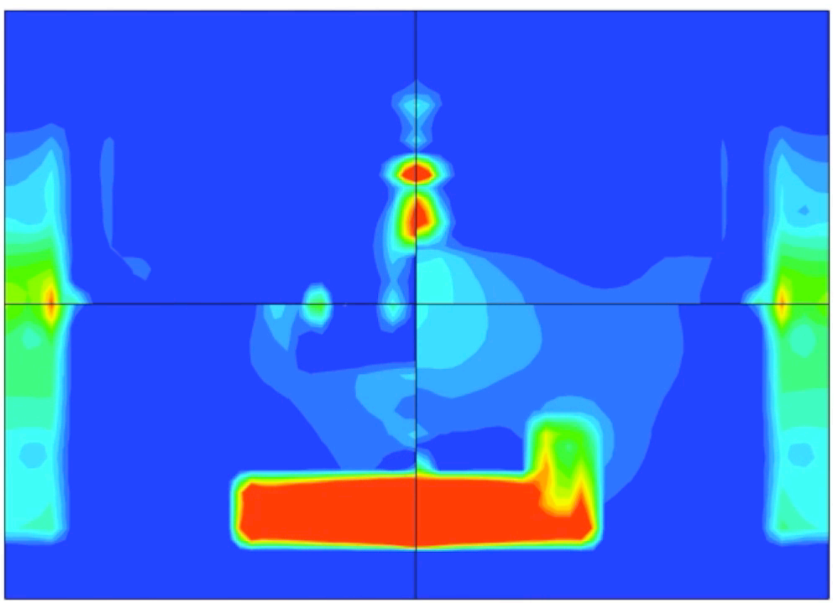
$$k_{ro} = 0.5\,(1 - s_e)^2$$

# Saturation Errors



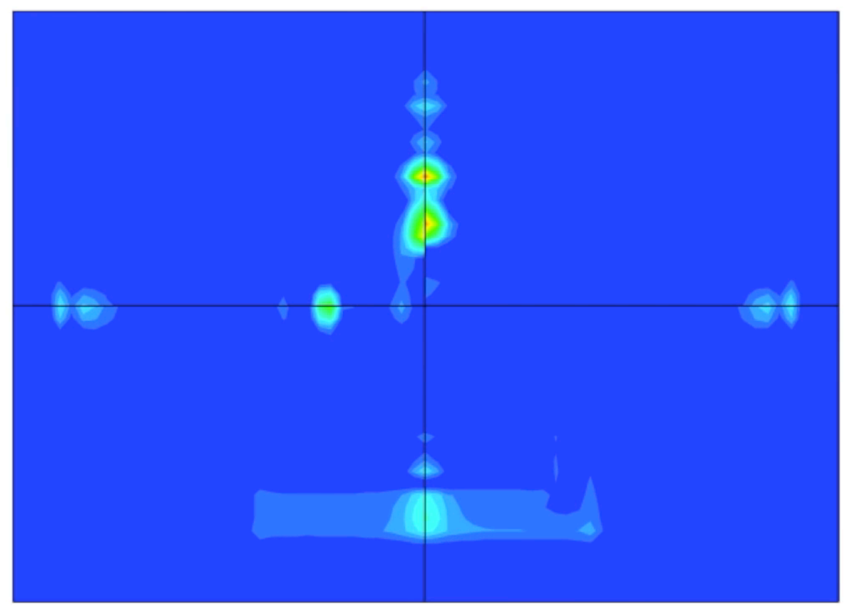

Accurate integration of phase mobility can improve mass conservation and solvability of linear and nonlinear systems.

Upwind using Lagrange multiplier



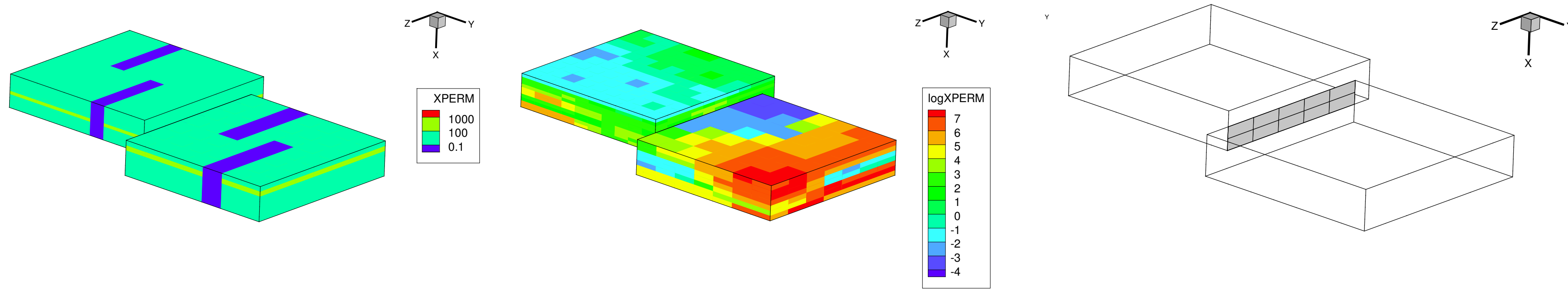
**Max. Pointwise Error = 0.37**

Upwind using adjacent subdomain values



**Max. Pointwise Error = 0.07**

# Global Jacobian compared to Forward Difference Algorithm




| FD Method | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Perm. | Intf. Newton | | Intf. GMRES | | | Subdom. Newton | | | CPU |
| | Tot. | Avg. 1 | Tot. | Avg. 1 | Avg. 2 | Tot. | Avg. 1 | Avg. 2 | Time |
| Barrier | 331 | 1.66 | 6,355 | 31.78 | 19.20 | 20,662 | 103.31 | 3.25 | 161.49 |
| Heterog. | 241 | 1.21 | 2,629 | 13.15 | 10.91 | 9,212 | 46.06 | 3.50 | 71.18 |

| GJ Method | | | |
|---|---|---|---|
| Perm. | Global Newton | | CPU |
| | Tot. | Avg. 1 | Time |
| Barrier | 342 | 1.71 | 11.80 |
| Heterog. | 212 | 1.06 | 7.71 |

**FD**: best preconditioned GMRES and loose inner tolerances

**GJ**: direct solver

# Two-Stage Preconditioning

Two-Stage Preconditioners (or similar ideas) are necessary in fully-implicit multiphase models, because the linear systems have both elliptic and hyperbolic behaviors.

We applied the following preconditioner to the global Jacobian multiscale mortar system:

- Lacroix, S., Vassilevski, Y., Wheeler, M.F., 2001. Decoupling preconditioners in the implicit parallel accurate reservoir simulator (IPARS). *Numerical linear algebra with applications*, **8** (8), pp. 537–549.
  - Four decoupling approaches are discussed:
    - Constrained Pressure Reduction (CPR)
    - **Householder Reflection Decoupling** 🡐 We followed this approach.
    - Quasi-IMPES Decoupling
    - True IMPES Decoupling

# More Two-Stage References


- Vassilevski, P.S., 1984. Fast algorithm for solving a linear algebraic problem with separable variables. *Dokladi Na Bolgarskata Akademiya Na Naukite*, **37** (3): 305–308.
- Wallis, J.R., Kendall, R.P., and Little, T.E., 1985. Constrained residual acceleration of conjugate residual methods. In *SPE Reservoir Simulation Symposium*, SPE 13536.
- Cao, H., Tchelepi, H.A., Wallis, J.R., et al. 2005. Parallel scalable unstructured CPR-type linear solver for reservoir simulation. In *SPE Annual Technical Conference and Exhibition*. SPE 96809.
- Han, C. et al., 2013. Adaptation of the CPR preconditioner for efficient solution of the adjoint equation. *SPE Journal*, 18(02), pp. 207–213.

# Two-Stage Preconditioning for GJ

- Begin with the Schur complement system for subdomain unknowns.

$$J^3\ \delta\Theta = (J_{\Theta\Theta} - J_{\Theta\Lambda}J_{\Lambda\Lambda}^{-1}J_{\Lambda\Theta})\ \delta\Theta = R_\Theta - J_{\Theta\Lambda}J_{\Lambda\Lambda}^{-1}R_\Lambda = R^3.$$

- Perform Householder (QR) factorization to diagonal 2x2 blocks.

$$(P^{-1}Q^T P\ J^3)\ \delta\Theta = P^{-1}Q^T P\ R^3$$

$$\Leftrightarrow H\ \delta\Theta = \begin{bmatrix} H_{P_O P_O} & H_{P_O N_O} \\ H_{N_O P_O} & H_{N_O N_O} \end{bmatrix} \begin{bmatrix} \delta P_O \\ \delta N_O \end{bmatrix} = \begin{bmatrix} b_{P_O} \\ b_{N_O} \end{bmatrix} = b.$$

- Inside the outer **gmres**, get action Y = $M^{-1}$ Z in a three step process:

  1. Solve the pressure equation $Y_{Po}$ = **gmres**$(H_{PoPo}, Z_{Po})$ with preconditioner $M_{1S}^{-1}$ to a specified tolerance.
  2. Update the linear residual $R = Z - H[Y_{Po}, 0]$.
  3. Solve the second stage equation $Y$ = **gmres**$(H,R) + [Y_{Po}, 0]$ with preconditioner $M_{2S}^{-1}$ to a specified tolerance.

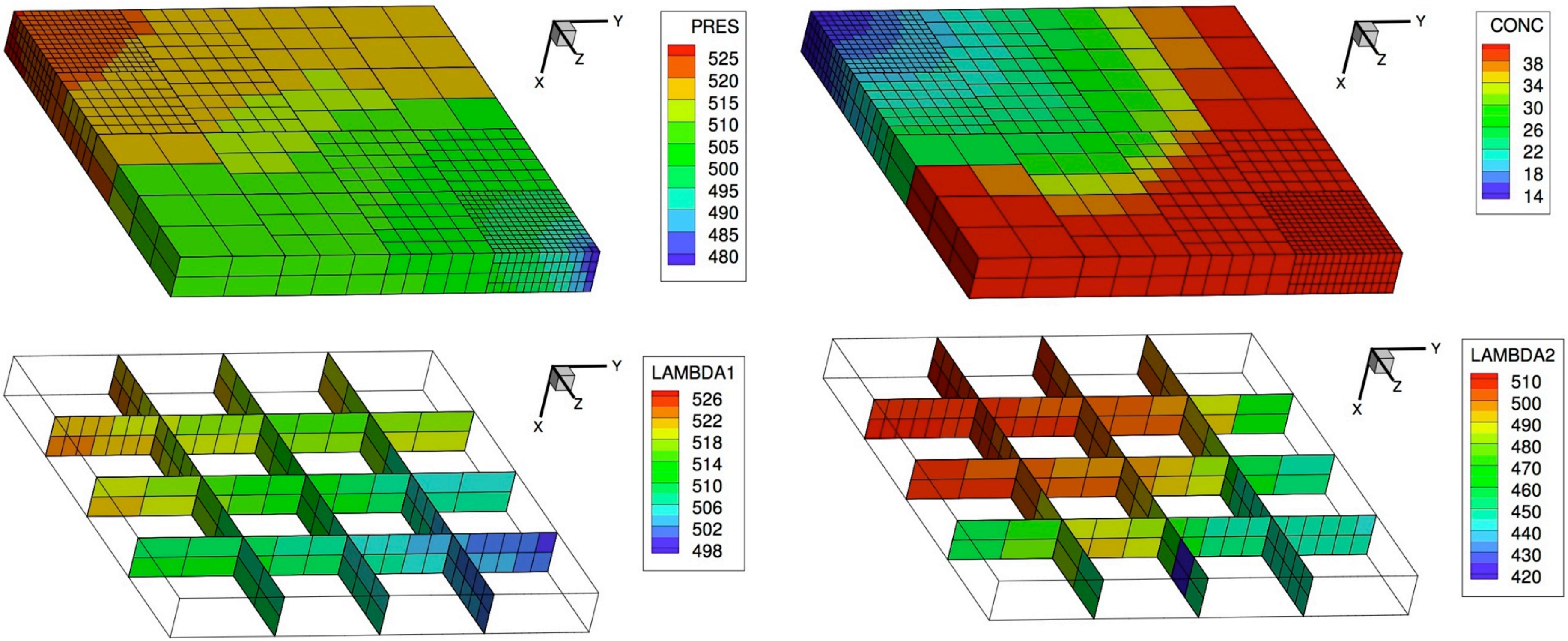
# Example 1: The full SPE10 benchmark problem with mortars in two-phase model

| CPU cores/ Subdomains | Total CPU time | Total Newton Steps Taken | Avg. Outer GMRES Iter. per Newton step | Time Step Cuts |
|---|---|---|---|---|
| 1×1×1=1 | 8331.79 | 51 | **4.88** | 0 |
| 1×1×2=2 | 4675.22 | 51 | **5.00** | 0 |
| 1×1×4=4 | 3102.14 | 52 | **5.65** | 1 |
| 1×2×4=8 | 2727.95 | 51 | **5.04** | 0 |
| 1×2×8=16 | 1216.14 | 52 | **5.71** | 1 |
| 1×4×8=32 | 517.69 | 51 | **5.02** | 0 |
| 1×4×16=64 | 618.41 | 109 | **5.71** | 2 |

1st Stage: GMRES(20), 1e–6 tolerance, 100 max iterations, $M_{1S}^{-1}$ = AMG V-cycle, 1 sweep ILU(0) smoother, coarse solve 1000x1000 with Sparse LU.

2nd Stage: GMRES(20), 1e–3 tolerance, no restarts, $M_{2S}^{-1} = M_{1S}^{-1}$.

# Example 2: A multiscale problem on non-matching subdomain grids




<u>1st Stage</u>: GMRES(20), 1e–3 tolerance, no restarts, $M_{1S}^{-1}$ = AMG V-cycle, 1 sweep ILU(0) smoother, coarse solve 1000x1000 with Sparse LU.

<u>2nd Stage</u>: GMRES(1), $M_{2S}^{-1}$ = *5 Gauss-Seidel iterations.*

# Example 3: A heterogeneous 10M Cell Problem with mortars on 1024 Processors

| | |
|---|---|
| Total time steps | 1007 |
| Total Newton iterations | 1007 |
| Total outer GMRES iterations | 2449 |
| **Average GMRES iterations per Newton step** | **2.43** |
| Average Newton iterations per time step | 1.00 |
| Total time step cuts | 0 |

| | |
|---|---|
| Matrix assembly time | 86.04 |
| Outer GMRES time | 8459.16 |
| Householder decoupling time | 42.25 |
| Pressure solve GMRES time | 1394.55 |
| Second stage GMRES time | 3340.99 |
| Mass lumping time | 0.05 |
| Matrix-matrix multiply time | 1206.87 |
| Total CPU time | 8571.76 |

1st Stage: GMRES(20), 1e–3 tolerance, no restarts, $M_{1S}^{-1}$ = AMG V-cycle, 1 sweep ILU(0) smoother, coarse solve 1000x1000 with Sparse LU.

2nd Stage: GMRES(20), 1e–3 tolerance, no restarts, $M_{2S}^{-1} = M_{1S}^{-1}$.

# Conclusions

- We have developed new mortar algorithms using global linearization for single and two phase flow.
  - Easy to implement, fewer nested iterations and tolerances.
  - Inexpensive, showed parallel scalability for nonlinear problems.
  - Changed upwinding near interfaces for better fluid transport.
  - Applied two-stage preconditioner for parallel scalability.

# References


- Ganis, B., Juntunen, M., Pencheva, G., Wheeler, M.F., and Yotov, I. 2014. A global Jacobian method for mortar discretizations of nonlinear porous media flows. *SIAM Journal on Scientific Computation* **36** (2): A522–A542.

- Ganis, B., Kumar, K., Pencheva, G., Wheeler, M.F., and Yotov, I. 2014. A global Jacobian method for mortar discretizations of a fully-implicit two-phase flow model. *Multiscale Modeling & Simulation* **12** (4): 1401–1423.

- Ganis, B., Kumar, K., Pencheva, G., Wheeler, M.F., Yotov, I. A multiscale mortar method and two-stage preconditioner for multiphase flow using a global Jacobian approach. SPE 172990-MS.


Thank you!