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

- Example:
  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.
**Single Phase Mortar Theory**


**Forward Difference (FD) Algorithms for Nonlinear problems**


**Global Jacobian (GJ) Algorithms for Nonlinear problems**

- 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

**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 forms 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

◊ = convergence check
□ = forward difference approximation used
Algorithms for nonlinear mortar problems

New Methods

**GJ Method**
- Time Step
- Nonlinear Global Newton Step
- Linear Global GMRES Step
- = convergence check

**GJS Method**
- Time Step
- Nonlinear Global Newton Step
- Linear Interface GMRES Step
- Linear Subdomain GMRES Step
- = forward difference approximation used

**Prior Method**

**FD Method**
- Time Step
- Nonlinear Interface FD-Newton Step
- Linear Interface GMRES Step
- Linear Subdomain GMRES Step
- Nonlinear Subdom. Newton Step
- Linear Subdom. GMRES Step

Ben Ganis | GJ Mortar Algorithms | bganis@ices.utexas.edu
Novelty of this work

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

Two Phase Model

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

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

Darcy Law: $u_\alpha = \frac{k_{r\alpha} \rho_\alpha}{\mu_\alpha} \tilde{u}_\alpha$ in $\Omega^k \times (0, T)$

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

Boundary condition: $u \cdot n = 0$ on $\partial \Omega \times (0, T)$

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

Flux continuity: $u^k_\alpha \cdot n^k + u^l_\alpha \cdot n^l = 0$ on $\Gamma^{kl} \times (0, T)$

Saturation constraint: $s_w + s_o = 1$

Capillary pressure: $p_c(s_w) = p_o - p_w$

Slightly compressible density: $\rho_\alpha(p_\alpha) = \rho_\alpha^{ref} e^{c_\rho p_\alpha}$.
Finite element discretization

Primary Unknowns: \((p_o, n_o)\)

Phase Velocities: \((\tilde{u}_o, \tilde{u}_w, u_o, u_w)\)

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

Velocity, Pressure, Mortar Spaces:

\[
V_h = \bigoplus_{k=1}^{N_\Omega} V_h^k, \quad W_h = \bigoplus_{k=1}^{N_\Omega} W_h^k, \quad M_H = \bigoplus_{k=1}^{N_\Omega} M_H^{kl}.
\]

Time discretization:

\[
0 = t^0 < t^1 < \cdots < t^{NT} = T, \text{ with } \delta t^n = t^n - t^{n-1}.
\]
Expanded multiscale mortar method for fully-implicit two-phase flow:

Phase concentration: \[ n_{\alpha} = \rho_{\alpha} s_{\alpha} \]

Phase mobility: \[ m_{\alpha} = \frac{k_{\tau\alpha} \rho_{\alpha} \mu_{\alpha}}{\mu_{\alpha}} \]

\[ A_{\alpha}^k = \int_{\Omega_k} \mathbf{u}_{\alpha}^k \cdot \mathbf{v} \, dx - \int_{\Omega_k} m_{\alpha} \tilde{\mathbf{u}}_{\alpha}^k \cdot \mathbf{v} \, dx = 0, \]

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

\[ B_{\alpha}^k = \int_{\Omega_k} \frac{\phi n_{\alpha}^k - \phi n_{\alpha}^{n-1}}{\delta t} w \, dx + \int_{\Omega_k} \nabla \cdot \mathbf{u}_{\alpha}^k w \, dx - \int_{\Omega_k} q_{\alpha} w \, dx = 0, \]

\[ H_{\alpha} = \int_{\Gamma_{kl}} (\mathbf{u}_{\alpha}^k \cdot \mathbf{n}_k + \mathbf{u}_{\alpha}^l \cdot \mathbf{n}_l) \mu \, d\sigma = 0. \]
Express 8 unknowns as linear combinations of finite element basis functions, insert into discrete form.

\[ p_O^k = \sum_{i=1}^{N_p^k} P_{O,i}^k w_i^k \]

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

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

\[ B_\alpha^k = \int_{\Omega^k} \frac{\phi n_\alpha^k - \phi n_\alpha^{n-1}}{\delta t} w \, dx + \int_{\Omega^k} \nabla \cdot u_\alpha^k w \, dx - \int_{\Omega^k} q_\alpha w \, dx = 0, \]

\[ H_\alpha = \int_{\Gamma_{kl}} (u_{\alpha}^k \cdot n_k + u_{\alpha}^l \cdot n_l) \mu \, d\sigma = 0. \]

Obtain a nonlinear system for the global coefficient vectors:

\[ \tilde{U}_o, \tilde{U}_w, U_o, U_w \in \mathbb{R}^{N_u} \quad P_o, N_o \in \mathbb{R}^{N_p}. \]

\[ \Lambda_1, \Lambda_2 \in \mathbb{R}^{N_\lambda} \]

\[ N_u = \sum_{i=1}^{N_\Omega} N_u^k \quad N_p = \sum_{i=1}^{N_\Omega} N_p^k \quad N_\lambda = \sum_{1 \leq k < l \leq N_\Omega} N_{\lambda}^{kl} \]
Global nonlinear system

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

\[
\begin{align*}
A_o(\widetilde{U}_o, U_o, P_o, N_o) &= 0 \\
A_w(\widetilde{U}_w, U_w, P_o, N_o) &= 0 \\
D_o(\widetilde{U}_o, P_o, \Lambda_1, \Lambda_2) &= 0 \\
D_w(\widetilde{U}_w, P_o, N_o, \Lambda_1, \Lambda_2) &= 0 \\
B_o(U_o, N_o) &= 0 \\
B_w(U_w, P_o, N_o) &= 0 \\
H_o(U_o) &= 0 \\
H_w(U_w) &= 0
\end{align*}
\]

\{ Aux. Velocity \}
\{ Darcy Velocity \}
\{ Mass Balance \}
\{ Flux Continuity \}
Forming Jacobian entries

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

\[
(A^k_1)_{ji} = \frac{\partial A^k_{o,j}}{\partial U^i_{o,i}} = - (m_o v_i, v_j)_k,
\]

\[
(A^k_2)_{ji} = \frac{\partial A^k_{o,j}}{\partial U^i_{o,i}} = (v_i, v_j)_k,
\]

\[
(\hat{A}^k_3)_{ji} = \frac{\partial A^k_{o,j}}{\partial P^i_{o,i}} = - \left( \left( \frac{c_o n_o}{\mu_o} k^i_{ro} + \frac{c_o \rho_o}{\mu_o} k^i_{ro} \right) w_i u_0, v_j \right)_k,
\]

- Drop slightly compressible terms. \((\hat{A}^k_3)_{ji} \approx 0\)

- Group matrices together by subdomain and interface.

\[
A_1 = \begin{pmatrix}
A^1_1 & & \\
& \ddots & \\
& & A^{N_\Omega}_1
\end{pmatrix}, \quad C_3 = \begin{pmatrix}
C^{12}_3 \\
\vdots \\
C^{(N_\Omega-1)N_\Omega}_3
\end{pmatrix}
\]
Global Newton step

The 8x8 fully implicit two phase global Jacobian system:

\[
\begin{bmatrix}
A_1 & 0 & A_2 & 0 & 0 & A_4 & 0 & 0 \\
0 & B_1 & 0 & B_2 & 0 & B_4 & 0 & 0 \\
C_1 & 0 & 0 & 0 & C_2 & 0 & C_3 & C_4 \\
0 & D_1 & 0 & 0 & D_2 & D_3 & D_4 & D_5 \\
0 & 0 & E_1 & 0 & 0 & E_2 & 0 & 0 \\
0 & 0 & 0 & F_1 & F_2 & F_3 & 0 & 0 \\
0 & 0 & L_1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & L_2 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
\delta \tilde{U}_o \\
\delta \tilde{U}_w \\
\delta U_o \\
\delta U_w \\
\delta P_o \\
\delta N_o \\
\delta \Lambda_1 \\
\delta \Lambda_2 \\
\end{bmatrix}
= -
\begin{bmatrix}
A_o \\
A_w \\
D_o \\
D_w \\
B_o \\
B_w \\
H_o \\
H_w \\
\end{bmatrix}
\]
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 \( \delta\Lambda = \begin{bmatrix} \delta\Lambda_1 \\ \delta\Lambda_2 \end{bmatrix} \)

\[
J_{\Theta\Theta} = \begin{bmatrix}
J_{P_0P_0} & J_{P_0N_0} \\
J_{N_0P_0} & J_{N_0N_0}
\end{bmatrix}
\quad J_{\Theta\Lambda} = \begin{bmatrix}
J_{P_0\Lambda_1} & J_{P_0\Lambda_2} \\
J_{N_0\Lambda_1} & J_{N_0\Lambda_2}
\end{bmatrix}
\]

\[
J_{\Lambda\Theta} = \begin{bmatrix}
J_{\Lambda_1P_0} & J_{\Lambda_1N_0} \\
J_{\Lambda_2P_0} & J_{\Lambda_2N_0}
\end{bmatrix}
\quad J_{\Lambda\Lambda} = \begin{bmatrix}
J_{\Lambda_1\Lambda_1} & J_{\Lambda_1\Lambda_2} \\
J_{\Lambda_2\Lambda_1} & J_{\Lambda_2\Lambda_2}
\end{bmatrix}
\]
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 \((\Theta, \Lambda)\)–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 \(\Lambda\)–Schur complement

\[
(J_{\Lambda\Lambda} - J_{\Lambda\Theta}J_{\Theta\Theta}^{-1}J_{\Theta\Lambda}) \delta\Lambda = R_{\Lambda} - J_{\Lambda\Theta}J_{\Theta\Theta}^{-1}R_{\Theta}
\]

“GJS method”

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

3. Can eliminate mortar unknowns to form \(\Theta\)–Schur complement

\[
(J_{\Theta\Theta} - J_{\Theta\Lambda}J_{\Lambda\Lambda}^{-1}J_{\Lambda\Theta}) \delta\Theta = R_{\Theta} - J_{\Theta\Lambda}J_{\Lambda\Lambda}^{-1}R_{\Lambda}
\]

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

Ben Ganis | GJ Mortar Algorithms | bganis@ices.utexas.edu
Sparsity Pattern of GJ Matrices

Unknowns $(\delta P_o, \delta N_o, \delta \Lambda_1, \delta \Lambda_2)$

Unknowns $(\delta P_o, \delta N_o)$ without mass lumping

Unknowns $(\delta P_o, \delta N_o)$ with mass lumping

We will precondition this system in this work.

nnz=41505  nnz=63642  nnz=44075
Choice of interface unknowns

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

- (Choice $\lambda_1 = p_o^\Gamma$, $\lambda_2 = p_w^\Gamma$).

$$
(C')_{3kl}^{ji} = \left\langle \eta_j^{kl}, \mathbf{v}_i^k \cdot \mathbf{n}^k \right\rangle_{kl}, \quad (C')_{4kl}^{ji} = 0,
$$

$$
(D')_{4kl}^{ji} = 0, \quad (D')_{5kl}^{ji} = \left\langle \eta_j^{kl}, \mathbf{v}_i^k \cdot \mathbf{n}^k \right\rangle_{kl}.
$$

- (Choice $\lambda_1 = p_o^\Gamma$, $\lambda_2 = p_c^\Gamma$). With this choice, $p_w^\Gamma = \lambda_1 - \lambda_2$.

$$
(C')_{3kl}^{ji} = \left\langle \eta_j^{kl}, \mathbf{v}_i^k \cdot \mathbf{n}^k \right\rangle_{kl}, \quad (C')_{4kl}^{ji} = 0,
$$

$$
(D')_{4kl}^{ji} = \left\langle \eta_j^{kl}, \mathbf{v}_i^k \cdot \mathbf{n}^k \right\rangle_{kl}, \quad (D')_{5kl}^{ji} = \left\langle -\eta_j^{kl}, \mathbf{v}_i^k \cdot \mathbf{n}^k \right\rangle_{kl}.
$$
Choice of interface unknowns

- (Choice $\lambda_1 = p_o^\Gamma, \lambda_2 = n_o^\Gamma$). Using $\rho_o$, we have $s_w = 1 - \lambda_2/\rho_o$, hence

$$p_w = \lambda_1 - p_c \left(1 - \frac{\lambda_2}{\lambda_1}\right).$$

$$(C_{3}^{kl})_{ji} = \left\langle \eta^{kl}_j, \nu^{k}_i \cdot n^k \right\rangle_{kl}, \quad (C_{4}^{kl})_{ji} = 0,$$

$$(D_{4}^{kl})_{ji} = \left\langle \left(1 - c_o \frac{p'_c \lambda_2}{\rho_o}\right) \eta^{kl}_j, \nu^{k}_i \cdot n^k \right\rangle_{kl}, \quad (D_{5}^{kl})_{ji} = \left\langle \frac{p'_c}{\rho_o} \eta^{kl}_j, \nu^{k}_i \cdot n^k \right\rangle_{kl}.$$

$$(D_{4}^{kl})_{ji} \approx \left\langle \eta^{kl}_j, \nu^{k}_i \cdot n^k \right\rangle_{kl}.$$
Upwinding on a single domain

\[ \Delta p_o \approx p_o^R - p_o^L \]

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

\[ \int_\Omega m_o u_o \cdot u_o dx \approx TM m_o^{up} \times \left( \frac{h_x^L}{2 h_y h_z} + \frac{h_x^R}{2 h_y h_z} \right) \]
Upwinding “through a mortar”

\[ \Delta p^L_o \approx p^\lambda_o - p^L_o \]

\[ \Delta p^R_o \approx p^R_o - p^\lambda_o \]

\[
m^{up,L}_o = \begin{cases} 
  m^L_o, & \text{if } \Delta p^L_o < 0 \\
  m^\lambda_o, & \text{if } \Delta p^L_o > 0 
\end{cases}
\]

\[
m^{up,R}_o = \begin{cases} 
  m^\lambda_o, & \text{if } \Delta p^R_o < 0 \\
  m^R_o, & \text{if } \Delta p^R_o > 0 
\end{cases}
\]

\[
\int_{E^L} m_o \hat{u}^L_o \cdot \hat{u}^L_o \, dx \approx m^{up,L}_o \times \left( \frac{h^L_x}{2 \, h_y \, h_z} \right)
\]

\[
\int_{E^R} m_o \hat{u}^R_o \cdot \hat{u}^R_o \, dx \approx m^{up,R}_o \times \left( \frac{h^R_x}{2 \, h_y \, h_z} \right)
\]
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
Upwinding “block-to-block”

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

\[ \Delta p_o \approx p_o^R - p_o^L \] by directly projecting \[ \Omega^L|_{\Gamma} \leftrightarrow \Omega^R|_{\Gamma} \]

\[ m_{o}^{up} = \begin{cases} m_{o}^L, & \text{if } \Delta p_o < 0 \\ m_{o}^R, & \text{if } \Delta p_o > 0 \end{cases} \]

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

LogYPERM

7
5
3
1
-1
-3
-5
Two Rock Type Example

Two Rock Types

<table>
<thead>
<tr>
<th></th>
<th>$p_d$</th>
<th>$\lambda$</th>
<th>$K$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>rock type 1</td>
<td>135</td>
<td>2.49</td>
<td>504</td>
<td>0.2</td>
</tr>
<tr>
<td>rock type 2</td>
<td>37.7</td>
<td>3.86</td>
<td>52.6</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Capillary Pressure

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

Effective Saturation

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

Relative Permeability

$$k_{rw} = 0.9 s_e^2$$
$$k_{ro} = 0.5 (1 - s_e)^2$$

Water Saturation

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

Ben Ganis | GJ Mortar Algorithms | bganis@ices.utexas.edu
Saturation Errors

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

Upwind using Lagrange multiplier

Upwind using adjacent subdomain values

Max. Pointwise Error = 0.37

Max. Pointwise Error = 0.07
porosity in this case is subdomain subdomain of SPE10 case (right).

This example has non-conforming spatial geometry in the form of a fault between two domains with coarse mortars (right) for Example 6.2.

Figure 7: Absolute permeability (left) and an exploded view of the division into four subdomains with different permeability levels (right).

Table 2: Computational cost for Example 6.1 measured in terms of iteration counts and CPU time.

<table>
<thead>
<tr>
<th>Perm.</th>
<th>FD Method</th>
<th>GJ Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tot.</td>
<td>Avg. 1</td>
</tr>
<tr>
<td>Barrier</td>
<td>331</td>
<td>1.66</td>
</tr>
<tr>
<td>Heterog.</td>
<td>241</td>
<td>1.21</td>
</tr>
</tbody>
</table>

**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:


  - 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


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.

\[ (P^{-1}Q^TPJ^3) \delta \Theta = P^{-1}Q^TPR^3 \]

\[ \Leftrightarrow H \delta \Theta = \begin{bmatrix} H_{P_0P_0} & H_{P_0N_0} \\ H_{N_0P_0} & H_{N_0N_0} \end{bmatrix} \begin{bmatrix} \delta P_0 \\ \delta N_0 \end{bmatrix} = \begin{bmatrix} b_{P_0} \\ b_{N_0} \end{bmatrix} = b. \]

- Inside the outer \texttt{gmres}, get action \( Y = M^{-1}Z \) in a three step process:

1. Solve the pressure equation \( Y_{P_0} = \texttt{gmres}(H_{P_0P_0}, Z_{P_0}) \) with preconditioner \( M_{1S}^{-1} \) to a specified tolerance.

2. Update the linear residual \( R = Z - H[Y_{P_0}, 0] \).

3. Solve the second stage equation \( Y = \texttt{gmres}(H, R) + [Y_{P_0}, 0] \) with preconditioner \( M_{2S}^{-1} \) to a specified tolerance.
Example 1: The full SPE10 benchmark problem with mortars in two-phase model

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

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

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

Ben Ganis | GJ Mortar Algorithms | bganis@ices.utexas.edu
Example 2: A multiscale problem on non-matching subdomain grids

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

2nd Stage: GMRES(1), $M_{2S}^{-1} = 5 \text{ Gauss-Seidel iterations.}$
Example 3: A heterogeneous 10M Cell Problem with mortars on 1024 Processors

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total time steps</td>
<td>1007</td>
</tr>
<tr>
<td>Total Newton iterations</td>
<td>1007</td>
</tr>
<tr>
<td>Total outer GMRES iterations</td>
<td>2449</td>
</tr>
<tr>
<td>Average GMRES iterations per Newton step</td>
<td>2.43</td>
</tr>
<tr>
<td>Average Newton iterations per time step</td>
<td>1.00</td>
</tr>
<tr>
<td>Total time step cuts</td>
<td>0</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix assembly time</td>
<td>86.04</td>
</tr>
<tr>
<td>Outer GMRES time</td>
<td>8459.16</td>
</tr>
<tr>
<td>Householder decoupling time</td>
<td>42.25</td>
</tr>
<tr>
<td>Pressure solve GMRES time</td>
<td>1394.55</td>
</tr>
<tr>
<td>Second stage GMRES time</td>
<td>3340.99</td>
</tr>
<tr>
<td>Mass lumping time</td>
<td>0.05</td>
</tr>
<tr>
<td>Matrix-matrix multiply time</td>
<td>1206.87</td>
</tr>
<tr>
<td>Total CPU time</td>
<td>8571.76</td>
</tr>
</tbody>
</table>

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, $M_{2S}^{-1} = M_{1S}^{-1}$. 

Ben Ganis | GJ Mortar Algorithms | bganis@ices.utexas.edu
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., 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!