Multilevel solvers with adaptive coarse space construction for lithosphere dynamics

Jed Brown\textsuperscript{1}, Mark Adams\textsuperscript{2}, Matt Knepley\textsuperscript{3}, Barry Smith\textsuperscript{1}

\textsuperscript{1}Mathematics and Computer Science Division, Argonne National Laboratory
\textsuperscript{2}Columbia University
\textsuperscript{3}University of Chicago

Frontiers in Computational Physics, 2012-12-19
The Great Solver Schism: Monolithic or Split?

Monolithic

- Direct solvers
- Coupled Schwarz
- Coupled Neumann-Neumann (need unassembled matrices)
- Coupled multigrid

\[ \text{X Need to understand local spectral and compatibility properties of the coupled system} \]

- Preferred data structures depend on which method is used.
- Interplay with geometric multigrid.

Split

- Physics-split Schwarz (based on relaxation)
- Physics-split Schur (based on factorization)
  - approximate commutators
  - SIMPLE, PCD, LSC
  - segregated smoothers
  - Augmented Lagrangian
  - “parabolization” for stiff waves

\[ \text{X Need to understand global coupling strengths} \]
Status quo for implicit solves in lithosphere dynamics

- global linearization using Newton or Picard
- assembly of a sparse matrix
- “block” factorization preconditioner with approximate Schur complement
- algebraic or geometric multigrid on positive-definite systems

Why is this bad?

- nonlinearities (e.g., plastic yield) are mostly local
  - feed back through nearly linear large scales
  - frequent visits to fine-scales even in nearly-linear regions
  - no way to locally update coarse grid operator
  - Newton linearization introduces anisotropy

- assembled sparse matrices are terrible for performance on modern hardware
  - memory bandwidth is very expensive compared to flops
  - fine-scale assembly costs a lot of memory
  - assembled matrices are good for algorithmic experimentation

- block preconditioners require more parallel communication
## Hardware Arithmetic Intensity

<table>
<thead>
<tr>
<th>Operation</th>
<th>Arithmetic Intensity (flops/B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse matrix-vector product</td>
<td>1/6</td>
</tr>
<tr>
<td>Dense matrix-vector product</td>
<td>1/4</td>
</tr>
<tr>
<td>Unassembled matrix-vector product</td>
<td>$\approx 8$</td>
</tr>
<tr>
<td>High-order residual evaluation</td>
<td>$&gt; 5$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Processor</th>
<th>BW (GB/s)</th>
<th>Peak (GF/s)</th>
<th>Balanced AI (F/B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E5-2670 8-core</td>
<td>35</td>
<td>166</td>
<td>4.7</td>
</tr>
<tr>
<td>Magny Cours 16-core</td>
<td>49</td>
<td>281</td>
<td>5.7</td>
</tr>
<tr>
<td>Blue Gene/Q node</td>
<td>43</td>
<td>205</td>
<td>4.8</td>
</tr>
<tr>
<td>Tesla M2090</td>
<td>120</td>
<td>665</td>
<td>5.5</td>
</tr>
<tr>
<td>Kepler K20Xm</td>
<td>160</td>
<td>1310</td>
<td>8.2</td>
</tr>
<tr>
<td>Xeon Phi</td>
<td>150</td>
<td>1248</td>
<td>8.3</td>
</tr>
</tbody>
</table>
Performance of assembled versus unassembled

- High order Jacobian stored unassembled using coefficients at quadrature points, can use local AD
- Choose approximation order at run-time, independent for each field
- Precondition high order using assembled lowest order method
- Implementation > 70% of FPU peak, SpMV bandwidth wall < 4%
**τ formulation of Full Approximation Scheme (FAS)**

- classical formulation: “coarse grid *accelerates* fine grid ↓ ↗
- τ formulation: “fine grid feeds back into coarse grid” ↗ ↘
- To solve $Nu = f$, recursively apply

  pre-smooth \[ \tilde{u}^h \leftarrow S^h_{\text{pre}}(u_0^h, f^h) \]

  solve coarse problem for $u^H$

  \[
  N^H u^H = I^H f^h + N^H \hat{I}^H \tilde{u}^h - I^H N^h \tilde{u}^h
  \]

  correction and post-smooth \[ u^h \leftarrow S^h_{\text{post}}(\tilde{u}^h + I^H (u^H - \hat{I}^H \tilde{u}^h), f^h) \]

  \[
  \begin{align*}
  I^H_h & \quad \text{residual restriction} \\
  \hat{I}^H_h & \quad \text{solution restriction} \\
  I^H & \quad \text{solution interpolation} \\
  f^H & \quad \text{restricted forcing} \\
  \{S^h_{\text{pre}}, S^h_{\text{post}}\} & \quad \text{smoothing operations on the fine grid}
  \end{align*}
  \]

- At convergence, $u^{H*} = \hat{I}^H_h u^{h*}$ solves the τ-corrected coarse grid equation $N^H u^H = f^H + \tau^H_h$, thus $\tau^H_h$ is the “fine grid feedback” that makes the coarse grid equation accurate.

- $\tau^H_h$ is *local* and need only be recomputed where it becomes stale.
Multiscale compression and recovery using $\tau$

- checkpoint converged coarse state
- recover using FMG anchored at $\ell_{cp} + 1$
- needs only $\ell_{cp}$ neighbor points
- $\tau$ correction is local

- Fine state $u^{h*}$ recovered *locally* from converged coarse state
  \[ u^{H*} = \hat{I}_h^H u^{h*} \]
- Normal multigrid cycles visit all levels moving from $n \rightarrow n + 1$
- FMG recovery only accesses levels finer than $\ell_{CP}$
- Only need neighborhood of desired region for decompression
- Lightweight checkpointing for transient adjoint computation
- Postprocessing applications, e.g., in-situ visualization at high temporal resolution in part of the domain
Four Schools of Thought for Multilevel Methods

- **Multigrid** (Brandt, Hackbusch, ...)
  - originally for resolved/asymptotic spatial discretizations
  - “textbook”: reach discretization error in one F-cycle
  - matrix-light/free, good for memory bandwidth
  - FAS well-developed for nonlinear problems

- **Multilevel Domain Decomposition** (Mandel, Dohrmann, Widlund)
  - leverage direct subdomain solvers, minimize communication
  - rapid coarsening $\kappa(P^{-1}A) \sim (1 + \log \frac{H}{h})^{2(L-1)}$
  - often formulated only as two-level methods, domain-conforming coefficients
  - lightly developed for nonlinear (e.g. ASPIN [Cai and Keyes])

- **Multiscale Finite Elements** (Babuska, Arbogast, ...)
  - local preprocessing to construct linear coarse operator
  - popular in porous media and composite materials (robust theory)

- **Equation-based multiscale models** (many)
  - Renormalization multigrid/systematic upscaling (Brandt)
    - interpolation, equilibration (compatible relaxation/Monte-Carlo), restriction
  - Heterogeneous multiscale method (E, Engquist)
    - reconstruction, constrained microscale simulation, data processing/compression
Computable Convergence Measures (Linear correction notation)

- Prolongation \( P : V_{\text{coarse}} \rightarrow V_{\text{fine}} \)
- Restriction \( R : V_{\text{fine}} \rightarrow V_{\text{coarse}} \)
- Smoother \( S^{-1} : V_{\text{fine}} \rightarrow V_{\text{fine}} \) should remove high-frequency component of error
- \( I - PR : V_{\text{fine}} \rightarrow V_{\text{fine}} \) removes part of vector visible in coarse space
- Error iteration \( I - M^{-1}A \), worst-case convergence factor is \( \lambda_{\text{max}} \)
- “Interpolation must be able to approximate an eigenvector with error bound proportional to the size of the associated eigenvalue.”

- Upper bound for convergence rate: \( \max_x \|x\|_{(I-PR)S(I-PR)}/\|x\|_A \)
- Distinct challenges to constructing coarse space and operator
  - Is the coarse space large enough to distinguish all low-energy modes?
  - Are those modes accurately represented? (Is \( P \) accurate enough?)
  - Is the coarse operator accurate? (Automatic with Galerkin-type \( RAP \) for nice problems.)
Compatible Relaxation

- Apply smoother subject to constraint $\hat{R}x = 0$
  1. $\tilde{x}_n = x_{n-1} + S^{-1}_A(r(x_{n-1}))$
  2. $x_n = \tilde{x}_n + S^{-1}_R(\hat{R}\tilde{x}_n)$
- Method to determine when coarse space is rich enough
- Slow to relax points/regions good candidates for coarse points/aggregates
- If subdomain solves used for smoothing, only interfaces are candidates

[Livne 2004]
Coarse basis functions

- \[ \| PRx \|_A + \| (I - PR)x \|_A \leq C \| x \|_A \]
- “decompose any \( x \) into parts without increasing energy much”
- near-null spaces must be represented exactly (partition of unity)
- number of rows of \( R \) determined already, usually \( P = R^T \)
- energy minimization with specified support [Wan, Chan, Smith; Mandel, Brezina, Vanek; Xu, Zikatanov]
- smoothed aggregation: \( P_{\text{smooth}} = (I - \omega D^{-1}A)P_{\text{agg}} \)
- classical AMG: each fine point processed independently
- domain decomposition/multiscale FEM: solve subdomain problems
Example: BDDC/FETI-DP coarse basis function

- only low-order continuity between subdomains
- corrected by more technical subdomain smoother

[Mandel and Sousedik 2010]
Why I like subdomain problems

- subassembly avoids explicit matrix triple product $A_{\text{coarse}} \leftarrow P^T A_{\text{fine}} P$
- can update the coarse operator locally (e.g. local nonlinearity)
- need not assemble entire fine grid operator
- if repetitive structure, need not store entire fine grid state
- can coarsen very rapidly (especially in smooth regions)
- lower communication setup phase

[Arbogast 2011]
Subdomain Interfaces and Energy Minimization

- minimize energy of all basis functions (columns of $P$) subject to
  - fixed compact support
  - partition of unity (near-null space)
- enforce partition of unity using Lagrange multipliers
  - $\lambda(x) = 0$ in coarse element interiors
  - means that globally optimal coarse basis functions are harmonic extensions of some interface values

[Xu and Zikatanov 2004]
Local edge/face-centered problems

\[ \mathbf{v}_e^{MD} = -a_e \nabla \phi_e^{MD} \quad \text{in } E_e, \]
\[ \nabla \cdot \mathbf{v}_e^{MD} = \pm |e|/|E_{e,i}| \quad \text{in } E_{e,i}, \ i = 1, 2, \]
\[ \mathbf{v}_e^{MD} \cdot \mathbf{v} = 0 \quad \text{on } \partial E_e. \]

- Arbogast’s multiscale dual-support elements for porous media
  - inconsistent for unaligned anisotropy
  - homogenization approach: upscale effective conductivity tensor from solution of periodic dual-support problem
- Dohrmann and Pechstein’s balancing domain decomposition for elasticity with unaligned coefficients
  - balance “torn” interface values \( u_{ie}, u_{je} \), written in terms of subdomain Schur complements
  - \( \bar{f}_e = S_{iee}u_{ie} + S_{jee}u_{je} \): sum of forces required along face \( e \) to displace subdomains \( i \) and \( j \) by \( u_{ie}, u_{je} \)
  - \( \bar{u}_e = (S_{iee} + S_{jee})^{-1}\bar{f}_e \): continuous displacement
  - equivalent to a (different) dual-support basis
Complication for saddle point problems

\[
\begin{pmatrix}
A & B^T \\
B & 0
\end{pmatrix}
\]

- want uniform stability for coarse problem
  - respect inf-sup condition, similar to fine grid
  - make coarse grid mimic fine grid \((Q_2 - P_{\text{disc}}^1)\)
- exact representation of volumetric mode
  - we can’t cheat on conservation while upscaling
  - naturally involves face integrals (inconvenient for recursive application)
  - obtain similar quantity through solution of inhomogeneous Stokes problems
- heuristic algebraic coarsening also possible [Adams 2004]
Nonlinear problems

- matrix-based smoothers require global linearization
- nonlinearity often more efficiently resolved locally
- nonlinear additive or multiplicative Schwarz
- nonlinear/matrix-free is good if

\[ C = \frac{\text{(cost to evaluate residual at one point)} \cdot N}{\text{(cost of global residual)}} \sim 1 \]

- finite difference: \( C < 2 \)
- finite volume: \( C \sim 2 \), depends on reconstruction
- finite element: \( C \sim \text{number of vertices per cell} \)

- larger block smoothers help reduce \( C \)
- additive correction like Jacobi reduces \( C \), but need to assemble corrector/scaling
Smoothing for saddle point systems

\[
\begin{pmatrix}
A & B^T \\
B & 0
\end{pmatrix}
\]

- pressure has no self-coupling
- pressure error modes not spectrally separated
- approaches
  - block smoothers (Vanka)
  - amplify fine-grid modes (distributive relaxation)
  - splitting with approximate Schur complement
Vanka block smoothers

- solve pressure-centered cell problems (better for discontinuous pressure)
- robust convergence factor $\sim 0.3$ if coarse grids are accurate
- 1D energy minimizing interpolants easy and effective
- can use assembled sparse matrices, but more efficient without
Changing Associativity: Distributive Smoothing

\[ PAx = Pb \quad APy = b, \quad x = Py \]

▶ Normal Preconditioning: make \( PA \) or \( AP \) well-conditioned
▶ Alternative: amplify high-frequency modes
  ▶ Multigrid smoothers only need to relax high-frequency modes
  ▶ Easier to do when spectrally separated: \( h \)-ellipticity
    ▶ pointwise smoothers (Gauss-Seidel) and polynomial/multistage methods
▶ Mechanics: form the product \( PA \) or \( AP \) and apply “normal” method
▶ Example (Stokes)

\[
A \sim \begin{pmatrix} -\nabla^2 & \nabla \\ \nabla & 0 \end{pmatrix} \quad P \sim \begin{pmatrix} 1 & -\nabla \\ 0 & -\nabla^2 \end{pmatrix} \quad AP \sim \begin{pmatrix} -\nabla^2 & "0" \\ \nabla & -\nabla^2 \end{pmatrix}
\]

▶ Convergence factor 0.32 (as good as Laplace) for smooth problems
Coupled MG for Stokes, split smoothers

\[ J = \begin{pmatrix} A & B^T \\ B & C \end{pmatrix} \]

\[ P_{\text{smooth}} = \begin{pmatrix} A_{\text{SOR}} & 0 \\ B & M \end{pmatrix} \]

-pc_type mg -pc_mg_levels 5 -pc_mg_galerkin
-mg_levels_pc_type fieldsplit
-mg_levels_pc_fieldsplit_block_size 3
-mg_levels_pc_fieldsplit_0_fields 0,1
-mg_levels_pc_fieldsplit_1_fields 2
-mg_levels_fieldsplit_0_pc_type sor
Outlook

- smoothing with point-block Jacobi Chebyshev and scaled diagonal for pressure
- needs only (subdomain “Neumann”) nonlinear function evaluations and assembly of point-block diagonal matrices
- convergence rates similar to smoothed aggregation, but without fine-grid assembly
- allows local updates of coarse operator, but currently slower due to naive implementation
- Development in progress within PETSc
  - parallel implementation of dual-support problems without duplicating lots of work
  - homogenization-based nonlinear coarsening
  - true $\tau$ formulation with adaptive fine-grid visits and partial coarse operator updates
  - microstructure-compatible pressure interpolation
  - “spectrally-correct” nonlinear saddle-point smoothers
  - locally-computable spectral estimates for guaranteed-stable additive smoothers