Multilevel solvers with adaptive coarse space construction for lithosphere dynamics

Jed Brown[1], Mark Adams[2], Matt Knepley[3], Barry Smith[1]

[1]Mathematics and Computer Science Division, Argonne National Laboratory
[2]Columbia University
[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

X Need to understand local spectral and compatibility properties of the coupled system

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

X Need to understand global coupling strengths

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

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?

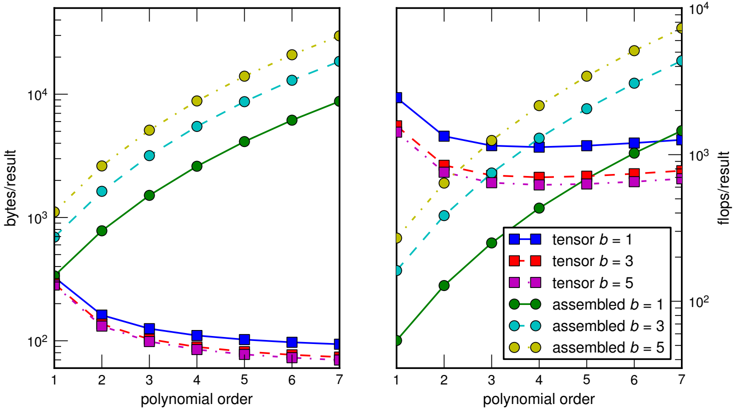
- nonlinearities (e.g., plastic yield) are mostly local
  - feed back through nearly linear large scales
  - frequent visits to fine-scales even in nearly-linear regions
  - no way to locally update coarse grid operator
  - Newton linearization introduces anisotropy
- assembled sparse matrices are terrible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

Operation	Arithmetic Intensity (flops/B)
Sparse matrix-vector product	1/6
Dense matrix-vector product	1/4
Unassembled matrix-vector product	≈ 8
High-order residual evaluation	> 5

Processor	BW (GB/s)	Peak (GF/s)	Balanced AI (F/B)
E5-2670 8-core	35	166	4.7
Magny Cours 16-core	49	281	5.7
Blue Gene/Q node	43	205	4.8
Tesla M2090	120	665	5.5
Kepler K20Xm	160	1310	8.2
Xeon Phi	150	1248	8.3

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

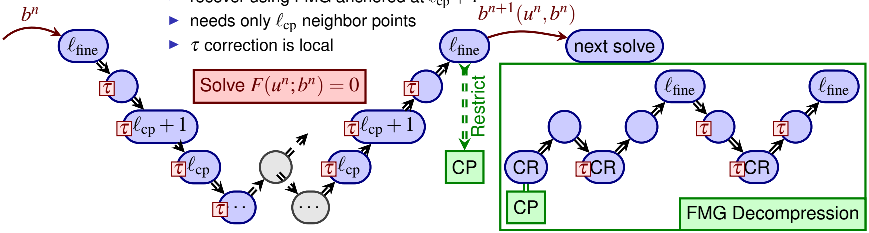
$$\begin{aligned}
\text{pre-smooth} \quad & \tilde{u}^h \leftarrow S^h_{\text{pre}}(u^h_0, f^h) \\
\text{solve coarse problem for } u^H \quad & N^H u^H = \underbrace{I^H_h f^h}_{f^H} + \underbrace{N^H \hat{I}^H_h \tilde{u}^h - I^H_h N^h \tilde{u}^h}_{\tau^H_h} \\
\text{correction and post-smooth} \quad & u^h \leftarrow S^h_{\text{post}}\left(\tilde{u}^h + I^h_H(u^H - \hat{I}^H_h \tilde{u}^h), f^h\right)
\end{aligned}$$

I^H_h	residual restriction	$\hat{I}^H_h$	solution restriction
I^h_H	solution interpolation	$f^H = I^H_h f^h$	restricted forcing
$\{S^h_{\text{pre}}, S^h_{\text{post}}\}$	smoothing operations on the fine grid		

- 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 τ^H_h is the “fine grid feedback” that makes the coarse grid equation accurate.
- τ^H_h is *local* and need only be recomputed where it becomes stale.

Multiscale compression and recovery using τ

- checkpoint converged coarse state
- recover using FMG anchored at $\ell_{\text{cp}}+1$
- needs only ℓ_{cp} neighbor points
- τ 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 $n \to n+1$
- FMG recovery only accesses levels finer than ℓ_{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

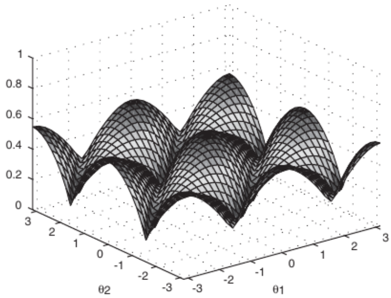
Four Schools of Thought for Multilevel Methods

- 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 $\kappa(P^{-1}A) \sim \left(1+\log\frac{H}{h}\right)^{2(L-1)}$
  - 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, equilibriation (compatible relaxation/Monte-Carlo), restriction
  - Heterogeneous multiscale method (E, Engquist)
    - reconstruction, constrained microscale simulation, data processing/compression

Computable Convergence Measures (Linear correction notation)

- Prolongation $P : V_{\text{coarse}} \to V_{\text{fine}}$
- Restriction $R : V_{\text{fine}} \to V_{\text{coarse}}$
- Smoother $S^{-1} : V_{\text{fine}} \to V_{\text{fine}}$ should remove high-frequency component of error
- $I - PR : V_{\text{fine}} \to V_{\text{fine}}$ removes part of vector visible in coarse space
- Error iteration $I - M^{-1}A$, worst-case convergence factor is $\lambda_{\max}$
- "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



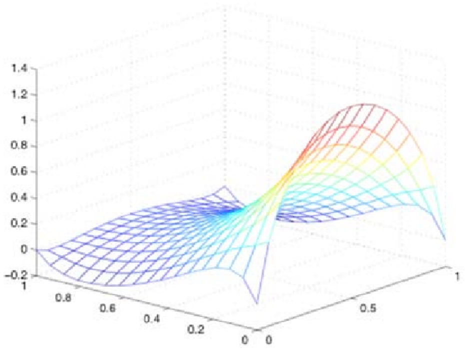
[Livne 2004]

- Apply smoother subject to constraint $\hat{R}x = 0$
  1. $\tilde{x}_n = x_{n-1} + S_A^{-1}\left(r(x_{n-1})\right)$
  2. $x_n = \tilde{x}_n + S_R^{-1}\left(\hat{R}\tilde{x}_n)\right)$
- 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

Coarse basis functions

- $\|PRx\|_A + \|(I - PR)x\|_A \le C \|x\|_A$
- “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

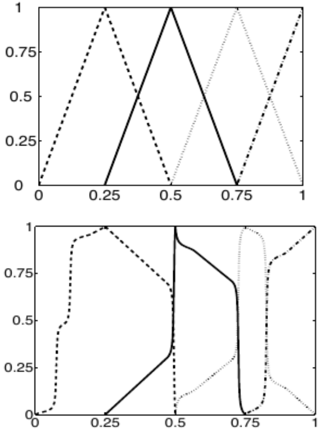
Example: BDDC/FETI-DP coarse basis function



[Mandel and Sousedik 2010]

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

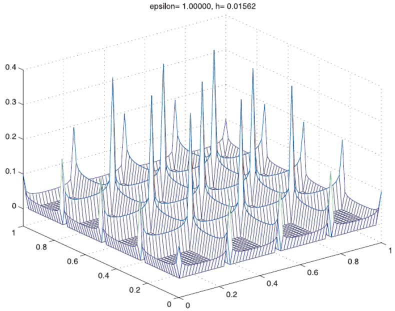
Why I like subdomain problems



[Arbogast 2011]

- 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

Subdomain Interfaces and Energy Minimization

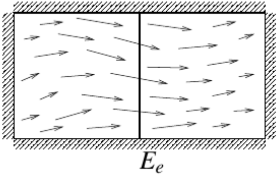


[Xu and Zikatanov 2004]

- 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

Local edge/face-centered problems

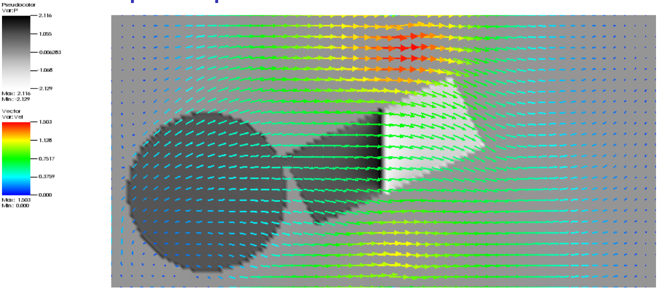
$$\begin{aligned} \mathbf{v}_e^{\mathrm{MD}} &= -a_\varepsilon \nabla \phi_e^{\mathrm{MD}} && \text{in } E_e, \\ \nabla \cdot \mathbf{v}_e^{\mathrm{MD}} &= \pm |e|/|E_{e,i}| && \text{in } E_{e,i},\ i = 1,2, \\ \mathbf{v}_e^{\mathrm{MD}} \cdot \nu &= 0 && \text{on } \partial E_e. \end{aligned}$$



- 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
  - $\overline{f}_e = S_{iee} u_{ie} + S_{jee} u_{je}$: sum of forces required along face e to displace subdomains i and j by u_{ie}, u_{je}
  - $\overline{u}_e = (S_{iee} + S_{jee})^{-1} \overline{f}_e$: continuous displacement
  - equivalent to a (different) dual-support basis

Complication for saddle point problems

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



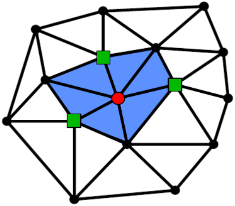
- want uniform stability for coarse problem
  - respect inf-sup condition, similar to fine grid
  - make coarse grid mimic fine grid ($Q_2 - P_1^{\text{disc}}$)
- *exact* representation of volumetric mode
  - we can't cheat on conservation while upscaling
  - naturally involves face integrals (inconvenient 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: $C \sim$ number of vertices per cell
- larger block smoothers help reduce C
- additive correction like Jacobi reduces C, but need to assemble corrector/scaling

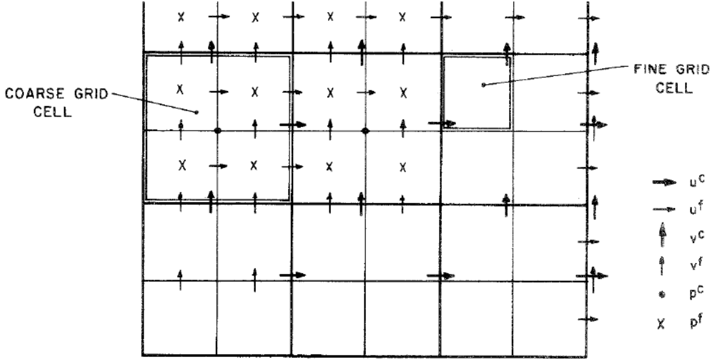


Smoothing for saddle point systems

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

- 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 ~ 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

$$PAx = Pb \qquad\qquad APy = b, \quad x = Py$$

- 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)

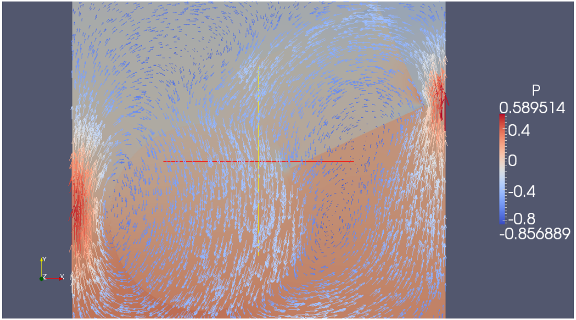
$$A \sim \begin{pmatrix} -\nabla^2 & \nabla \\ \nabla\cdot & 0 \end{pmatrix} \quad P \sim \begin{pmatrix} 1 & -\nabla \\ 0 & -\nabla^2 \end{pmatrix} \quad AP \sim \begin{pmatrix} -\nabla^2 & \text{“0”} \\ \nabla\cdot & -\nabla^2 \end{pmatrix}$$

- 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 τ 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