Low-Communication Multigrid, with Applications to Time-Dependent Adjoints, in-Situ Visualization, and Resilience

Jed Brown jedbrown@mcs.anl.gov (ANL and CU Boulder)
Mark Adams (LBL), Matt Knepley (UChicago)

SIAM PP, 2014-02-19

Plan: ruthlessly eliminate communication

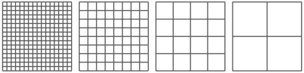
Why?

- Local recovery despite global coupling
- Tolerance for high-frequency load imbalance
  - From irregular computation or hardware error correction
- More scope for dynamic load balance

Requirements

- Must retain optimal convergence with good constants
- Flexible, robust, and debuggable

Multigrid Preliminaries



Multigrid is an $O(n)$ method for solving algebraic problems by defining a hierarchy of scale. A multigrid method is constructed from:

1. a series of discretizations
   - coarser approximations of the original problem
   - constructed algebraically or geometrically
2. intergrid transfer operators
   - residual restriction I_h^H (fine to coarse)
   - state restriction $\hat{I}_h^H$ (fine to coarse)
   - partial state interpolation I_H^h (coarse to fine, 'prolongation')
   - state reconstruction $\mathbb{I}_H^h$ (coarse to fine)
3. Smoothers (S)
   - correct the high frequency error components
   - Richardson, Jacobi, Gauss-Seidel, etc.
   - Gauss-Seidel-Newton or optimization methods

τ formulation of Full Approximation Scheme (FAS)

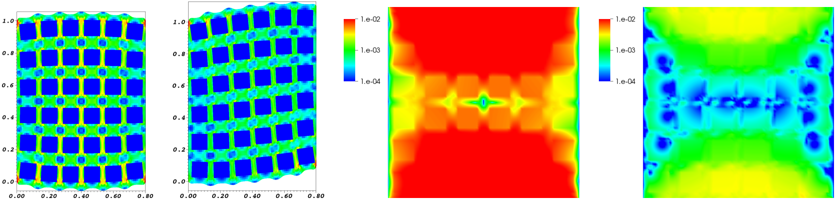
- classical formulation: "coarse grid *accelerates* fine grid $\searrow\nearrow$
- τ formulation: "fine grid feeds back into coarse grid" $\nearrow\searrow$
- To solve $Nu = f$, recursively apply

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

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.

τ corrections

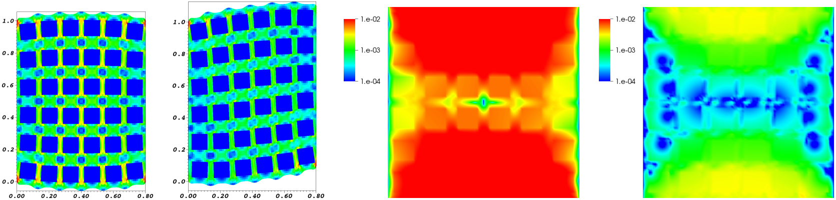


- Plane strain elasticity, $E = 1000, \nu = 0.4$ inclusions in $E = 1, \nu = 0.2$ material, coarsen by 3^2.
- Solve initial problem everywhere and compute $\tau_h^H = A^H \hat{I}_h^H u^h - I_h^H A^h u^h$
- Change boundary conditions and solve FAS coarse problem

$$N^H \acute{u}^H = \underbrace{I_h^H \acute{f}^h}_{\acute{f}^H} + \underbrace{N^H \hat{I}_h^H \tilde{u}^h - I_h^H N^h \tilde{u}^h}_{\tau_h^H}$$

- Prolong, post-smooth, compute error $e^h = \acute{u}^h - (N^h)^{-1} \acute{f}^h$
- Coarse grid *with* τ is nearly $10\times$ better accuracy

τ corrections



- Plane strain elasticity, $E = 1000, \nu = 0.4$ inclusions in $E = 1, \nu = 0.2$ material, coarsen by 3^2.
- Solve initial problem everywhere and compute $\tau_h^H = A^H \hat{I}_h^H u^h - I_h^H A^h u^h$
- Change boundary conditions and solve FAS coarse problem

$$N^H \acute{u}^H = \underbrace{I_h^H \acute{f}^h}_{\acute{f}^H} + \underbrace{N^H \hat{I}_h^H \tilde{u}^h - I_h^H N^h \tilde{u}^h}_{\tau_h^H}$$

- Prolong, post-smooth, compute error $e^h = \acute{u}^h - (N^h)^{-1} \acute{f}^h$
- Coarse grid *with* τ is nearly $10\times$ better accuracy

τ adaptivity for heterogeneous media

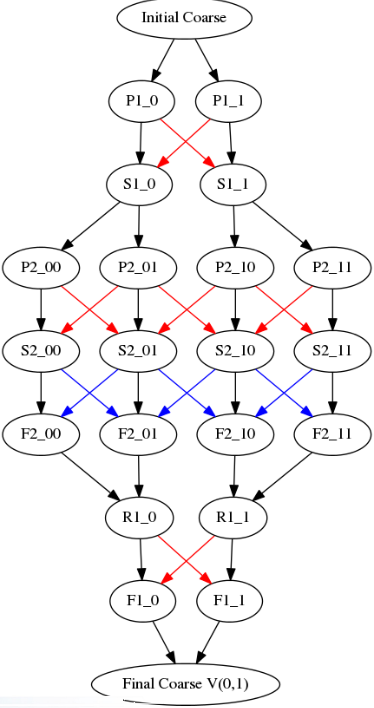
- Applications
  - Geo: reservoir engineering, lithosphere dynamics (subduction, rupture/fault dynamics)
  - carbon fiber, biological tissues, fracture
  - Conventional adaptivity fails
- Traditional adaptive methods fail
  - Solutions are not “smooth”
  - Cannot build accurate coarse space without scale separation
- τ adaptivity
  - Fine-grid work needed everywhere at first
  - Then τ becomes accurate in nearly-linear regions
  - Only visit fine grids in “interesting” places: active nonlinearity, drastic change of solution

Comparison to nonlinear domain decomposition

- ASPIN (Additive Schwarz preconditioned inexact Newton)
  - Cai and Keyes (2003)
  - More local iterations in strongly nonlinear regions
  - Each nonlinear iteration only propagates information locally
  - Many real nonlinearities are activated by long-range forces
    - locking in granular media (gravel, granola)
    - binding in steel fittings, crack propagation
  - Two-stage algorithm has different load balancing
    - Nonlinear subdomain solves
    - Global linear solve
- τ adaptivity
  - Minimum effort to communicate long-range information
  - Nonlinearity sees effects as accurate as with global fine-grid feedback
  - Fine-grid work always proportional to “interesting” changes

Low communication MG

- red arrows can be removed by τ-FAS with overlap
- blue arrows can also be removed, but then algebraic convergence stalls when discretization error is reached
- no simple way to check that discretization error is obtained
- if fine grid state is not stored, use compatible relaxation to complete prolongation P

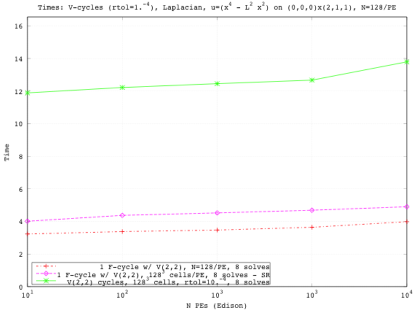


Segmental refinement: no horizontal communication

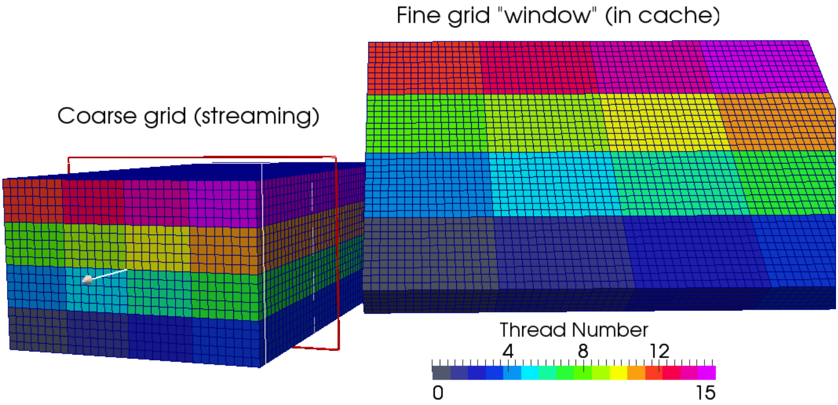
- 512^3 cells/process; 128 processes
- 27-point second-order stencil, manufactured analytic solution
- 5 SR levels: 16^3 cells/process local coarse grid
- Overlap $=$ Base $+ (L - \ell)$Increment
  - Implementation requires even number of cells—round down.
- FMG with $V(2,2)$ cycles

Table : $\|e_{SR}\|_\infty / \|e_{FMG}\|_\infty$

	Base		
Increment	1	2	3
1	1.59	2.34	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00



Reducing memory bandwidth



- Sweep through “coarse” grid with moving window
- Zoom in on new slab, construct fine grid “window” in-cache
- Interpolate to new fine grid, apply pipelined smoother (*s*-step)
- Compute residual, accumulate restriction of state and residual into coarse grid, expire slab from window

Arithmetic intensity of sweeping visit

- Assume 3D cell-centered, 7-point stencil
- 14 flops/cell for second order interpolation
- $\geq$ 15 flops/cell for fine-grid residual or point smoother
- 2 flops/cell to enforce coarse-grid compatibility
- 2 flops/cell for plane restriction
- assume coarse grid points are reused in cache
- Fused visit reads u^H and writes $\hat{I}_h^H u^h$ and $I_h^H r^h$
- Arithmetic Intensity

$$\frac{\overbrace{15}^{\text{interp}} + \overbrace{2\cdot(15+2)}^{\text{compatible relaxation}} + \overbrace{2\cdot 15}^{\text{smooth}} + \overbrace{15}^{\text{residual}} + \overbrace{2}^{\text{restrict}}}{3\cdot \texttt{sizeof(scalar)} / \underbrace{2^3}_{\text{coarsening}}} \gtrsim 30 \tag{1}$$

- Still $\gtrsim$ 10 with non-compressible fine-grid forcing

Regularity

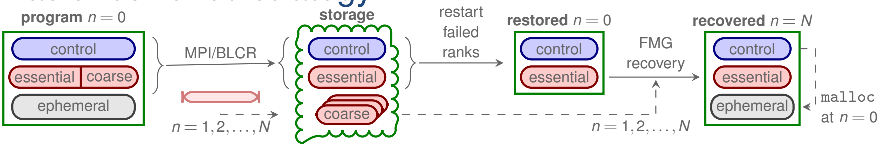
Accuracy of recovery depends on operator regularity

- Even with regularity, we can only converge up to discretization error, unless we add a *consistent* fine-grid residual evaluation
- Visit fine grid with some overlap, but patches do not agree exactly in overlap
- Need decay length for high-frequency error components (those that restrict to zero) that is bounded with respect to grid size
- Required overlap J is proportional to the number of cells to cover decay length
- Can enrich coarse space along boundary, but causes loss of coarse-grid sparsity
- Brandt and Diskin (1994) has two-grid LFA showing $J \lesssim 2$ is sufficient for Laplacian
- With L levels, overlap $J(k)$ on level k,

$$2J(k) \geq s(L-k+1)$$

where s is the smoothness order of the solution or the discretization order (whichever is smaller)

Basic resilience strategy



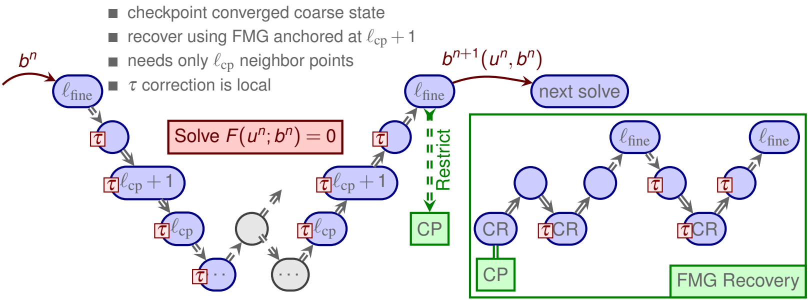
control: contains program stack, solver configuration, etc.

essential: program state that cannot be easily reconstructed: time-dependent solution, current optimization/bifurcation iterate

ephemeral: easily recovered structures: assembled matrices, preconditioners, residuals, Runge-Kutta stage solutions

- Essential state at time/optimization step n is inherently globally coupled to step $n-1$ (otherwise we could use an explicit method)
- *Coarse* level checkpoints are orders of magnitude smaller, but allow rapid recovery of essential state
- FMG recovery needs only nearest neighbors

Multiscale compression and recovery using τ form



- Normal multigrid cycles visit all levels moving from $n \to n+1$
- FMG recovery only accesses levels finer than ℓ_{CP}
- Only failed processes and neighbors participate in recovery
- Lightweight checkpointing for transient adjoint computation
- Postprocessing applications, e.g., in-situ visualization at high temporal resolution in part of the domain

First-order cost model for FAS resilience

Extend first-order locality-unaware model of Young (1974):

- t_W time to write a heavy fine-grid checkpointed state
- t_R time to read back lost state
- R fraction of forward simulation needed for recomputation from a saved state
- P the heavy checkpoint interval
- M mean time to failure

Neglect cost of I/O for lightweight coarse-grid checkpoints

$$\text{Overhead} = 1 - \text{AppUtilization} = \underbrace{\frac{t_\mathrm{W}}{P}}_{\text{writing}} + \underbrace{\frac{t_\mathrm{R}}{M}}_{\text{reading after failure}} + \underbrace{\frac{RP}{2M}}_{\text{recomputation}}$$

Minimized for a heavy checkpointing interval $P = \sqrt{2Mt_\mathrm{W}/R}$

$$\text{Overhead}^* = \sqrt{2t_\mathrm{W}R/M} + t_\mathrm{R}/M$$

where the first term is always larger than the second. Conventional checkpointing schemes store only fine-grid state, thus $R = 1$ (recovery costs the same as initial computation).

Other uses

- Transient adjoints
  - Adjoint model runs backward-in-time, needs state from solution of forward model
  - Status quo: hierarchical checkpointing
  - Memory-constrained and requires computing forward model multiple times
  - If forward model is stiff, each step has global dependence
  - Compression via τ-FAS accelerates recomputation, can be local
- Visualization and analysis
  - Targeted visualization in small part of domain
  - Interesting features emergent so can't predict where to look

Outlook

- System software challenge: Partial restart interfaces
  - Low-overhead recruitment protocol via `MPI_Ibarrier`
  - Can we “reconnect” MPI communicators?
  - Can a library return a solution on a different communicator?
- τ-FAS adaptivity and compression
  - Coarse-centric restructuring is a major interface change
  - Worthwhile for resilience and to better use hardware
  - Nonlinear smoothers (and discretizations)
  - Dynamic load balancing
  - Reliability of error estimates for refreshing τ