Adaptive multigrid methods for heterogeneous problems

Scott MacLachlan

Department of Computer Science and Engineering,
University of Minnesota
maclach@cs.umn.edu

June 21,2006

Target Applications

- Fluid flow in porous media
  - Highly heterogeneous media
  - Interested in global properties of the solution
- Coupled fluid-elastic systems
  - Multiple material regimes
  - Different models require different treatment
- Lattice quantum chromodynamics
  - Highly heterogeneous operator
  - Randomized heterogeneity within Monte Carlo process

Modelling Heterogeneity

Two important considerations:

1. Capturing relevant features of continuum model
2. Solver efficiency

We'll assume Step 1 has been taken care of

Focus on efficient solvers for heterogeneous discrete models

- Large problem sizes
- Large condition numbers
- Multiscale structure of operator

Solving Homogeneous Problems

Heterogeneity is an added complication, but not fundamental

Still need techniques to handle

- Large problem sizes
- Large condition numbers
- Multiscale structure of operator

Solving Homogeneous Problems

Heterogeneity is an added complication, but not fundamental

Still need techniques to handle

- Large problem sizes
- Large condition numbers
- Multiscale structure of operator

These features are present even in homogeneous problems

- Consider solution strategy for homogeneous models
  - Geometric/Algebraic multigrid
- Look for where heterogeneity plays a role

Stationary Iterative Methods

- Want to improve approximation, $x^{(0)}$, to $x = A^{-1}b$
- Residual, $r^{(0)}$, is a measure of the error

$$r^{(0)} = b - Ax^{(0)} = Ax - Ax^{(0)} = A(x - x^{(0)})$$

- Choose $B^{-1} \approx A^{-1}$
- Take $x^{(1)} = x^{(0)} + B^{-1}r^{(0)}$

Error propagation form: $e^{(1)} = (I - B^{-1}A)e^{(0)}$

Stationary Iterative Methods

- Want to improve approximation, $x^{(0)}$, to $x = A^{-1}b$
- Residual, $r^{(0)}$, is a measure of the error

$$r^{(0)} = b - Ax^{(0)} = Ax - Ax^{(0)} = A(x - x^{(0)})$$

- Choose $B^{-1} \approx A^{-1}$
- Take $x^{(1)} = x^{(0)} + B^{-1}r^{(0)}$

Error propagation form:
$$\begin{aligned} e^{(1)} &= (I - B^{-1}A)e^{(0)} \\ e^{(2)} &= (I - B^{-1}A)e^{(1)} \end{aligned}$$

Stationary Iterative Methods

- Want to improve approximation, $x^{(0)}$, to $x = A^{-1}b$
- Residual, $r^{(0)}$, is a measure of the error

$$r^{(0)} = b - Ax^{(0)} = Ax - Ax^{(0)} = A(x - x^{(0)})$$

- Choose $B^{-1} \approx A^{-1}$
- Take $x^{(1)} = x^{(0)} + B^{-1}r^{(0)}$

Error propagation form:
$$e^{(1)} = (I - B^{-1}A)e^{(0)}$$
$$e^{(2)} = (I - B^{-1}A)^2e^{(0)}$$

Stationary Iterative Methods

- Want to improve approximation, $x^{(0)}$, to $x = A^{-1}b$
- Residual, $r^{(0)}$, is a measure of the error

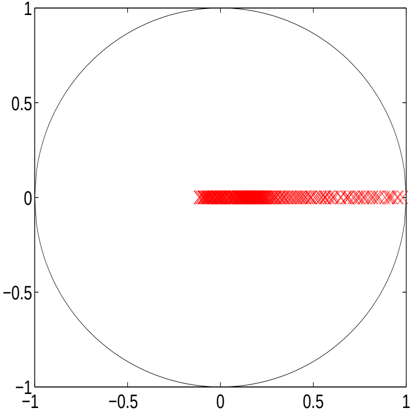
$$r^{(0)} = b - Ax^{(0)} = Ax - Ax^{(0)} = A(x - x^{(0)})$$

- Choose $B^{-1} \approx A^{-1}$
- Take $x^{(1)} = x^{(0)} + B^{-1}r^{(0)}$

Error propagation form:
$$\begin{aligned} e^{(1)} &= (I - B^{-1}A)e^{(0)} \\ e^{(2)} &= (I - B^{-1}A)^2 e^{(0)} \\ &\vdots \\ e^{(n)} &= (I - B^{-1}A)^n e^{(0)} \end{aligned}$$

Convergence of Stationary Iterations

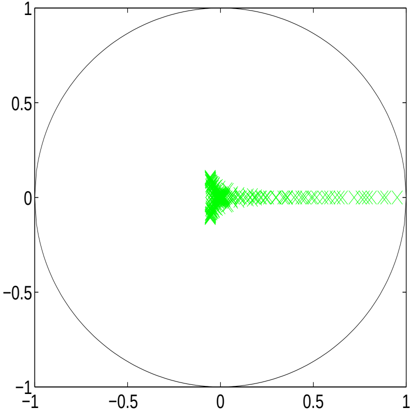
Convergence depends on spectrum of $I - B^{-1}A$



Weighted Jacobi Iteration: $e^{(n)} = (I - \frac{4}{3}D^{-1}A)^n e^{(0)}$

Convergence of Stationary Iterations

Convergence depends on spectrum of $I - B^{-1}A$



Gauss-Seidel Iteration: $e^{(n)} = (I - L^{-1}A)^n e^{(0)}$

Failing in a Structured Way

Small $B^{-1}A$-Rayleigh quotients cause trouble

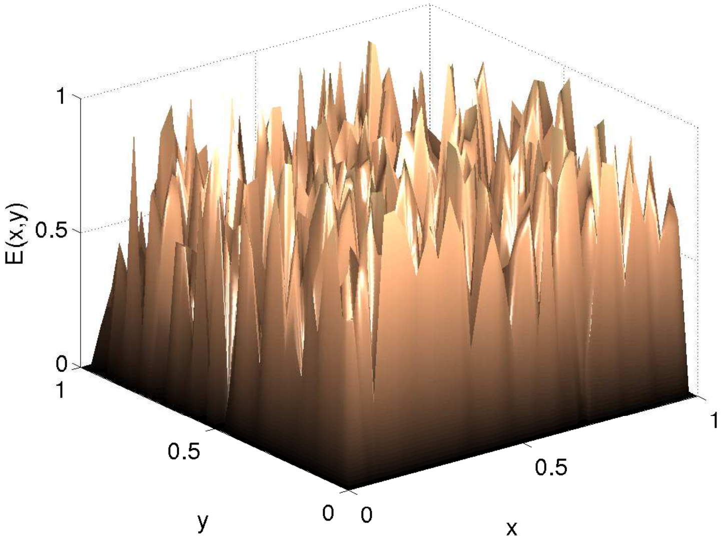
$$\lambda_{\max}(I - B^{-1}A) = 1 - \min_y \frac{y^T A y}{y^T B y}$$

For simple B, equivalent to small A-Rayleigh quotients

$$\frac{y^T A y}{y^T B y} = \left(\frac{y^T A y}{y^T y}\right)\left(\frac{y^T y}{y^T B y}\right)$$

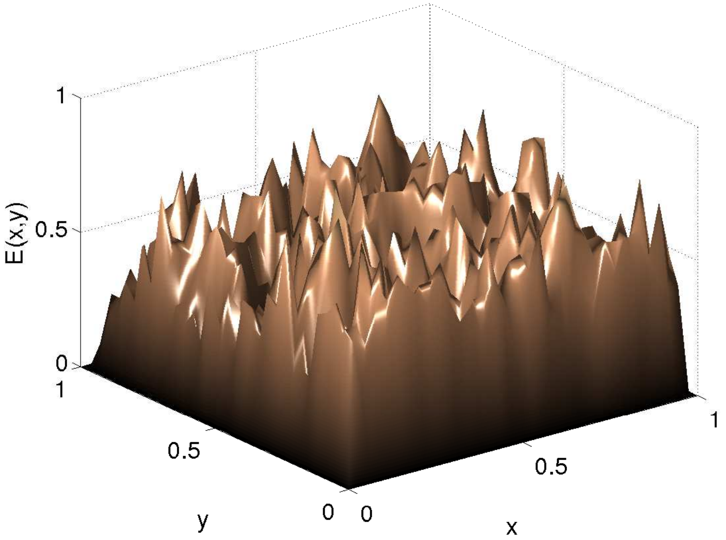
Can we use this to our advantage?

Smoothing Property



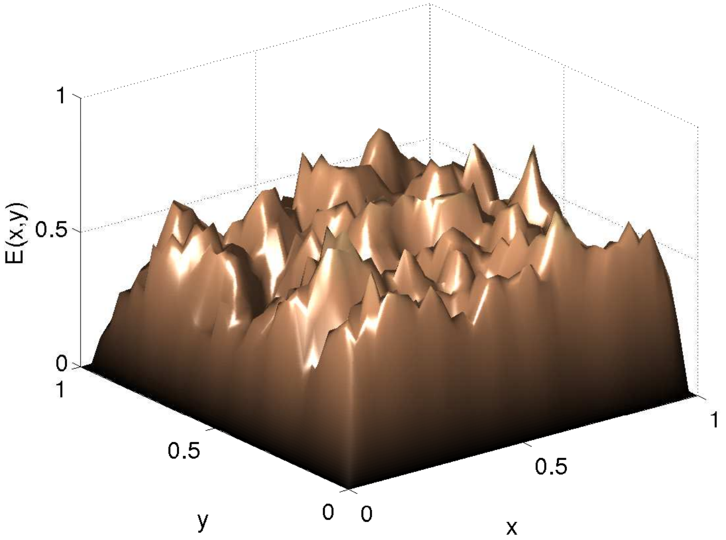
Random initial error

Smoothing Property



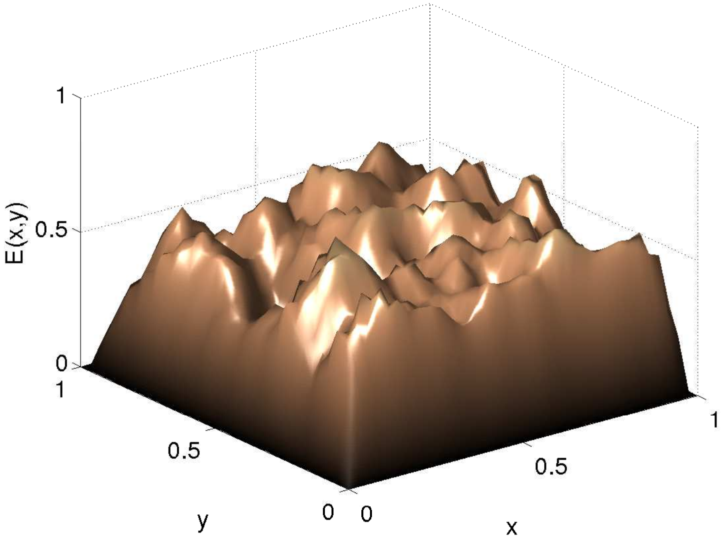
Error after 1 weighted Jacobi iteration

Smoothing Property



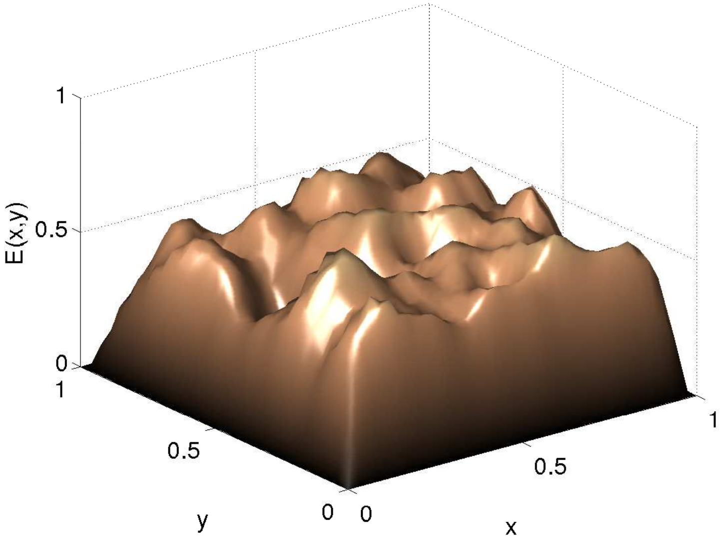
Error after 2 weighted Jacobi iterations

Smoothing Property



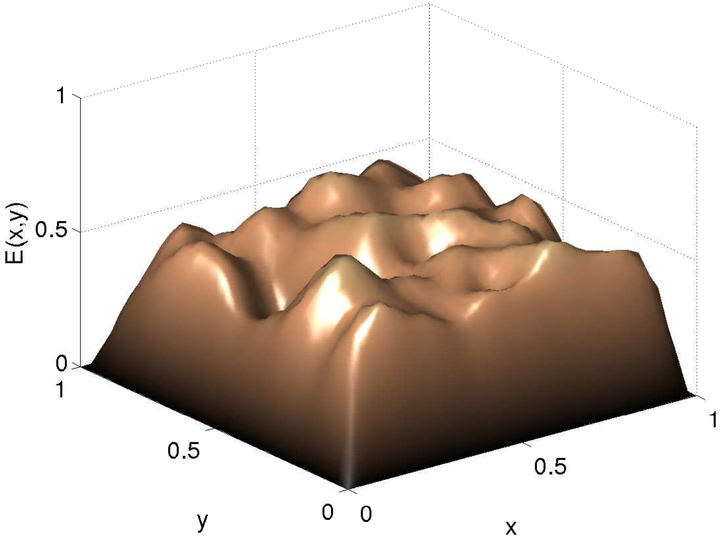
Error after 3 weighted Jacobi iterations

Smoothing Property



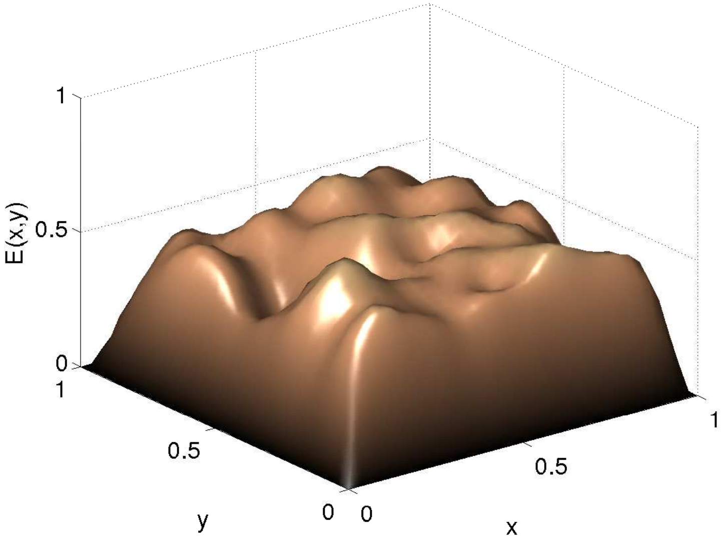
Error after 4 weighted Jacobi iterations

Smoothing Property



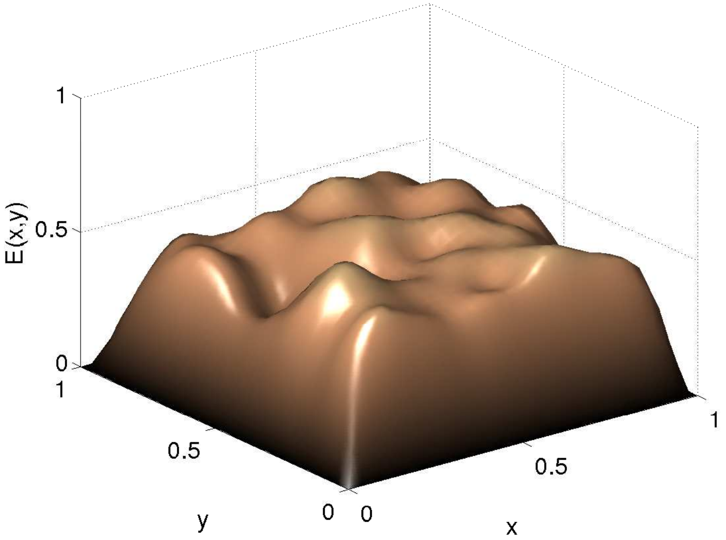
Error after 5 weighted Jacobi iterations

Smoothing Property



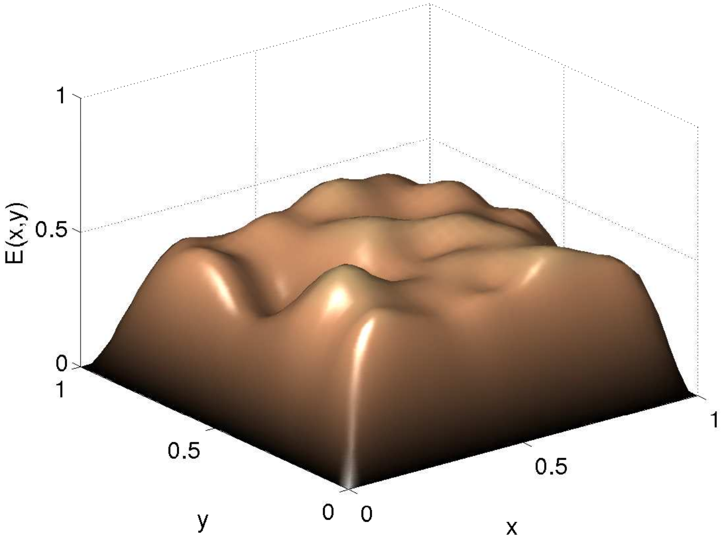
Error after 6 weighted Jacobi iterations

Smoothing Property



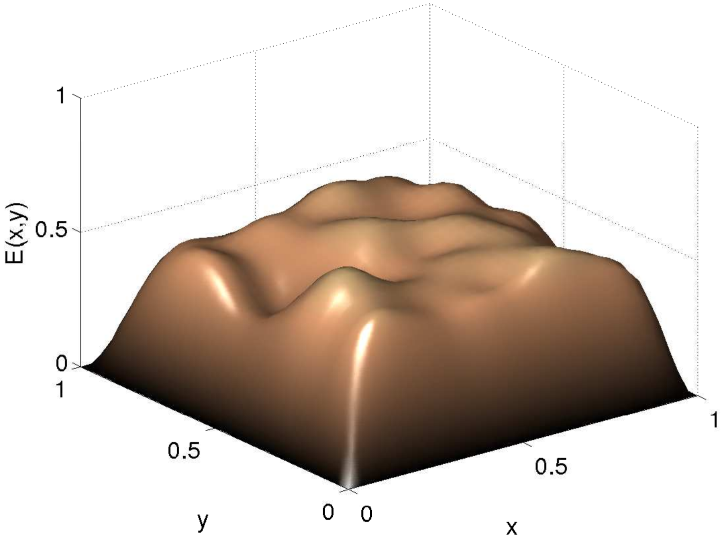
Error after 7 weighted Jacobi iterations

Smoothing Property



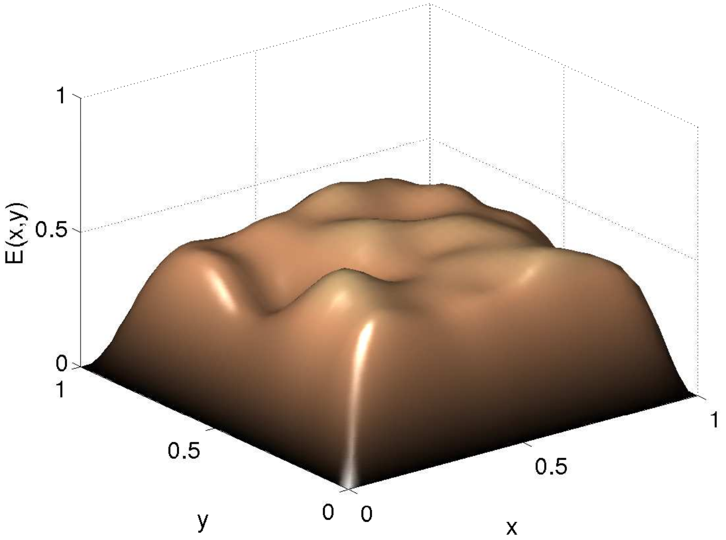
Error after 8 weighted Jacobi iterations

Smoothing Property



Error after 9 weighted Jacobi iterations

Smoothing Property



Error after 10 weighted Jacobi iterations

Complementarity

- Error after a few weighted Jacobi iterations has structure
- Instead of throwing out the method, look to complement its failings

How can we best correct error modes that are slow to be reduced by relaxation?

Complementarity

- Error after a few weighted Jacobi iterations has structure
- Instead of throwing out the method, look to complement its failings

How can we best correct error modes that are slow to be reduced by relaxation?

- Slow-to-converge errors are smooth
- Smooth vectors can be easily represented using fewer degrees of freedom

Coarse-Grid Correction

- Smooth vectors can be accurately represented using fewer degrees of freedom
- Idea: transfer job of resolving smooth components to a coarser grid version of the problem
- Need:
  - Complementary process for resolving smooth components of the error on the coarse grid
  - Way to combine the results of the two processes

Variational Coarsening

- Correct the approximation after relaxation, $x^{(1)}$, from an auxilliary (coarse-grid) problem
- Need interpolation map, P, from coarse grid to fine grid
- Corrected approximation will be $x^{(2)} = x^{(1)} + Px_c$

What is the *best* x_c for correction?

A-norm and A-inner product

- Asking for the *best* solution implies a metric
- Symmetric and positive-definite matrix, A, defines an inner product and a norm:

$$\langle x, y\rangle_A = y^T A x \quad \text{and} \quad \|x\|_A^2 = x^T A x$$

- *Best* then means closest to the exact solution in norm

$$y^\star = \operatorname*{argmin}_y \|x - y\|_A$$

Variational Coarsening

- Want to correct the approximation after relaxation, $x^{(1)}$, from a coarse-grid version of of the problem
- Need interpolation map, P, from coarse grid to fine grid
- Corrected approximation will be $x^{(2)} = x^{(1)} + Px_c$

What is the *best* x_c for correction?

- *Best* means closest to the exact solution in norm

$$x_c = \underset{y_c}{\operatorname{argmin}} \|x - (x^{(1)} + Py_c)\|_A$$

- *Best* x_c satisfies $(P^T AP)x_c = P^T A(x - x^{(1)}) = P^T r^{(1)}$

Multigrid

Multigrid Components

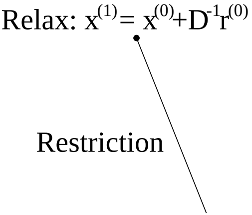
Relax: $x^{(1)} = x^{(0)} + D^{-1} r^{(0)}$

- Relaxation

- Use a smoothing process (such as Jacobi or Gauss-Seidel) to eliminate oscillatory errors
- Remaining error satisfies $Ae^{(1)} = r^{(1)} = b - Ax^{(1)}$

Multigrid

Multigrid Components

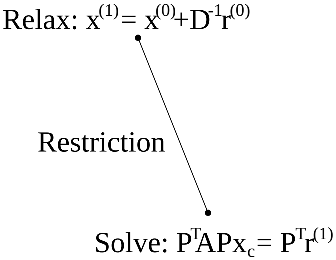


- Relaxation
- Restriction
- Transfer residual to coarse grid
- Compute $P^T r^{(1)}$

Multigrid

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction



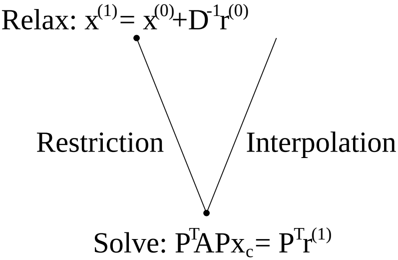
- Use coarse-grid correction to eliminate smooth errors
- Best correction, x_c, in terms of A-norm satisfies

$$P^T A P x_c = P^T r^{(1)}$$

Multigrid

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation

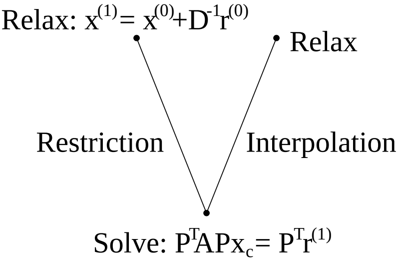


- Transfer correction to fine grid
- Compute $x^{(2)} = x^{(1)} + Px_c$

Multigrid

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation
- Relaxation

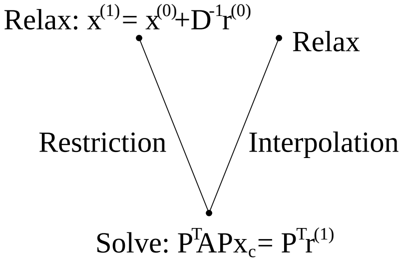


- Relax once again to remove oscillatory error introduced in coarse-grid correction

Multigrid

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation
- Relaxation



Direct solution of coarse-grid problem isn't practical

Recursion!

Apply same methodology to solve coarse-grid problem

Algebraic Picture

On any level, error reduced by

1. Relaxation
2. Coarse-grid correction

Coarse-grid correction treats errors in Range(P)

- Range(P) must include errors for which relaxation is slow
- Relaxation must be effective on Range(P)$^\perp$

$$\text{Domain}(A) = \text{Range}(P) \oplus \text{Range}(P)^\perp$$

Assumptions on Interpolation

- Error after relaxation on Poisson's equation is smooth
  - Low-order geometric interpolation accurate

Classical geometric multigrid defines interpolation based on

- grid geometry
- operator properties
- assumptions on performance of relaxation

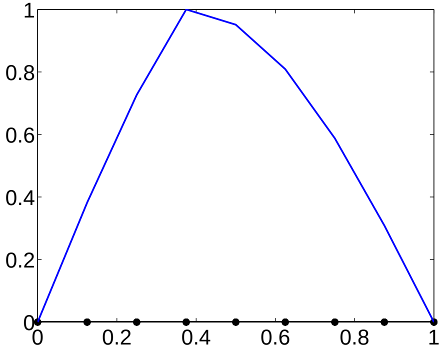
Heterogeneity strongly influences performance of relaxation

"Smooth" Errors

- Linear interpolation can make O(1) errors for problems with non-smooth coefficients

Slowest to converge error for $\frac{d}{dx}\left(\sigma \frac{du}{dx}\right)$, for

$$\sigma = \begin{cases} 10^{-8} & x \leq \frac{3}{8} \\ 1 & x > \frac{3}{8} \end{cases}$$



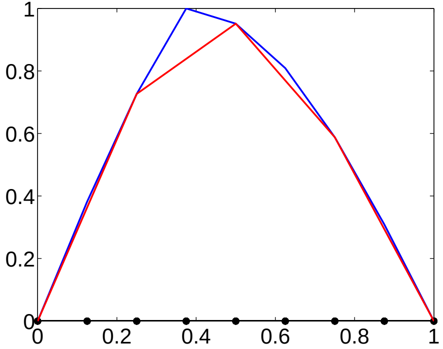
"Smooth" Errors

- Linear interpolation can make O(1) errors for problems with non-smooth coefficients

Slowest to converge error for $\frac{d}{dx}\left(\sigma \frac{du}{dx}\right)$, for

$$\sigma = \begin{cases} 10^{-8} & x \leq \frac{3}{8} \\ 1 & x > \frac{3}{8} \end{cases}$$

and linear interpolant from coarse grid



"Smooth" Errors

- Linear interpolation can make O(1) errors for problems with non-smooth coefficients
- The abrupt change in character of slow-to-converge errors is reflected in matrix entries

$$A = \frac{1}{h^2}\begin{bmatrix} 2\times 10^{-8} & -10^{-8} & & & & & \\ -10^{-8} & 2\times 10^{-8} & -10^{-8} & & & & \\ & -10^{-8} & 1+10^{-8} & -1 & & & \\ & & -1 & 2 & -1 & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix}$$

“Smooth” Errors

- Linear interpolation can make O(1) errors for problems with non-smooth coefficients
- The abrupt change in character of slow-to-converge errors is reflected in matrix entries
- Idea: Use the entries in the matrix operator to help define interpolation

Algebraic Multigrid Interpolation

- Assume a partition into fine (F) and coarse (C) grid sets
- Define interpolation based only on entries in A
- Start with assumption that errors left after relaxation have small residuals: for $i \in F$,

$$(Ae)_i \approx 0$$

$$a_{ii}e_i = -\sum_{j \in F} a_{ij}e_j - \sum_{k \in C} a_{ik}e_k$$

- Use assumptions about slow-to-converge error to collapse connections to $j \in F$ onto $k \in C \cap \{k : a_{ik} \neq 0\}$

A. Brandt, S. McCormick, J. Ruge, in *Sparsity and Its Applications*, 1984
J. Ruge and K. Stüben, in *Multigrid Methods*, 1987

Calibrating Interpolation

What if we don't know what to assume about slow-to-converge errors?

A. Brandt and D. Ron, in *Multilevel Optimization in VLSICAD*, 2003
M. Brezina et al., SISC 2004, **25**:1896-1920

Calibrating Interpolation

What if we don't know what to assume about
slow-to-converge errors?
Run relaxation to find out!

- Run relaxation on $Ax = 0$ with a random initial guess
- This exposes the local character of slow-to-converge errors
- Use resulting vector as a prototype of errors to be corrected by interpolation within algebraic multigrid

---

A. Brandt and D. Ron, in *Multilevel Optimization in VLSICAD*, 2003
M. Brezina et al., SISC 2004, **25**:1896-1920

Adaptive Multigrid

Automatic probing of relaxation and algebraic coarsening

- Given matrix A, Relaxation operation $B^{-1}r$
- Iterate on homogeneous problem, $Ax = 0$, with a random initial guess
- Create AMG-style interpolation such that prototype of slow-to-converge error is in its range
- Create coarse-grid problem and recurse

Adaptive Multigrid

Automatic probing of relaxation and algebraic coarsening

- Given matrix A, Relaxation operation $B^{-1}r$
- Iterate on homogeneous problem, $Ax = 0$, with a random initial guess
- Create AMG-style interpolation such that prototype of slow-to-converge error is in its range
- Create coarse-grid problem and recurse

Relaxation can be anything

Adaptive Multigrid

Automatic probing of relaxation and algebraic coarsening

- Given matrix A, Relaxation operation $B^{-1}r$
- Iterate on homogeneous problem, $Ax = 0$, with a random initial guess
- Create AMG-style interpolation such that prototype of slow-to-converge error is in its range
- Create coarse-grid problem and recurse

Relaxation can be anything,
even the multigrid method itself!

- Allows for iterative improvement of a poorly performing multigrid cycle

Controlling Adaptation

- Two possible sources of slow adaptive MG convergence
  - Prototype is a bad representative error
  - Prototype is good, but there is distinct slow-to-converge error
- Want a measure to distinguish cause of bad performance

Use estimates of $\|I - B^{-1}A\|$ to measure both performance and quality of prototype sets

- Estimate $\lambda_{\min}(B^{-1}A)$ using Rayleigh Quotients

Algorithm Overview

- while $\|I - B_{\mathrm{MG}}^{-1}A\|_{\mathrm{est}}$ is large
  - if $\|I - B_{\mathrm{rel}}^{-1}A\|_{\mathrm{est}}$ is increasing
    - iterate on $Ax = 0$ with "relaxation", $x \leftarrow (I - B_{\mathrm{rel}}^{-1}A)x$
    - recalibrate interpolation based on new x
    - recompute coarse-grid operator
    - restrict x to coarse grid and cycle there
    - interpolate further improved x after coarse-grid cycle
  - else
    - Replace "relaxation" with multigrid cycle: $B_{\mathrm{rel}} \leftarrow B_{\mathrm{MG}}$

Testing Adaptation

- 2-D Finite Element Shifted Laplacian, Dirichlet BCs, 512×512 grid

$$-\Delta u - 2\pi^2(1 - 2^{-15})u = 0$$

- $\lambda_{\min} = 6.64 \times 10^{-4}$, random $x^{(0)}$

Iteration	$\|I - B_{\text{rel}}^{-1}A\|_{\text{est}}$	$\|I - B_{\text{MG}}^{-1}A\|_{\text{est}}$
1	0.87	0.9999998
2	0.996	0.999985
3	0.99988	0.9996
4	0.999997	0.986
5	0.99999993	0.622
6	0.999999997	0.078
7	0.999999998	0.071

Flow in Porous Media

- Model pressure, p, of single-phase steady-state saturated flow in media with conductivity, K,

$$-\nabla \cdot K \nabla p = f$$

- Problem 1:

$$K(x,y) = \begin{cases} 10^{-8} & \text{if}(x,y) \in [\frac{1}{3}, \frac{2}{3}]^2 \\ 1 & \text{otherwise} \end{cases}$$

- Problem 2:

$$K(x,y) = \begin{cases} 10^{-8} & \text{on 20\% of elements, chosen randomly} \\ 1 & \text{otherwise} \end{cases}$$

Numerical Results: Porous Media

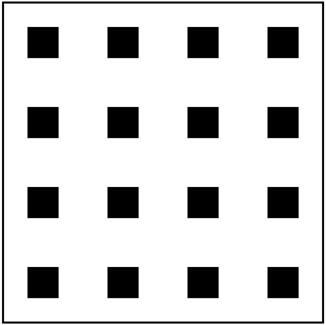
2D square, fixed coarsening, 10^{10} residual reduction

		Classical AMG			Adaptive AMG		
	h	ρ_{MG}	Itns	CPU (s)	ρ_{MG}	Itns	CPU (s)
1	$\frac{1}{256}$	0.130	9	0.9	0.081	8	0.9
	$\frac{1}{512}$	0.136	9	3.4	0.110	8	3.6
	$\frac{1}{1024}$	0.141	9	13.2	0.103	8	14.6
2	$\frac{1}{256}$	0.233	11	1.0	0.243	11	1.1
	$\frac{1}{512}$	0.290	13	4.4	0.288	13	4.8
	$\frac{1}{1024}$	0.375	14	17.6	0.376	16	22.1

M. Brezina et al., SISC 2006, **27**:1261-1286

Relationship to Modelling

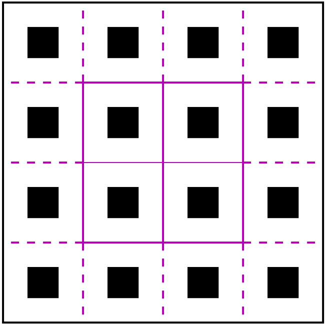
As interpolation is adapted, better resolution of physical problem appears on the coarse scales



Tiling of periodic inclusion of $K = 10^3$ (black), $K = 1$ in background

Relationship to Modelling

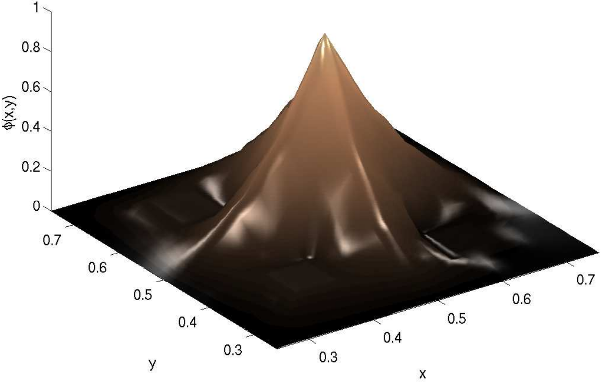
As interpolation is adapted, better resolution of physical problem appears on the coarse scales



Tiling of periodic inclusion of $K = 10^3$ (black), $K = 1$ in background

Relationship to Modelling

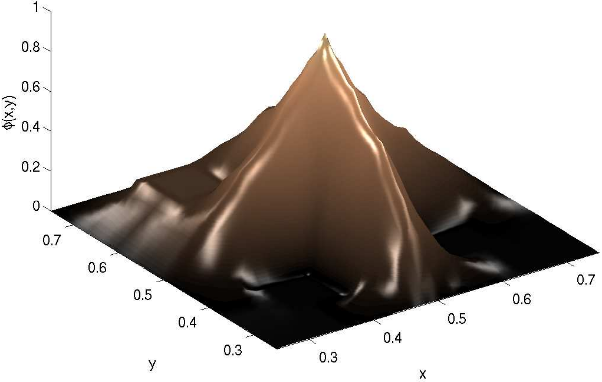
As interpolation is adapted, better resolution of physical problem appears on the coarse scales



Interpolant of $\delta_{(\frac{1}{2},\frac{1}{2})}$ after 1 cycle, $\rho_{\mathsf{MG}} = 0.973$

Relationship to Modelling

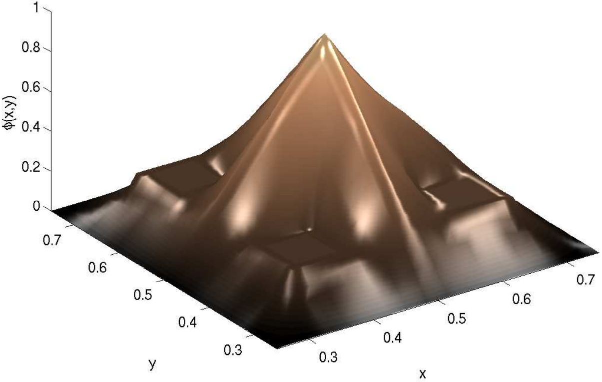
As interpolation is adapted, better resolution of physical problem appears on the coarse scales



Interpolant of $\delta_{(\frac{1}{2},\frac{1}{2})}$ after 2 cycles, $\rho_{\mathsf{MG}} = 0.851$

Relationship to Modelling

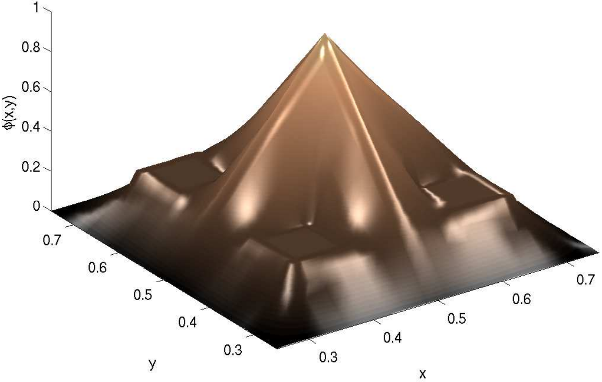
As interpolation is adapted, better resolution of physical problem appears on the coarse scales



Interpolant of $\delta_{(\frac{1}{2},\frac{1}{2})}$ after 3 cycles, $\rho_{\mathrm{MG}} = 0.375$

Relationship to Modelling

As interpolation is adapted, better resolution of physical problem appears on the coarse scales



Interpolant of $\delta_{(\frac{1}{2},\frac{1}{2})}$ after 4 cycles, $\rho_{\text{MG}} = 0.100$

Linear Elasticity

- Model displacement, u, of an elastic body under external forces

$$-\mu\Delta u - (\lambda + \mu)\nabla\nabla \cdot u = f$$

- μ, λ are Lamé coefficients, defined as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{and} \quad \mu = \frac{E}{2(1+\nu)}$$

- Fix Poisson ratio, $\nu = 0.32$ (steel)
- Let Young modulus, E, vary between 1 (nylon/polypro) and 10^σ ($100 =$ titanium, $1000 =$ diamond)
- Know properties of slow-to-converge errors for small σ

Numerical Results: Linear Elasticity

3D cube, 201,720 DOFs, exponential distribution of E

	Standard SA			Adaptive SA		
σ	ρ_{MG}	Itns	CPU (s)	ρ_{MG}	Itns	CPU (s)
2	0.115	9	26.0	0.214	12	267.7
3	0.247	14	35.7	0.310	16	275.6
4	0.395	20	50.0	0.404	21	289.4
5	0.556	32	73.6	0.497	27	381.2

M. Brezina et al., SISC 2004, **25**:1896-1920

Lattice Quantum Chromodynamics

- Modelling interactions between fermions on a lattice
- Goal: Solve $H(u, \rho)f = b$, for multiple source vectors, b, at each step of a Monte Carlo simulation
- Difficulty: u is a complex unitary field defined on lattice edges, phases chosen randomly based on parameter, β
- H is Hermitian, but indefinite, so solve normal equations
- As ρ approaches a critical value, H^*H becomes singular (for any β)
- Structure of low-energy modes strongly depends on u
  - When $\beta \to \infty$, $u \to 1$, H^*H looks like a second-order discrete differential operator
  - For each state, new characterization of low-energy modes

Numerical Results: Lattice QCD

128×128 periodic lattice
average residual reduction per iteration

	Diagonal-PCG				AdaptiveMG-PCG			
$\rho - \rho_{cr}$	0.3	0.1	0.05	0.01	0.3	0.1	0.05	0.01
$\beta = 2$	0.85	0.94	0.96	0.99	0.31	0.31	0.31	0.33
$\beta = 3$	0.86	0.93	0.97	0.98	0.31	0.40	0.42	0.42
$\beta = 5$	0.83	0.92	0.96	0.99	0.28	0.29	0.31	0.31

Adaptive MG setup time:	13.7 seconds
Adaptive MG-PCG solve time:	0.8 seconds
Diagonal-PCG solve time:	4.7 seconds

J. Brannick et al., to appear in *Proc. DD16*, 2006

Summary

- Heterogeneity adds new complication to linear solvers
- Algebraic picture of multigrid gives insight
- Adaptive framework replaces assumptions on relaxation
- Adaptive cycling allows iterative improvement of solver
- Added expense can be recovered for some applications

Future Directions

- Coupled systems (e.g., fluid-elastic)
- New application areas
- Hybrid smoothers

Support and Collaboration

- Research supported by the DOE SciDAC TOPS program, the Center for Applied Scientific Computing at Lawrence Livermore National Lab, and Los Alamos National Laboratory.
    - Adaptive AMG/SA development in collaboration with Steve McCormick, Tom Manteuffel, John Ruge, Marian Brezina at CU-Boulder, and Rob Falgout from CASC-LLNL.
    - Basis functions for porous media in collaboration with David Mounton from LANL
    - QCD problem in collaboration with James Brannick, Marian Brezina, Tom Manteuffel, Steve McCormick, John Ruge at CU-Boulder, David Keyes from Columbia, Oren Livne from Univ. Utah, Irene Livshits from Ball State U, and L. Zikatanov from Penn. State U