Improving Robustness in Multigrid Methods

Scott MacLachlan

maclachl@colorado.edu

Department of Applied Mathematics, University of Colorado at Boulder

Outline

- Modern Scientific Computing
- Multigrid Methods
- Self-Correcting Multigrid Methods
- Upscaling and Homogenization
- Future Work

Collaborators

- Steve McCormick
- Tom Manteuffel
- John Ruge
- Marian Brezina
- Rob Falgout
- David Moulton

Why Compute?

- Interested in modeling physical processes

Why Compute?

- Interested in modeling physical processes
  - Diffusion (Heat, Energy, Chemical)
  - Fluid Flow
  - Particle Transport
  - Elastic Materials

Why Compute?

- Interested in modeling physical processes
  - Diffusion (Heat, Energy, Chemical)
  - Fluid Flow
  - Particle Transport
  - Elastic Materials
- Can describe these processes through differential equations (both ODEs and PDEs)

Why Compute?

- Interested in modeling physical processes
  - Diffusion (Heat, Energy, Chemical)
  - Fluid Flow
  - Particle Transport
  - Elastic Materials
- Can describe these processes through differential equations (both ODEs and PDEs)
- Cannot write down closed form solutions
- Need to find (approximate) solutions in other ways

Scientific Computation

- Interested in simulating complex physical systems with parameters, and hence solutions, which vary on multiple scales
- Accuracy constraints lead to discretizations with tens of millions, or even billions, of degrees of freedom (DOFs)
- Need scalability, both algorithmic and parallel

Properties of Discretizations

- We consider (primarily) discretizations of the underlying differential equations via finite elements or finite differences
- The matrices from these discretizations tend to be
  - Sparse (number of nonzeros per row doesn't change with n)
  - Ill-conditioned
  - Symmetric (if DE is)
  - Positive-Definite (if DE is)

Direct Methods

- Interested in solving $Ax = b$
- Gauss Elimination involves factoring linear system into an upper- and a lower-triangular part
- Naive cost is $O(N^3) = O(n^9)$ for a 3-dimensional problem
- Utilizing bandedness of our discretization matrix can reduce cost to $O(n^7)$

Stationary Iterative Methods

- Stationary iterative methods choose approximations $B = A^{-1}$ and iterate using the error equation
- If $Ae = A(x - x^k) = b - Ax^k$, then $x^{k+1} = x^k + B(b - Ax^k)$
- The Jacobi iteration chooses B to be the diagonal of A
- The Gauss-Seidel iteration chooses B to be the lower-triangular part of A
- SOR chooses B to try and minimize $\rho(I - BA)$

Stationary Iterative Methods . . .

- Jacobi and Gauss-Seidel converge to the level of discretization error in $O(n^5)$ operations for the 3-dimensional Poisson problem
- SOR with an optimal parameter choice converges in $O(n^4)$ operations

Stationary Iterative Methods . . .

- Jacobi and Gauss-Seidel converge to the level of discretization error in $O(n^5)$ operations for the 3-dimensional Poisson problem
- SOR with an optimal parameter choice converges in $O(n^4)$ operations
- But, Jacobi and Gauss-Seidel resolve some components much faster than others
- In particular, for Poisson the geometrically smoothest components of x are the slowest to be resolved
- For this reason, Jacobi and Gauss-Seidel are often called smoothers - they smooth the error in the approximation

Krylov Methods

- Krylov methods find the optimal approximation to the solution in a given subspace
- Iteratively increase the size of the subspace to improve accuracy
- For Poisson, the Conjugate Gradient algorithm converges in $O(n^4)$ operations (without any parameter choice)

Scalability

- Because the problems we look to solve are so large, even the cost of $O(N^{\frac{4}{3}})$ is too much
- If $n = 1000$, then $N^{\frac{4}{3}} = 10^{12}$
- An algorithm is said to be scalable (or fast) if it requires only $O(N)$ or $O(N \log N)$ operations

Scalability

- Because the problems we look to solve are so large, even the cost of $O(N^{\frac{4}{3}})$ is too much
- If $n = 1000$, then $N^{\frac{4}{3}} = 10^{12}$
- An algorithm is said to be scalable (or fast) if it requires only $O(N)$ or $O(N \log N)$ operations
- We must have scalable algorithms in order to solve problems of interest at resolutions of interest

Multigrid Basics

- Need a solver whose performance doesn't significantly degrade as our problem size increases

Multigrid Basics

- Need a solver whose performance doesn't significantly degrade as our problem size increases
- Multigrid methods obtain optimal efficiency through complementarity

Multigrid Basics

- Need a solver whose performance doesn't significantly degrade as our problem size increases
- Multigrid methods obtain optimal efficiency through complementarity
- Use a smoothing process (such as Gauss-Seidel) to eliminate oscillatory errors

Multigrid Basics

- Need a solver whose performance doesn't significantly degrade as our problem size increases
- Multigrid methods obtain optimal efficiency through complementarity
- Use a smoothing process (such as Gauss-Seidel) to eliminate oscillatory errors
- Use coarse grid correction to eliminate smooth errors

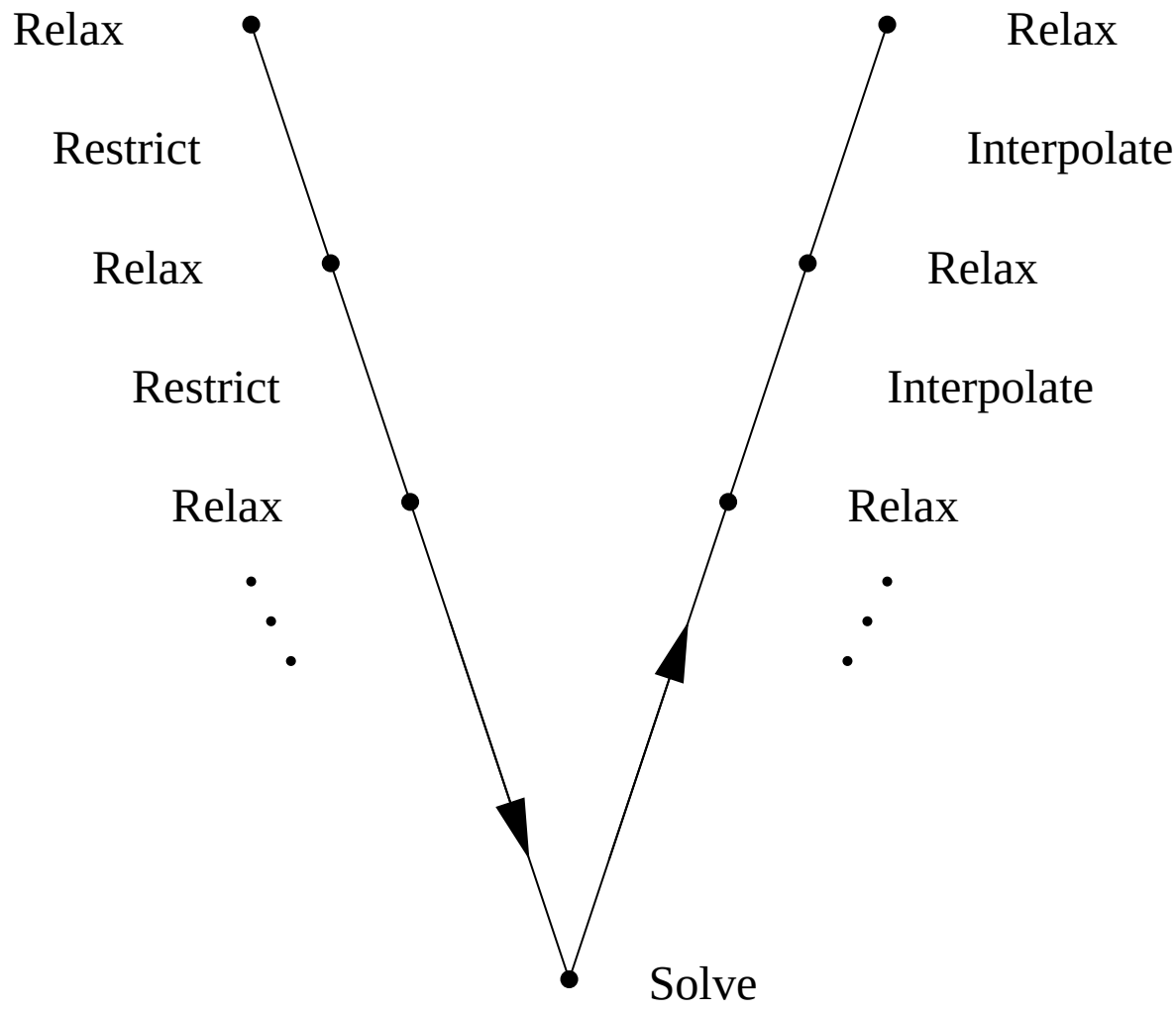
Multigrid Basics

- Need a solver whose performance doesn't significantly degrade as our problem size increases
- Multigrid methods obtain optimal efficiency through complementarity
- Use a smoothing process (such as Gauss-Seidel) to eliminate oscillatory errors
- Use coarse grid correction to eliminate smooth errors
- Obtain optimal efficiency through recursion

Coarse Grid Correction

- Smoothers, such as Jacobi or Gauss-Seidel, quickly reduce oscillatory error and leave smooth error
- Smooth error can be represented with fewer degrees of freedom
- Problems with fewer degrees of freedom can be solved with less effort
- Error which is smooth over many degrees of freedom appears oscillatory when represented on fewer DOFs

The V-Cycle


Relax
Restrict
Relax
Restrict
Relax
Solve
Relax
Interpolate
Relax
Interpolate
Relax

Intergrid Transfer Operators

- Multigrid V-Cycle requires transfers of residuals and corrections from one grid to the next

Intergrid Transfer Operators

- Multigrid V-Cycle requires transfers of residuals and corrections from one grid to the next
- Accomplished through Interpolation (Prolongation) and Restriction operators (matrices!)

Intergrid Transfer Operators

- Multigrid V-Cycle requires transfers of residuals and corrections from one grid to the next
- Accomplished through Interpolation (Prolongation) and Restriction operators (matrices!)
- Often pick a form of interpolation (P) and take restriction $R = P^T$ (theoretical benefits)

Intergrid Transfer Operators

- Multigrid V-Cycle requires transfers of residuals and corrections from one grid to the next
- Accomplished through Interpolation (Prolongation) and Restriction operators (matrices!)
- Often pick a form of interpolation (P) and take restriction $R = P^T$ (theoretical benefits)
- Many choices for interpolation
  - Piecewise constant
  - Linear, bilinear, trilinear
  - ...
  - Operator Induced

Coarse Grid Operators

- Smoothing on coarse grids requires operators on those grids

Coarse Grid Operators

- Smoothing on coarse grids requires operators on those grids
- These operators must well-approximate the fine grid operator

Coarse Grid Operators

- Smoothing on coarse grids requires operators on those grids
- These operators must well-approximate the fine grid operator
- Many ways to create coarse grid operators (CGOs)
  - Rediscretization
  - Averaging
  - Galerkin coarsening

Variational Multigrid

- Multigrid with $R = P^T$ and $A_c = RAP$ is called a *variational formulation*

Variational Multigrid

- Multigrid with $R = P^T$ and $A_c = RAP$ is called a *variational formulation*
- Terminology comes from minimization form of $Ax = b$:

$$F(v) = \frac{1}{2}\langle Av, v\rangle - \langle b, v\rangle$$

$$x = \arg\min_{v \in \mathcal{H}} F(v)$$

Variational Multigrid

- Multigrid with $R = P^T$ and $A_c = RAP$ is called a *variational formulation*
- Terminology comes from minimization form of $Ax = b$:

$$F(v) = \frac{1}{2}\langle Av, v\rangle - \langle b, v\rangle$$

$$x = \arg\min_{v \in \mathcal{H}} F(v)$$

- Given an approximation v to the solution on the fine level, it can be shown that the optimal coarse grid correction Pw solves

$$(P^T AP)w = P^T(b - Av)$$

Geometric Multigrid

- When the original problem $Ax = b$ comes from a geometrically regular discretization of a DE, we can use geometric information in the coarse-grid problems
- Coarse grids can be created by removing points from the fine-grid in a geometrically regular fashion
- Coarse grid operators can be determined by simple rediscretization on the reduced space
- Interpolation operators can be determined by geometric locations

Algebraic Multigrid

- In the absence of geometric information, choices must be made based on algebraic information
- Interpolation and coarse grids must be chosen based on the ability to interpolate a suitable correction
- Coarse grid operators must be chosen based on the fine-grid operator - Galerkin coarsening may be the most natural choice

Self-Correcting AMG

- Interested in applying algebraic multigrid methods to “more difficult” problems, such as systems (e.g. elasticity)

Self-Correcting AMG

- Interested in applying algebraic multigrid methods to “more difficult” problems, such as systems (e.g. elasticity)
- Develop an algebraic multigrid solver with increased robustness properties while not sacrificing optimality

Self-Correcting AMG

- Interested in applying algebraic multigrid methods to “more difficult” problems, such as systems (e.g. elasticity)
- Develop an algebraic multigrid solver with increased robustness properties while not sacrificing optimality
- Develop a solver which defaults to simplicity if given a simple problem

Basic Multigrid Properties

- Simple (Gauss-Seidel) Relaxation is inefficient for $Ax = b$ on error components e that give relatively small residuals: Ae is "small" relative to e (e is said to be algebraically smooth)

Basic Multigrid Properties

- Simple (Gauss-Seidel) Relaxation is inefficient for $Ax = b$ on error components e that give relatively small residuals: Ae is "small" relative to e (e is said to be algebraically smooth)
  - Example: For s.p.d. matrices, Richardson iteration stalls iff the error e satisfies

$$\langle Ae, e\rangle \ll \|A\|\langle e, e\rangle$$

Basic Multigrid Properties

- Simple (Gauss-Seidel) Relaxation is inefficient for $Ax = b$ on error components e that give relatively small residuals: Ae is "small" relative to e (e is said to be algebraically smooth)
  - Example: For s.p.d. matrices, Richardson iteration stalls iff the error e satisfies

$$\langle Ae, e\rangle \ll \|A\|\langle e, e\rangle$$

- For efficient multigrid performance, relaxation and coarse grid correction must be complementary

Main Ideas

- Build a multigrid hierarchy based on an approximation of the components that relaxation is slow to resolve

Main Ideas

- Build a multigrid hierarchy based on an approximation of the components that relaxation is slow to resolve
- Complementarity of relaxation and coarse grid correction means that if relaxation is inefficient on a component then that component must be treated by coarse grid correction

Main Ideas

- Build a multigrid hierarchy based on an approximation of the components that relaxation is slow to resolve
- Complementarity of relaxation and coarse grid correction means that if relaxation is inefficient on a component then that component must be treated by coarse grid correction
- Components that are slow to converge for $Ax = b$ will also be slow for $Ax = 0$

Details of the Method

- "Relax" on $Ax = 0$ with a random initial guess to quickly resolve a representative of the slow-to-converge components

Details of the Method

- "Relax" on $Ax = 0$ with a random initial guess to quickly resolve a representative of the slow-to-converge components or show that relaxation is sufficient for the problem

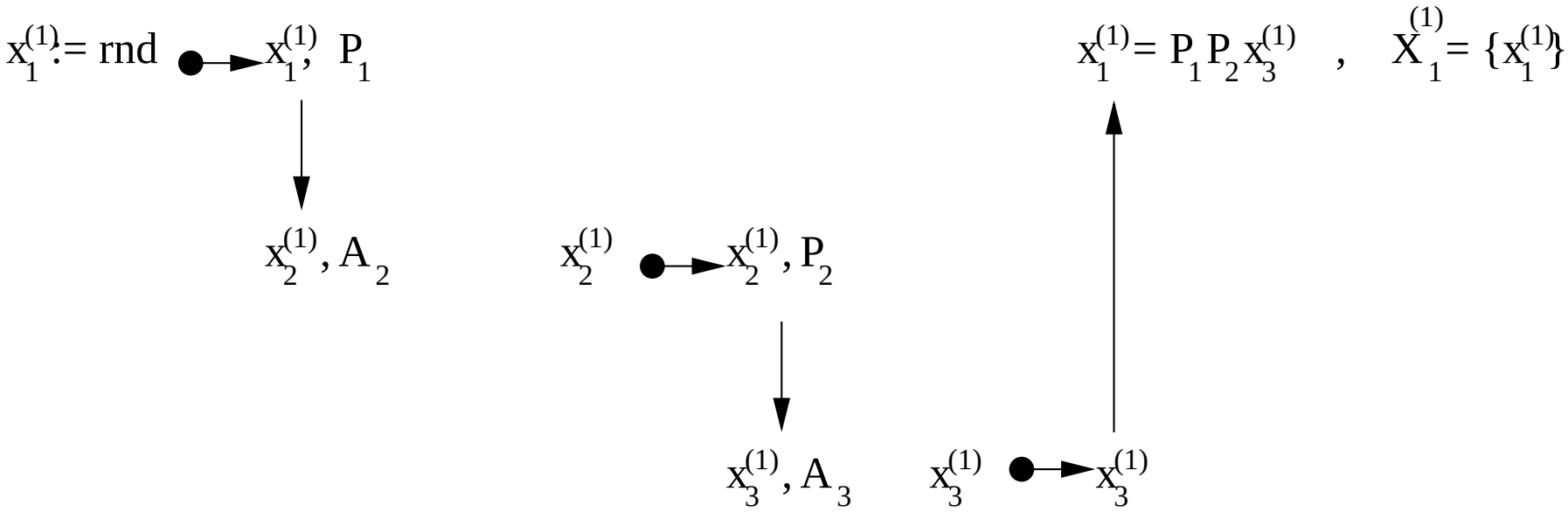
Details of the Method

- "Relax" on $Ax = 0$ with a random initial guess to quickly resolve a representative of the slow-to-converge components or show that relaxation is sufficient for the problem
- Define a 2-grid method by choosing a coarse grid and interpolation so that this component is in the range of interpolation (and using variational properties)

Details of the Method

- "Relax" on $Ax = 0$ with a random initial guess to quickly resolve a representative of the slow-to-converge components or show that relaxation is sufficient for the problem
- Define a 2-grid method by choosing a coarse grid and interpolation so that this component is in the range of interpolation (and using variational properties)
- Go to a multigrid method by recursion

Details . . .

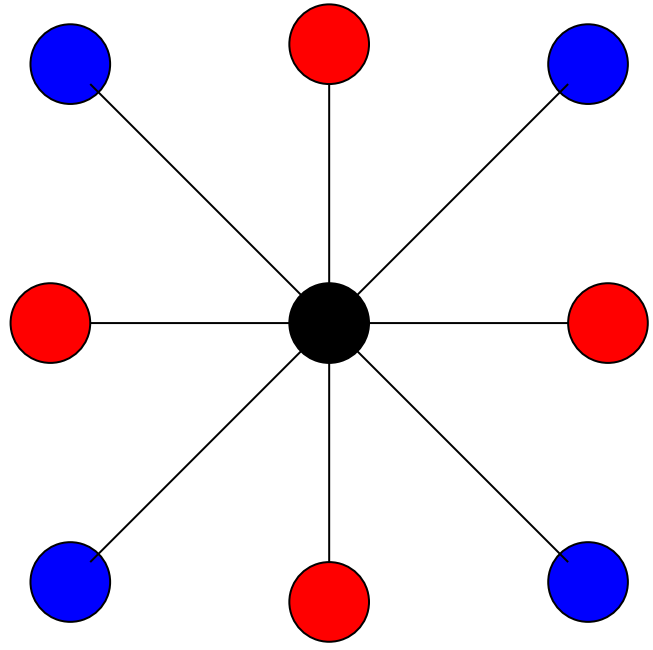

$x_1^{(1)} := \mathrm{rnd}$
$x_1^{(1)},\ P_1$
$x_2^{(1)}, A_2$
$x_2^{(1)}$
$x_2^{(1)}, P_2$
$x_3^{(1)}, A_3$
$x_3^{(1)}$
$x_3^{(1)}$
$x_1^{(1)} = P_1 P_2 x_3^{(1)}$
,
$X_1^{(1)} = \{x_1^{(1)}\}$

Choosing Interpolation

- Seek to define interpolation to fit an algebraically smooth vector
- Algebraic smoothness means

$$(Ae)_i \approx 0$$

N_i, the neighborhood of i



Fine Grid Points

Coarse Grid Points

Choosing Interpolation

- Seek to define interpolation to fit an algebraically smooth vector
- Algebraic smoothness means

$$(Ae)_i \approx 0$$

$$\text{or } a_{ii}e_i \approx -\sum_{j \in N_i} a_{ij}e_j$$

$$= -\sum_{j \in C_i} a_{ij}e_j - \sum_{k \in F_i} a_{ik}e_k$$

- To define interpolation, need to collapse connections from F_i to C_i

Choosing Interpolation . . .

- Seek to define interpolation to fit an algebraically smooth vector

Choosing Interpolation . . .

- Seek to define interpolation to fit an algebraically smooth vector
- If $k \in F_i$ is connected to a set of $j \in C_i$, we want to write

$$e_k = \sum_{j \in C_i} w_{kj} e_j$$

Choosing Interpolation . . .

- Seek to define interpolation to fit an algebraically smooth vector
- If $k \in F_i$ is connected to a set of $j \in C_i$, we want to write

$$e_k = \sum_{j \in C_i} w_{kj} e_j$$

- Have a vector, $x^{(1)}$, such that $(Ax^{(1)})_k \approx 0$ and so

$$a_{kk} x_k^{(1)} = - \sum_{j \in C_i} a_{kj} x_j^{(1)} - \sum_{j \notin C_i} a_{kj} x_j^{(1)}$$

Choosing Interpolation . . .

- Seek to define interpolation to fit an algebraically smooth vector
- If $k \in F_i$ is connected to a set of $j \in C_i$, we want to write

$$e_k = \sum_{j \in C_i} w_{kj} e_j$$

- Have a vector, $x^{(1)}$, such that $(Ax^{(1)})_k \approx 0$ and so

$$a_{kk} x_k^{(1)} = - \sum_{j \in C_i} a_{kj} x_j^{(1)} - \sum_{j \notin C_i} a_{kj} x_j^{(1)}$$

- Eliminate extra terms by replacing matrix entry a_{kk} with arbitrary d_{kk}

Choosing Interpolation . . .

- Eliminate extra terms by replacing matrix entry a_{kk} with arbitrary d_{kk}

$$d_{kk}x_k^{(1)} = -\sum_{j \in C_i} a_{kj}x_j^{(1)}$$

Choosing Interpolation . . .

- Eliminate extra terms by replacing matrix entry a_{kk} with arbitrary d_{kk}

$$d_{kk} x_k^{(1)} = - \sum_{j \in C_i} a_{kj} x_j^{(1)}$$

- Taking the value of d_{kk} given here, we can write

$$x_k^{(1)} = - \sum_{j \in C_i} \frac{a_{kj}}{d_{kk}} x_j^{(1)} = \sum_{j \in C_i} \frac{a_{kj} x_k^{(1)}}{\sum_{j' \in C_i} a_{kj'} x_{j'}^{(1)}} x_j^{(1)}$$

Choosing Interpolation . . .

- Eliminate extra terms by replacing matrix entry a_{kk} with arbitrary d_{kk}

$$d_{kk} x_k^{(1)} = - \sum_{j \in C_i} a_{kj} x_j^{(1)}$$

- Taking the value of d_{kk} given here, we can write

$$x_k^{(1)} = - \sum_{j \in C_i} \frac{a_{kj}}{d_{kk}} x_j^{(1)} = \sum_{j \in C_i} \frac{a_{kj} x_k^{(1)}}{\sum_{j' \in C_i} a_{kj'} x_{j'}^{(1)}} x_j^{(1)}$$

- Use this formula to collapse all algebraically smooth error

Choosing Interpolation . . .

$$e_k = \sum_{j \in C_i} \left(\frac{a_{kj} x_k^{(1)}}{\sum_{j' \in C_i} a_{kj'} x_{j'}^{(1)}} \right) e_j = \sum_{j \in C_i} w_{kj} e_j$$

- Then, using the definition of algebraic smoothness, we have

$$a_{ii} e_i \approx - \sum_{j \in C_i} a_{ij} e_j - \sum_{k \in F_i} a_{ik} e_k$$

$$a_{ii} e_i \approx - \sum_{j \in C_i} a_{ij} e_j - \sum_{k \in F_i} \sum_{j \in C_i} a_{ik} w_{kj} e_j$$

Choosing Interpolation . . .

So, we define interpolation to a fine grid point i as

$$e_i = -\sum_{j \in C_i} \frac{a_{ij} + \sum_{k \in F_i} a_{ik} w_{kj}}{a_{ii}} e_j$$

$$= -\sum_{j \in C_i} \frac{a_{ij} + \sum_{k \in F_i} a_{ik} \left(\frac{a_{kj} x_k^{(1)}}{\sum_{j' \in C_i} a_{kj'} x_{j'}^{(1)}} \right)}{a_{ii}} e_j$$

Scaling Property

- If we scale A as DAD for some diagonal matrix D, scAMG performance need not suffer
- If $Ax^{(1)} = 0$, then $DAD(D^{-1}x^{(1)}) = 0$
- Replacing $x^{(1)}$ with $D^{-1}x^{(1)}$, can show that coarse grid matrices for DAD are simply diagonally-rescaled versions of the coarse grid matrices for A
- Pointwise relaxation is also invariant to diagonal scaling
- If we can generate $D^{-1}x^{(1)}$ as easily as we get $x^{(1)}$, overall performance won't degrade

Current Assumptions

- Coarse grids are predetermined and sufficient for full multigrid efficiency
  - Currently choosing coarse grids based on geometric criteria, could also use Ruge-Stueben algebraic coarsening
  - Eventually hope to determine coarse grids adaptively as well (Compatible Relaxation)

Numerical Results

- $-\nabla \cdot D(x,y)\nabla u(x,y) = 0$ on $[0,1]^2$
- Geometric choice of coarse grids
- Interpolation chosen as above given the vector $x^{(1)}$ given by two cycles of a multilevel relaxation procedure

$D(x,y) = 1$ **(Laplace)**

size	Dirichlet BCs	Neumann BCs
32×32	0.058	0.064
64×64	0.065	0.067
128×128	0.068	0.070
256×256	0.070	0.070
512×512	0.070	0.070
1024×1024	0.070	0.070

Piecewise Constant $D(x,y)$

- Dirichlet BCs on left and right, Neumann BCs on top and bottom

$$D_1(x,y) = \begin{cases} 10^2 & (x,y) \in [\frac{1}{3}, \frac{2}{3}]^2 \\ 1 & \text{otherwise} \end{cases} \qquad D_2(x,y) = \begin{cases} 10^5 & y \leq \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

size	$D_1(x,y)$	$D_2(x,y)$
32×32		
64×64		
128×128		
256×256		
512×512		
1024×1024		

Piecewise Constant $D(x,y)$

- Dirichlet BCs on left and right, Neumann BCs on top and bottom

$$D_1(x,y) = \begin{cases} 10^2 & (x,y) \in [\frac{1}{3}, \frac{2}{3}]^2 \\ 1 & \text{otherwise} \end{cases} \qquad D_2(x,y) = \begin{cases} 10^5 & y \leq \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

size	$D_1(x,y)$	$D_2(x,y)$
32×32		0.060
64×64		0.067
128×128		0.069
256×256		0.070
512×512		0.070
1024×1024		0.070

Piecewise Constant $D(x,y)$

- Dirichlet BCs on left and right, Neumann BCs on top and bottom

$$D_1(x,y) = \begin{cases} 10^2 & (x,y) \in [\frac{1}{3}, \frac{2}{3}]^2 \\ 1 & \text{otherwise} \end{cases} \qquad D_2(x,y) = \begin{cases} 10^5 & y \leq \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

size	$D_1(x,y)$	$D_2(x,y)$
32×32	0.078	0.060
64×64	0.094	0.067
128×128	0.081	0.069
256×256	0.106	0.070
512×512	0.169	0.070
1024×1024	0.384	0.070

Scaled Laplace

- Scaling is done based on (x, y)-coordinates of each node,

$$1 + \sin(547\pi x)\sin(496\pi y) + 10^{-7}$$

size	Dirichlet BCs	Neumann BCs
32×32	0.058	0.064
64×64	0.066	0.067
128×128	0.068	0.070
256×256	0.070	0.070
512×512	0.070	0.070
1024×1024	0.070	0.070

Scaled PW Constant $D(x, y)$

size	$D_1(x, y)$	$D_2(x, y)$
32×32	0.078	0.060
64×64	0.091	0.067
128×128	0.087	0.069
256×256	0.348	0.070
512×512	0.190	0.070
1024×1024	0.918	0.070

Porous Media Flow

- Interested in simulating, for example, flow in a reservoir
- Model saturated flow via Darcy's Law:

$$u(x,y) = -D(x,y)\nabla p(x,y)$$
$$\nabla \cdot u(x,y) = Q(x,y)$$

Porous Media Flow

- Interested in simulating, for example, flow in a reservoir
- Model saturated flow via Darcy's Law:

$$u(x,y) = -D(x,y)\nabla p(x,y)$$
$$\nabla \cdot u(x,y) = Q(x,y)$$

- Simulation domain is on the order of 10^3 meters in length in each dimension
- Fine scale changes in material properties on the order of 10^{-3} meters
- Range of scales is on the order of 10^6

The Curse of Dimensionality

- As we consider 2- and 3-dimensional simulations, cost of resolution increases exponentially
- For 1-D porous media flow, need $\backsim 10^6$ DOFs

The Curse of Dimensionality

- As we consider 2- and 3-dimensional simulations, cost of resolution increases exponentially
- For 1-D porous media flow, need $\backsim 10^6$ DOFs
- For 2-D porous media flow, need $\backsim 10^{12}$ DOFs

The Curse of Dimensionality

- As we consider 2- and 3-dimensional simulations, cost of resolution increases exponentially
- For 1-D porous media flow, need $\backsim 10^6$ DOFs
- For 2-D porous media flow, need $\backsim 10^{12}$ DOFs
- For 3-D porous media flow, need $\backsim 10^{18}$ DOFs
- Fully resolved 3-D simulation is still beyond the capability of modern supercomputers (the fastest of which performs 3.5×10^{13} floating point operations per second)

The Need for Upscaling

- Naive discretizations require too many DOFs to be computationally feasible
- We must accurately account for the influence of fine-scale variation in the material properties if we hope to obtain physically meaningful solutions

The Need for Upscaling

- Naive discretizations require too many DOFs to be computationally feasible
- We must accurately account for the influence of fine-scale variation in the material properties if we hope to obtain physically meaningful solutions
- In general, we cannot directly account for the influence of fine-scale variation in material properties in a coarse-scale discretization
- The goal of upscaling and homogenization techniques is to derive effective, coarse-scale material properties to use in coarse-scale models and discretizations

Durlofsky's Approach

- Based on a two-scale asymptotic analysis, and thus strictly valid only for two-scale periodic media
- Consider pressure which is locally of the form

$$p = p_0 + G \cdot (x - x_0),$$

then the average flow through a local cell can be shown to be

$$\langle u \rangle = -\hat{D} \cdot G.$$

- So, the local effective permeability can be recovered by choosing boundary conditions to induce particular G and then calculating the average flow for that G
- Overall upscaling technique requires solution of 2 fine-scale problems over each macro-element (in 2D)

Interpretation of Multigrid CGOs

- Consider a fine-scale discretization via finite elements

$$A_{ij} = e_j^T A e_i = \int_\Omega \langle D(x,y)\nabla\phi_i, \nabla\phi_j\rangle d\Omega$$

- Use of Galerkin coarsening means that the coarse grid operator is equivalent to a finite element discretization on that grid

$$\begin{aligned}(A_c)_{ij} &= (P^T A P)_{ij} = (P\hat{e}_j)^T A (P\hat{e}_i) \\ &= (\sum_k p_{kj} e_k^T) A (\sum_l p_{li} e_l) \\ &= \sum_{k,l} p_{kj} p_{li} (e_k^T A e_l)\end{aligned}$$

Interpretation . . .

- So,

$$
\begin{aligned}
(A_c)_{ij} &= \sum_{k,l} p_{kj} p_{li} \int_\Omega \langle D(x,y) \nabla \phi_l, \nabla \phi_k \rangle d\Omega \\
&= \int_\Omega \left\langle D(x,y) \nabla \left(\sum_l p_{li} \phi_l \right), \nabla \left(\sum_k p_{kj} \phi_k \right) \right\rangle d\Omega \\
&= \int_\Omega \langle D(x,y) \nabla \hat{\phi}_i, \nabla \hat{\phi}_j \rangle d\Omega
\end{aligned}
$$

- Basis functions on coarse grids come from summing the fine grid basis functions (weighted by the interpolation/restriction operators)

BoxMG

- The Black Box Multigrid Algorithm (BoxMG) was developed by Dendy for diffusion problems with discontinuous coefficients
- Coarsening is done in a geometrically regular fashion
- BoxMG chooses interpolation in a manner which preserves the continuity of normal flux
- BoxMG uses a variational formulation, and is thus quite robust
- In 2-D, if initial operator is 5-point or 9-point, then all coarse grid operators are 9-point operators

Reinterpretation of Multigrid CGOs

- Consider a bilinear discretization in 2-D
- Using a full-coarsening multigrid algorithm (such as BoxMG) results in 9-point operators on all coarse grids
- Any 9-point operator can be written as a linear combination of the bilinear FE operators for
  $I, \partial_x, \partial_y, \partial_{xx}, \partial_{yy}, \partial_{xy}, \partial_{xxy}, \partial_{xyy}, \partial_{xxyy}$
- If we start with a symmetric, zero row-sum operator, BoxMG coarsening guarantees that the coarse grid operator will also have these properties
- This forces the coarse grid operator to be a linear combination of $\partial_{xx}, \partial_{yy}, \partial_{xy}, \partial_{xxyy}$

Reinterpretation . . .

- The coarse grid operator can thus be interpreted as the coarse grid discretization of

$$-\nabla \cdot (\hat{D}\nabla u) + \partial_{xy}\hat{E}(x,y)\partial_{xy}u = \hat{f}$$

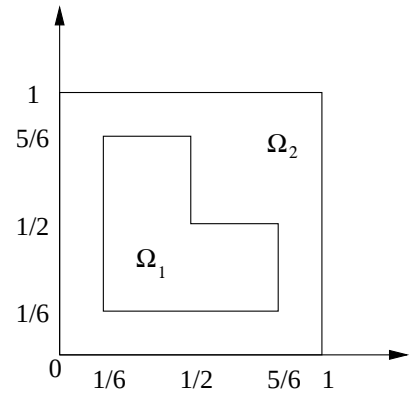
- It is possible to recover piecewise constant approximations of the effective $\hat{D}$ and $\hat{E}$ based on the stencil entries
- That is, we can recover the homogenized permeability tensor directly from the coarse grid operator

Numerical Homogenization

- Moulton et al and Knapek examine similar results for periodic BCs
- We have derived the needed relations to determine the effective material properties given a coarse-grid stencil in the case of Neumann BCs
- Consider Darcy Flow problem on $[0,1]^2$ with full Neumann BCs

Numerical Homogenization . . .

- Consider the domain



- With

$$D(x,y) = \begin{cases} 10 & (x,y) \in \Omega_1 \\ 1 & \text{otherwise} \end{cases}$$

Numerical Homogenization . . .

- The asymptotic computation of Bourgat gives

$$\begin{bmatrix} 1.915 & -0.101 \\ -0.101 & 1.915 \end{bmatrix} = Q \begin{bmatrix} 2.016 & 0 \\ 0 & 1.814 \end{bmatrix} Q^T,$$

where Q is the orthonormal matrix

$$\frac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$$

- On a grid of 768×768 gridpoints, we can recover

$$\hat{D} = \begin{bmatrix} 1.9339 & -0.1532 \\ -0.1532 & 1.9339 \end{bmatrix} = Q \begin{bmatrix} 2.0871 & 0 \\ 0 & 1.7807 \end{bmatrix} Q^T$$

Insight into Multigrid

- Accounting for the regularization term also explains the performance of multigrid on certain problems
- Consider, for example, the region $[0,1]^2$, with

$$D(x,y) = \begin{cases} 1 & \text{if } y > \frac{1}{2} \\ 0.01 & \text{if } y < \frac{1}{2} \end{cases}$$

- Can show that the homogenized permeability is anisotropic

$$\hat{D} = \begin{bmatrix} 0.505 & 0 \\ 0 & 0.0198 \end{bmatrix}$$

Insight . . .

- If we directly discretize the homogenized problem and use pointwise relaxation (such as Gauss-Seidel), we expect an inefficient algorithm
- Pointwise relaxation is inherently inefficient in anisotropic problems

Insight . . .

- If we directly discretize the homogenized problem and use pointwise relaxation (such as Gauss-Seidel), we expect an inefficient algorithm
- Pointwise relaxation is inherently inefficient in anisotropic problems
- But . . . we get good results!

Insight . . .

- If we directly discretize the homogenized problem and use pointwise relaxation (such as Gauss-Seidel), we expect an inefficient algorithm
- Pointwise relaxation is inherently inefficient in anisotropic problems
- But . . . we get good results!
- The regularization term makes the coarse scale problem effectively isotropic, while maintaining the coarse-scale effective permeability

Summary - scAMG

- Relax on $Ax = 0$ to find a representative vector of those that are slow to converge
- Determine multigrid interpolation operator based on matrix entries and this vector
- Multigrid algorithm obtained is invariant to diagonal scaling of A

Summary - Homogenization

- Can recover effective material properties from multigrid coarse-grid operators
- Actual coarsening introduces a regularization term into the CGO
- Understanding of regularization term provides insight into multigrid performance

Future Work - scAMG

- Want a more efficient scheme to determine a suitable representative vector
  - Choose a better starting guess than a random vector and use current procedure
  - Iterate on the eigenproblem to find low eigenmodes of A
- Consider iterating on problem $Ax = b$ to develop solver, instead of working on $Ax = 0$

Future Work - scAMG

- Want to extend class of problems for which algorithm performs well - particularly to include systems of PDEs
- May need to allow for more than one vector to be considered in interpolation
  - If we know the problem is a system, we can extend the definition of interpolation to include more vectors and retain its properties
  - May also look at updating interpolation to fit new vectors as they are determined

Future Work - scAMG

- Remove need for geometric coarsening in algorithm
  - Choose coarse grid based on algebraic criteria
  - For example, Compatible Relaxation which uses the efficiency of relaxation on the resulting fine-grid to choose the coarse-grid

Future Work - Homogenization

- Complete analysis of regularization term
- Complete MGH library
- Investigate use of homogenized permeabilities in Finite Volume discretizations and multigrid
- Investigate effect of regularization term in other multilevel solvers (AMG, scAMG)

Conclusions

- Have framework for self correcting multigrid solvers
- Self correcting ideas increase range of applicability of existing multigrid methods
- Multigrid coarse grid operators can be used to solve homogenization problem
- Homogenization can also provide significant insight into multigrid behavior

Conclusions

- Have framework for self correcting multigrid solvers
- Self correcting ideas increase range of applicability of existing multigrid methods
- Multigrid coarse grid operators can be used to solve homogenization problem
- Homogenization can also provide significant insight into multigrid behavior
- Much work still to be done