# Coarsening in Adaptive Algebraic Multigrid

Scott MacLachlan

maclach@cs.umn.edu

Department of Computer Science and Engineering, University of Minnesota

# 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 very different treatment
- Lattice quantum chromodynamics
  - Highly heterogeneous operator
  - Randomized heterogeneity within Monte Carlo process

# Target Discrete Models

- Finite element models of elliptic formulations of PDEs
- Matrices are
  - sparse
  - symmetric
  - positive definite

Solving $Ax = b$

- Sparsity with large bandwidth means that direct methods are not effective for these problems

# Stationary Iterative Methods

- Given some approximation, $x^{(0)}$, want to improve it
- Introduce residual, $r^{(0)} = b - Ax^{(0)}$, as a measure of the error
- $r^{(0)} = b - Ax^{(0)} = Ax - Ax^{(0)} = A(x - x^{(0)})$
- Let $B^{-1}$ be an approximation to $A^{-1}$
- Take $x^{(1)} = x^{(0)} + B^{-1}r^{(0)}$

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

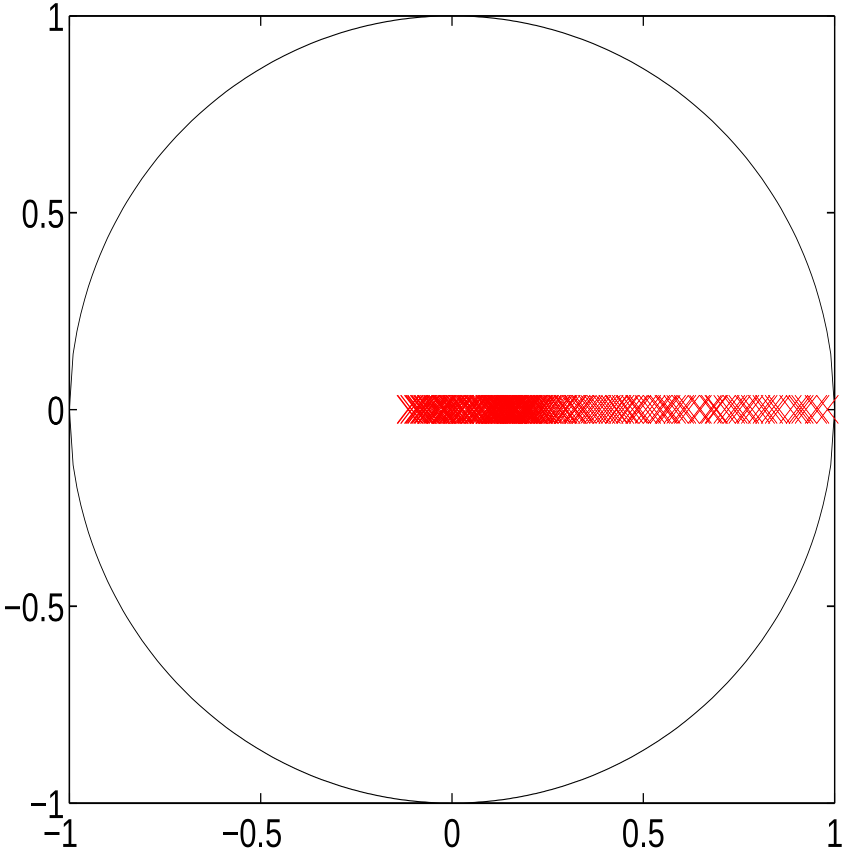
# Stationary Iterative Methods

- Given some approximation, $x^{(0)}$, want to improve it
- Introduce residual, $r^{(0)} = b - Ax^{(0)}$, as a measure of the error
- $r^{(0)} = b - Ax^{(0)} = Ax - Ax^{(0)} = A(x - x^{(0)})$
- Let $B^{-1}$ be an approximation to $A^{-1}$
- Take $x^{(1)} = x^{(0)} + B^{-1}r^{(0)}$

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

# Convergence of Stationary Iterative Methods

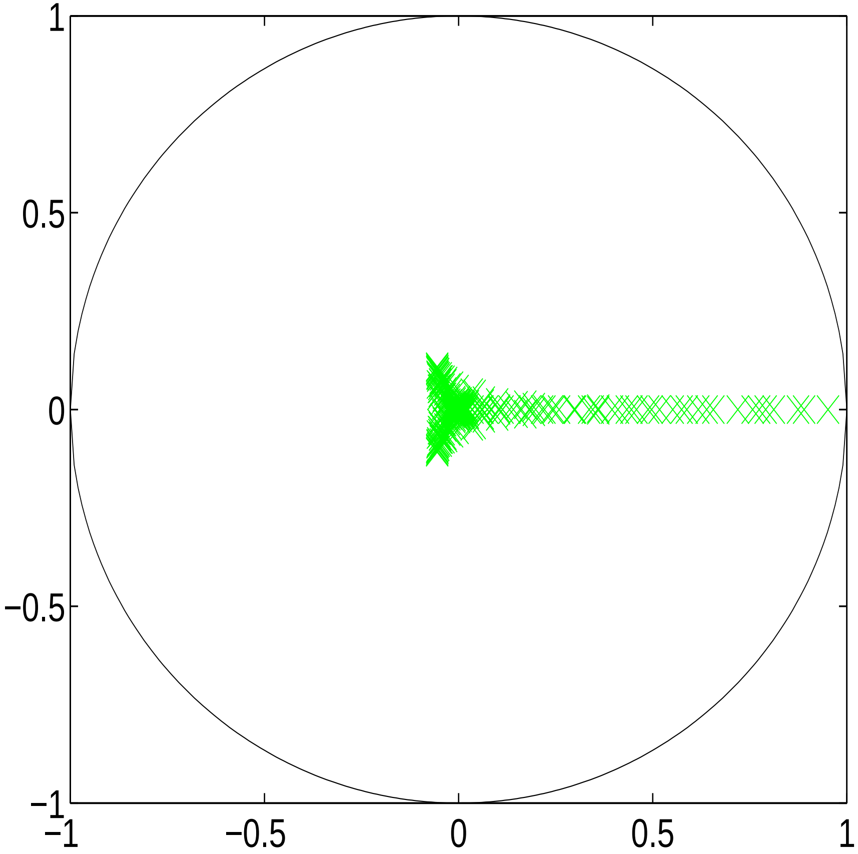
Convergence depends on spectrum of $I - B^{-1}A$



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

# Convergence of Stationary Iterative Methods

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

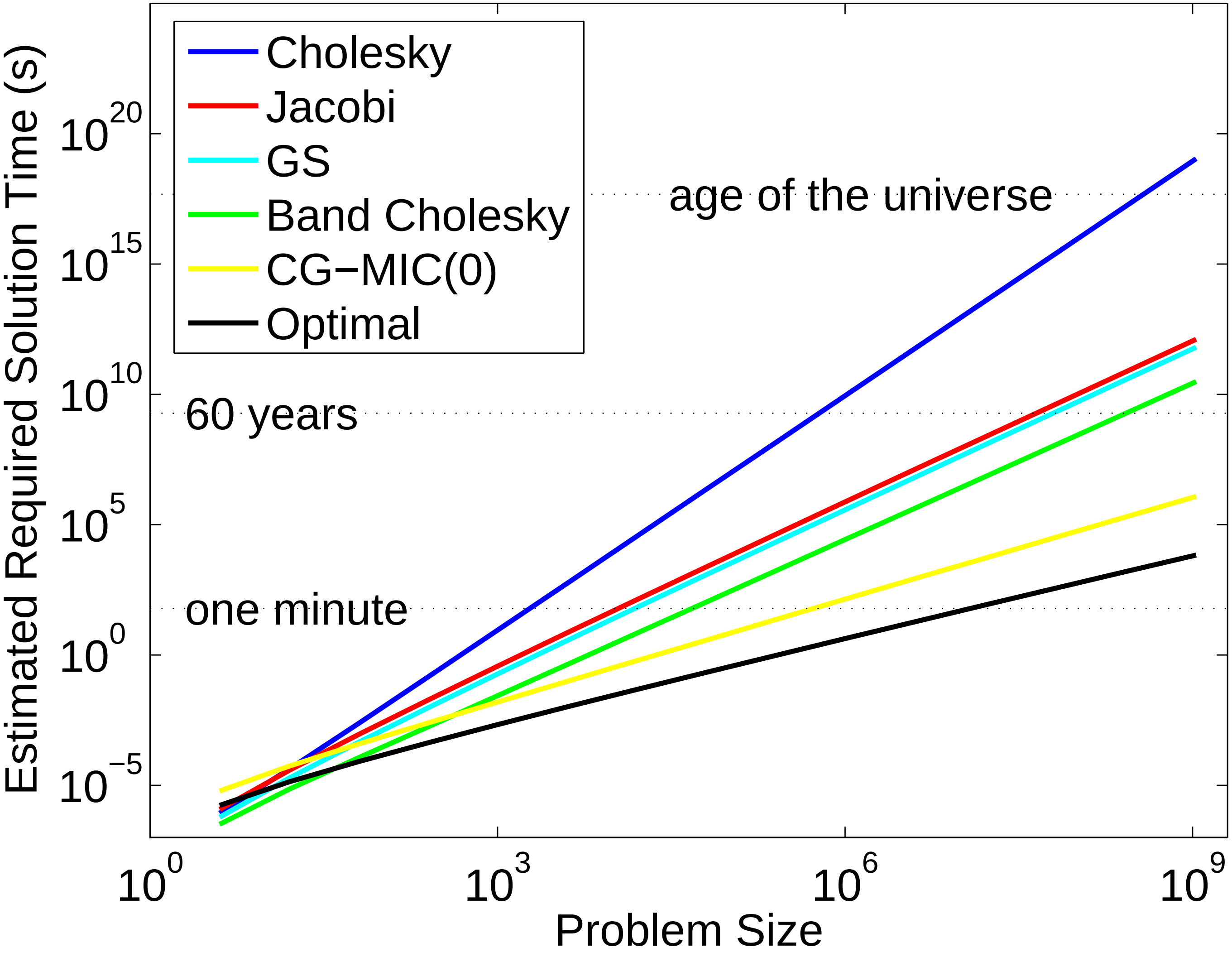


Gauss-Seidel Iteration: $e^{(n)} = (I - L^{-1}A)^n e^{(0)}$

Errors with small $B^{-1}A$-Rayleigh Quotients are slowest to converge

# Scalability

These methods fail when the problem size gets large enough!



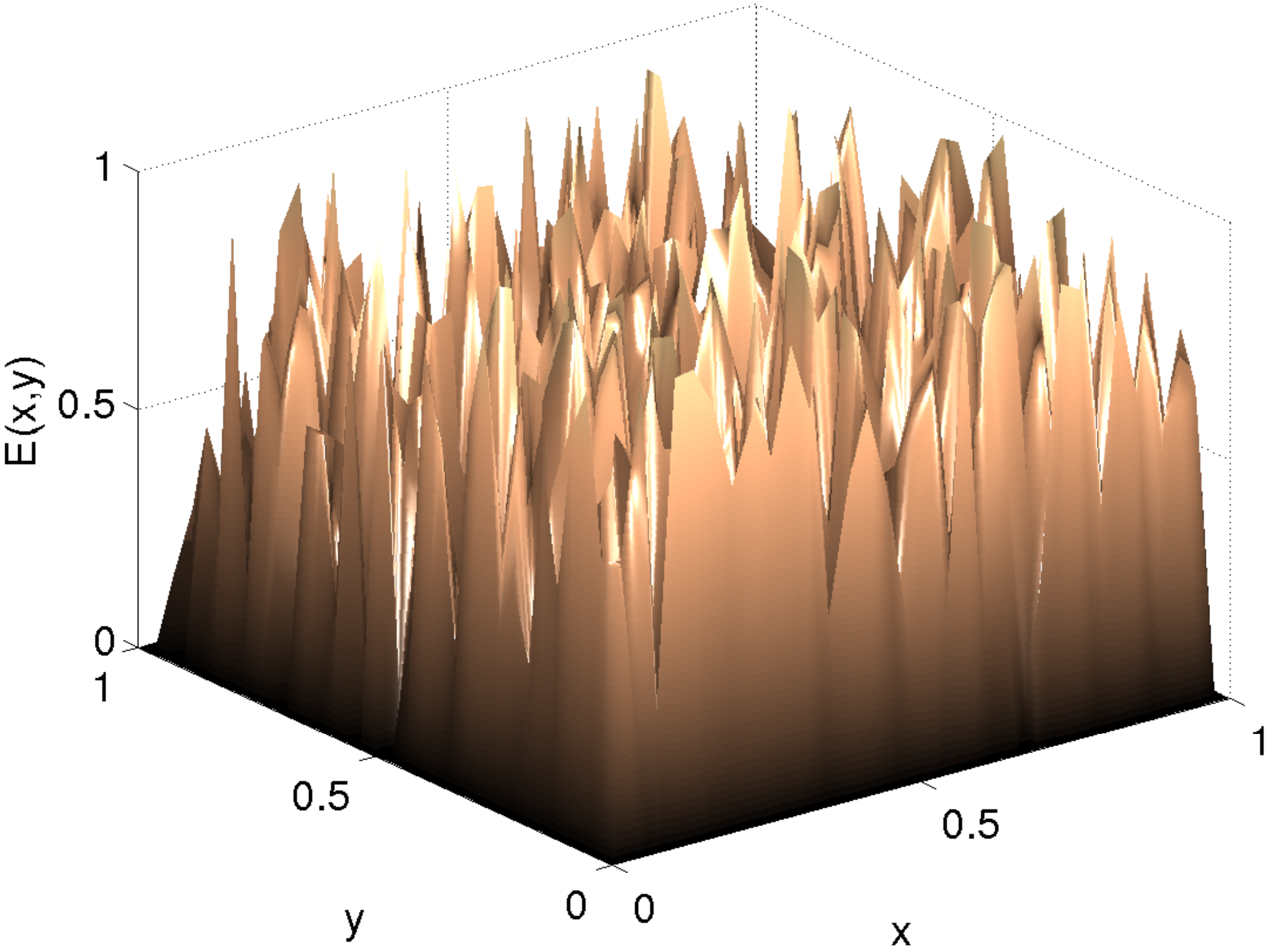
# Failing in a Structured Way

For all of these methods, low-energy modes of $B^{-1}A$ cause the most trouble

- For simple-enough $B$, these are the same as (or close to) the low-energy modes of $A$

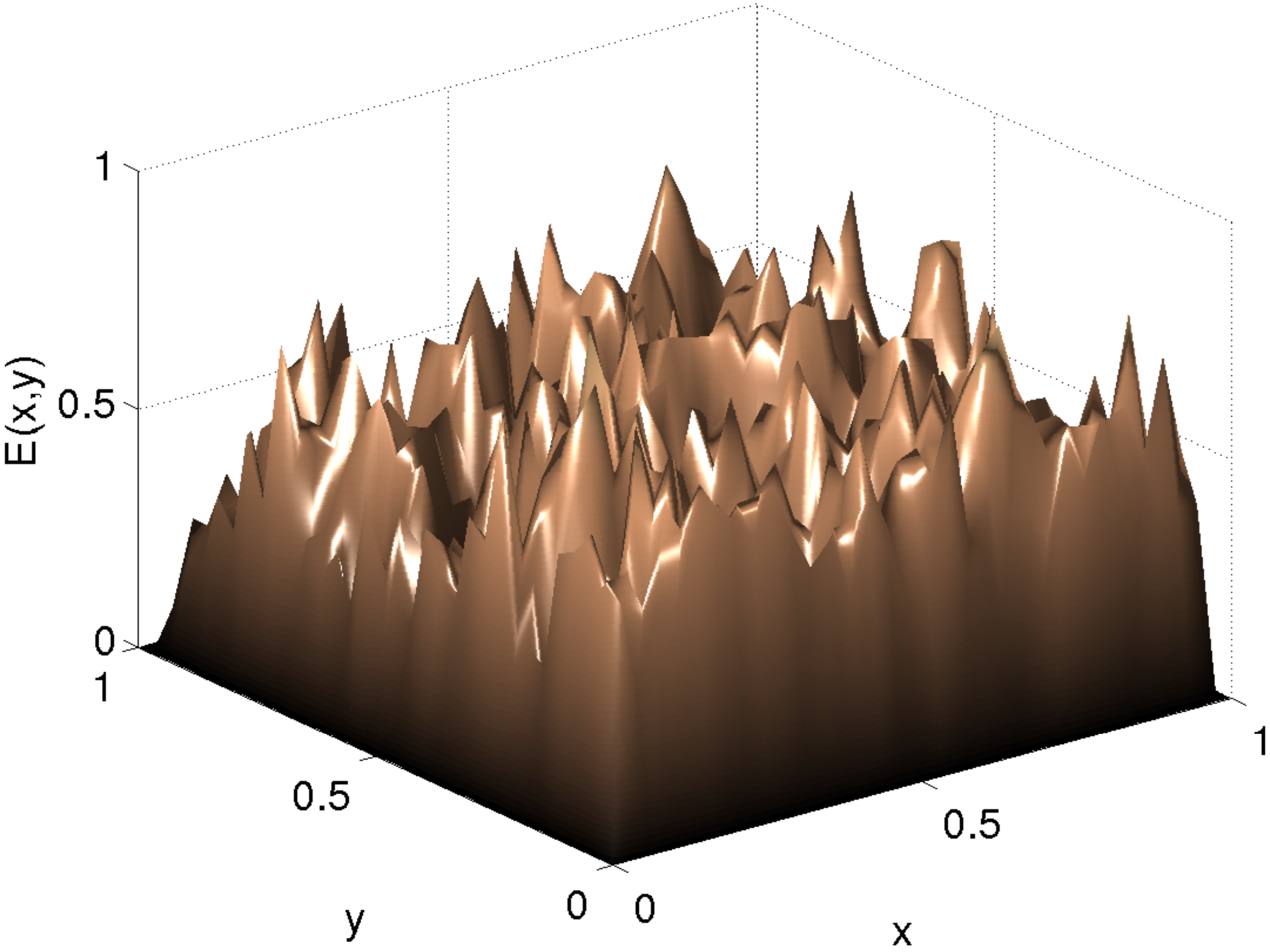
Can we use this to our advantage?

# Smoothing Property



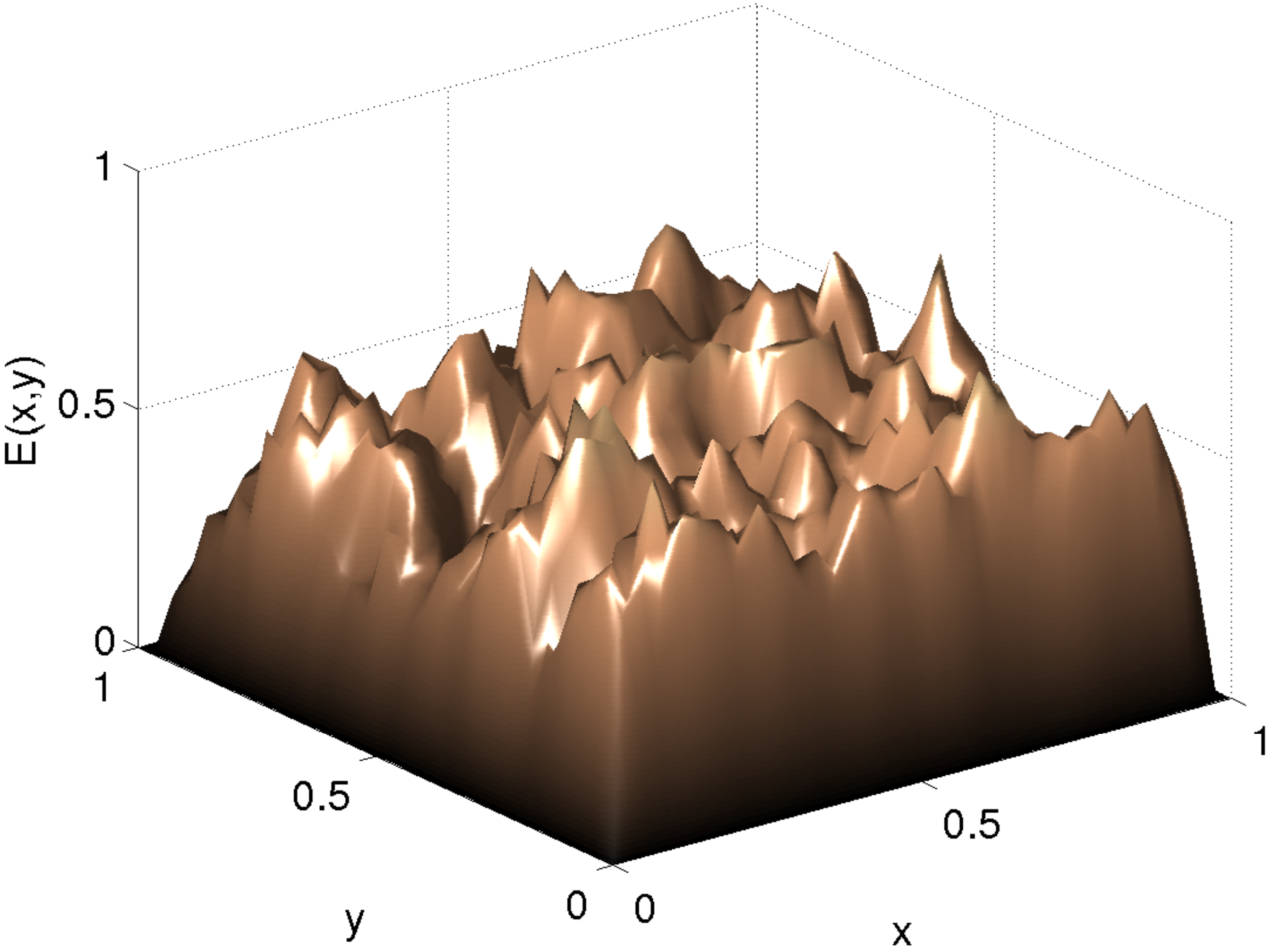
Random initial error

# Smoothing Property



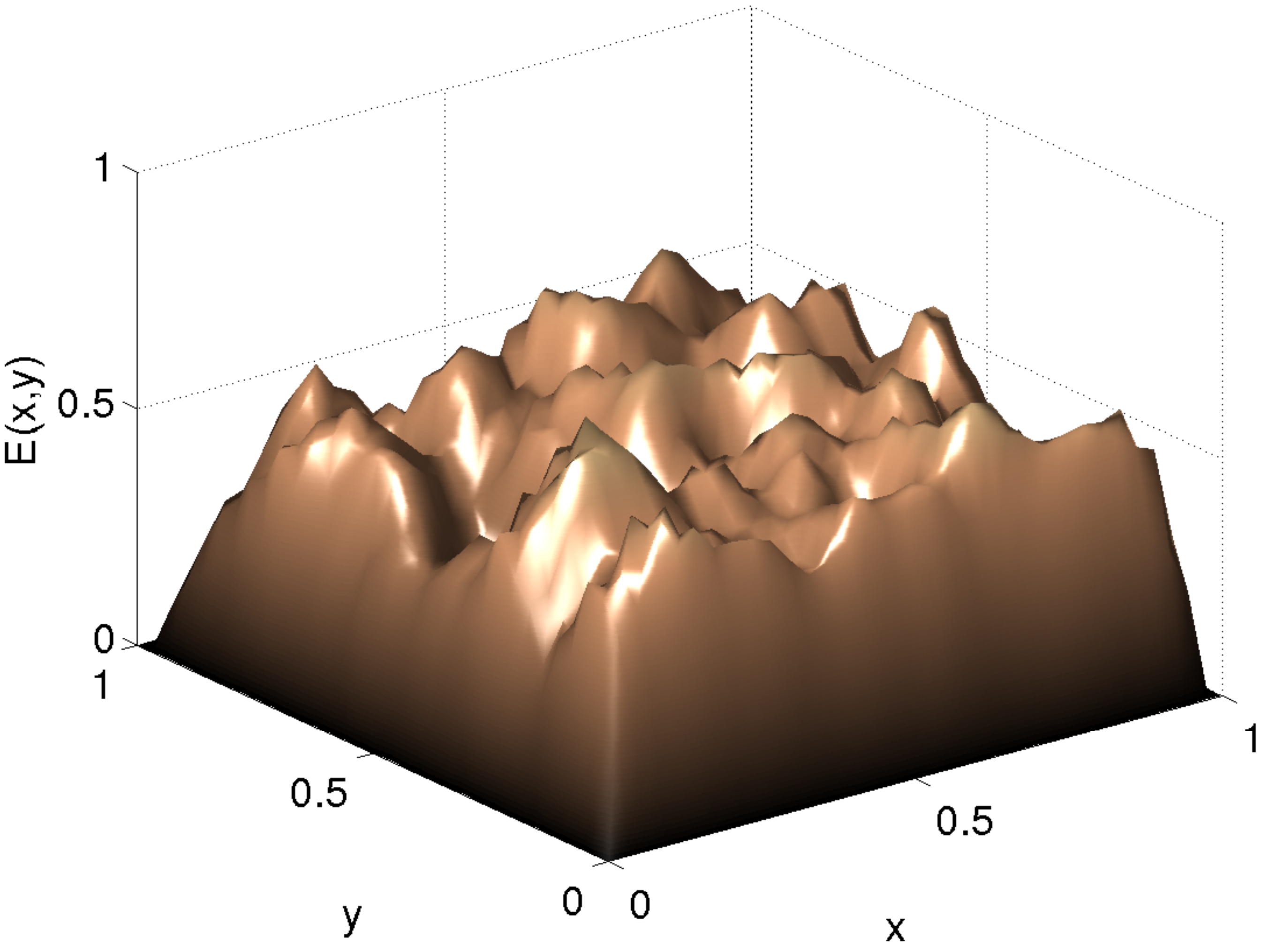
Error after 1 weighted Jacobi iteration

# Smoothing Property



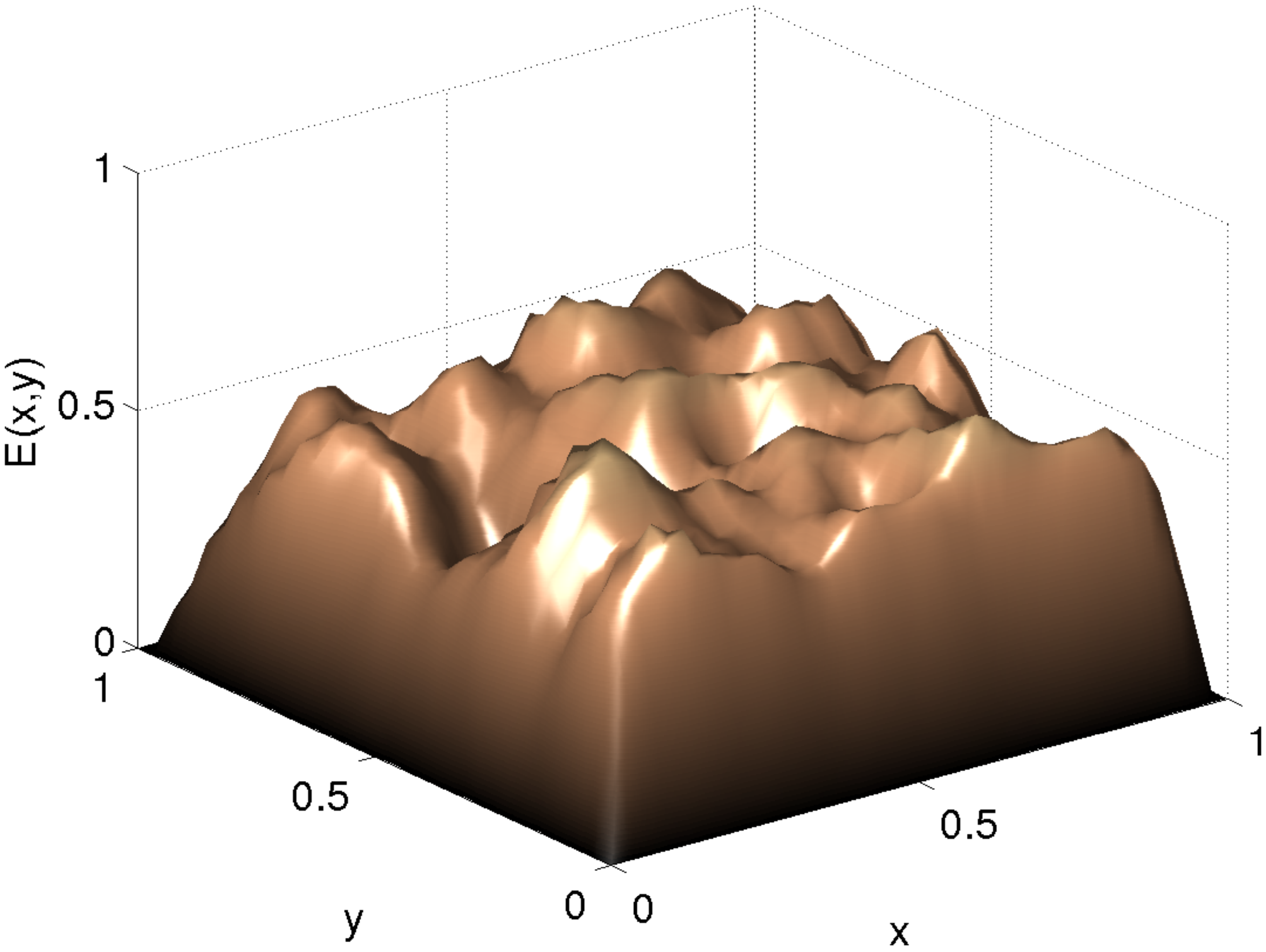
Error after 2 weighted Jacobi iterations

# Smoothing Property



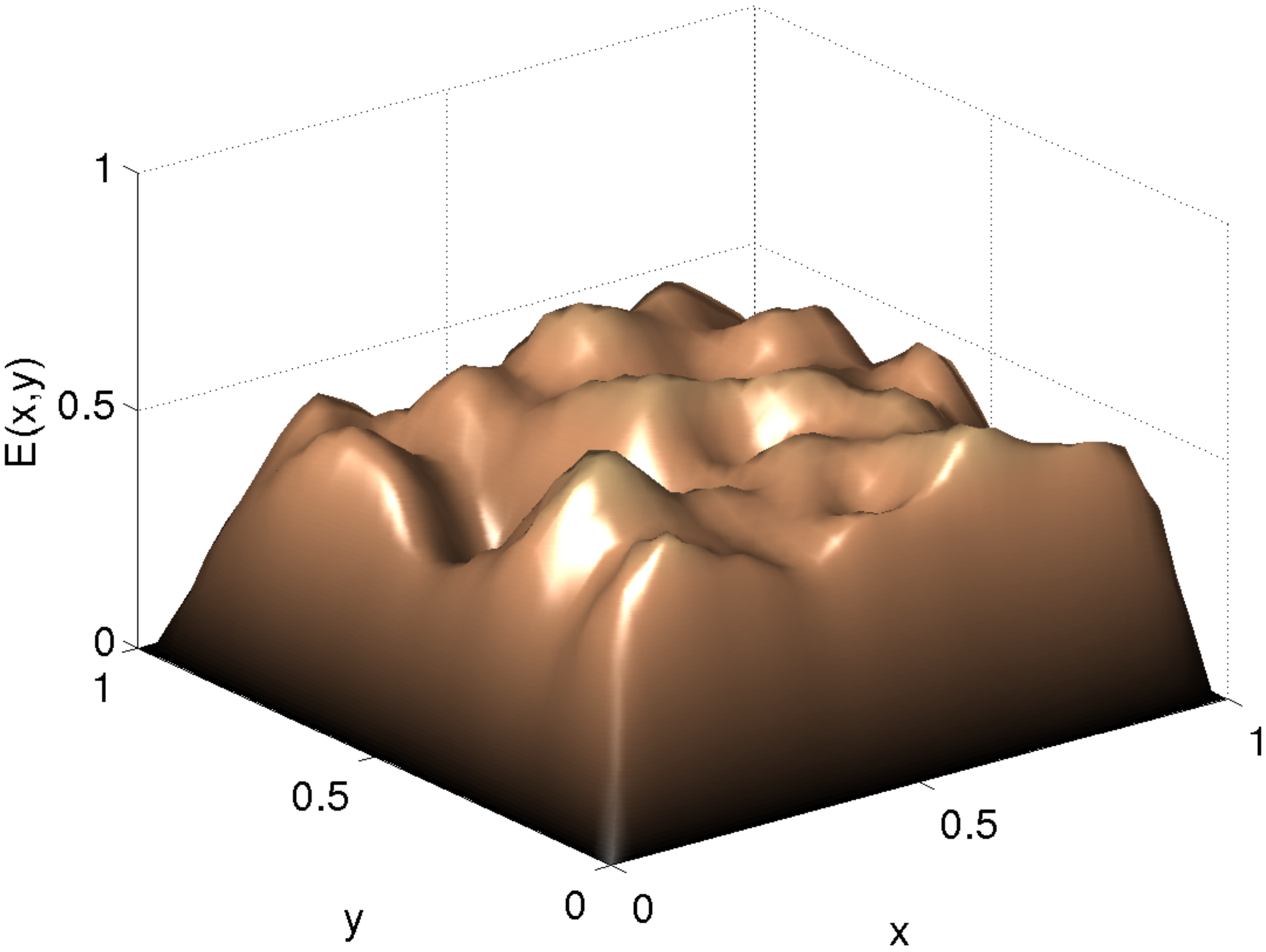
Error after 3 weighted Jacobi iterations

# Smoothing Property



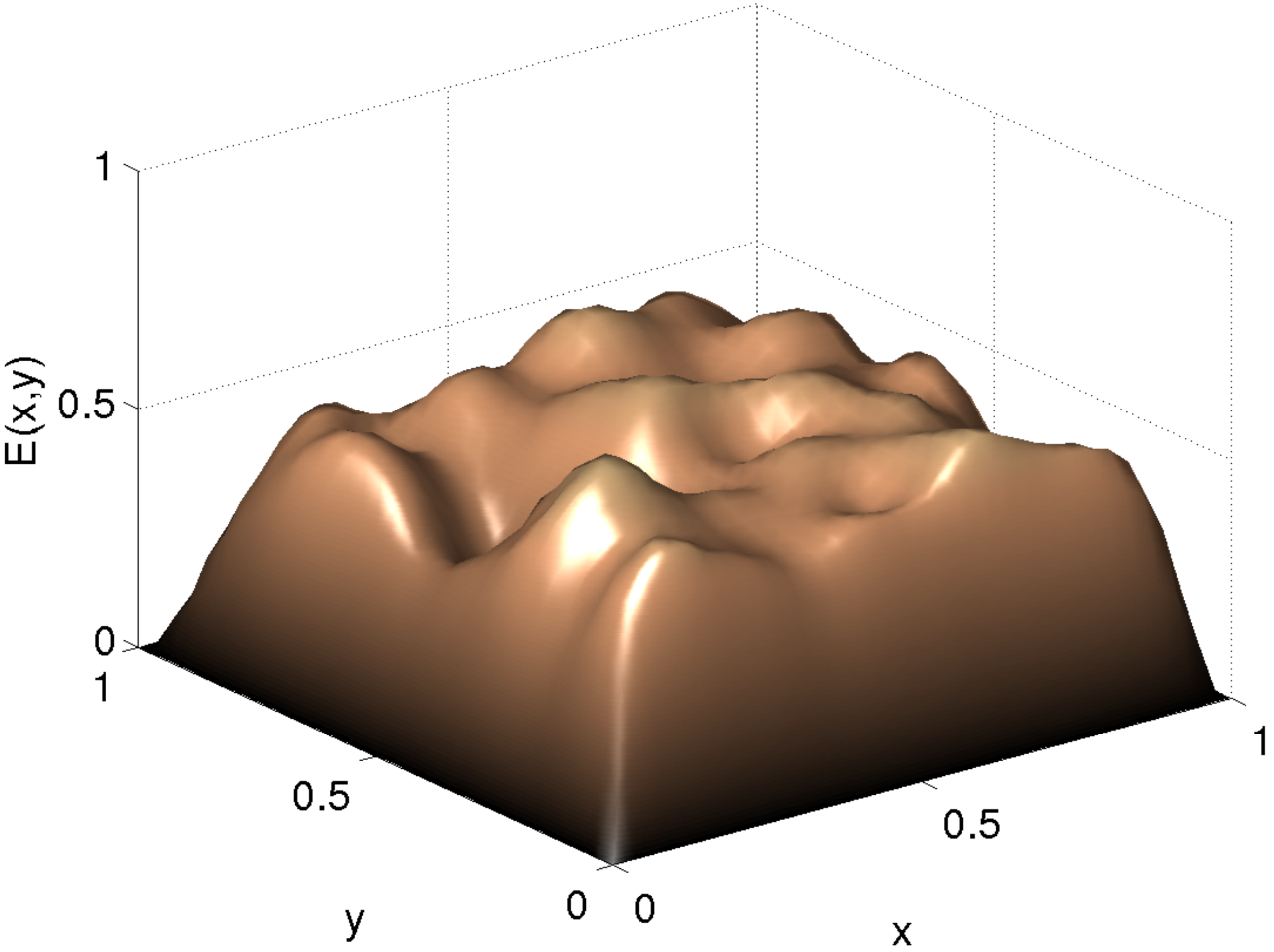
Error after 4 weighted Jacobi iterations

# Smoothing Property



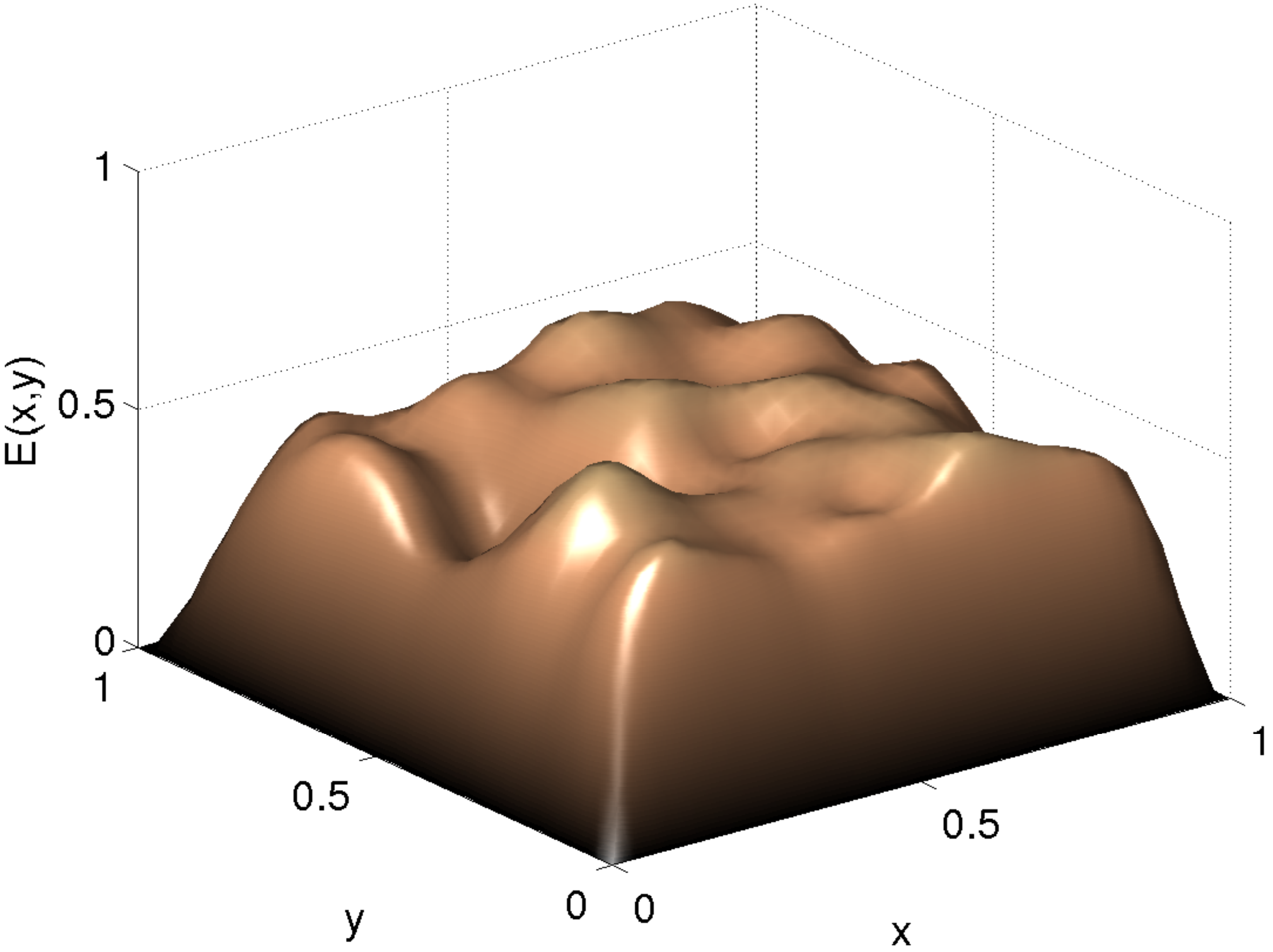
Error after 5 weighted Jacobi iterations

# Smoothing Property



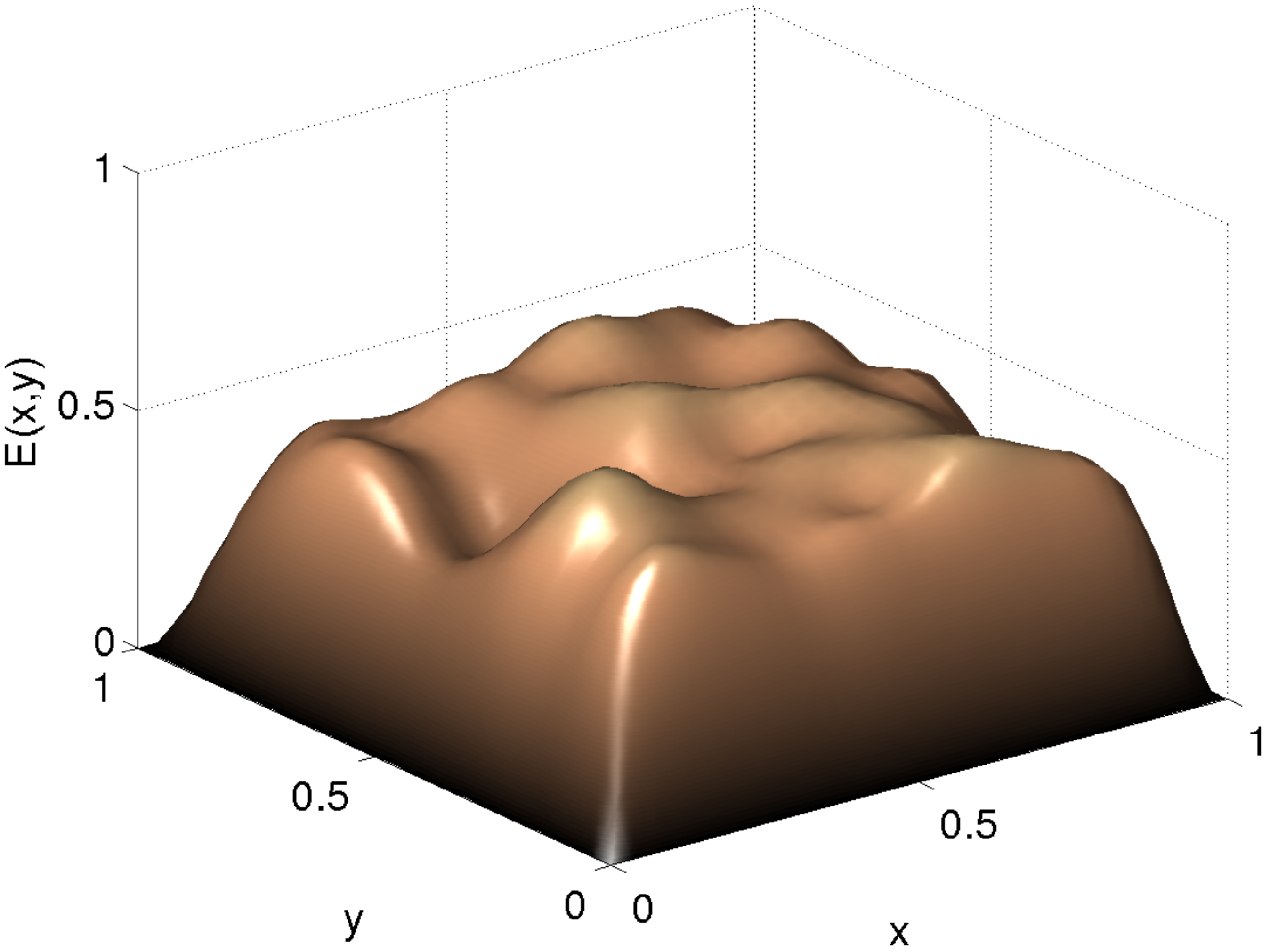
Error after 6 weighted Jacobi iterations

# Smoothing Property



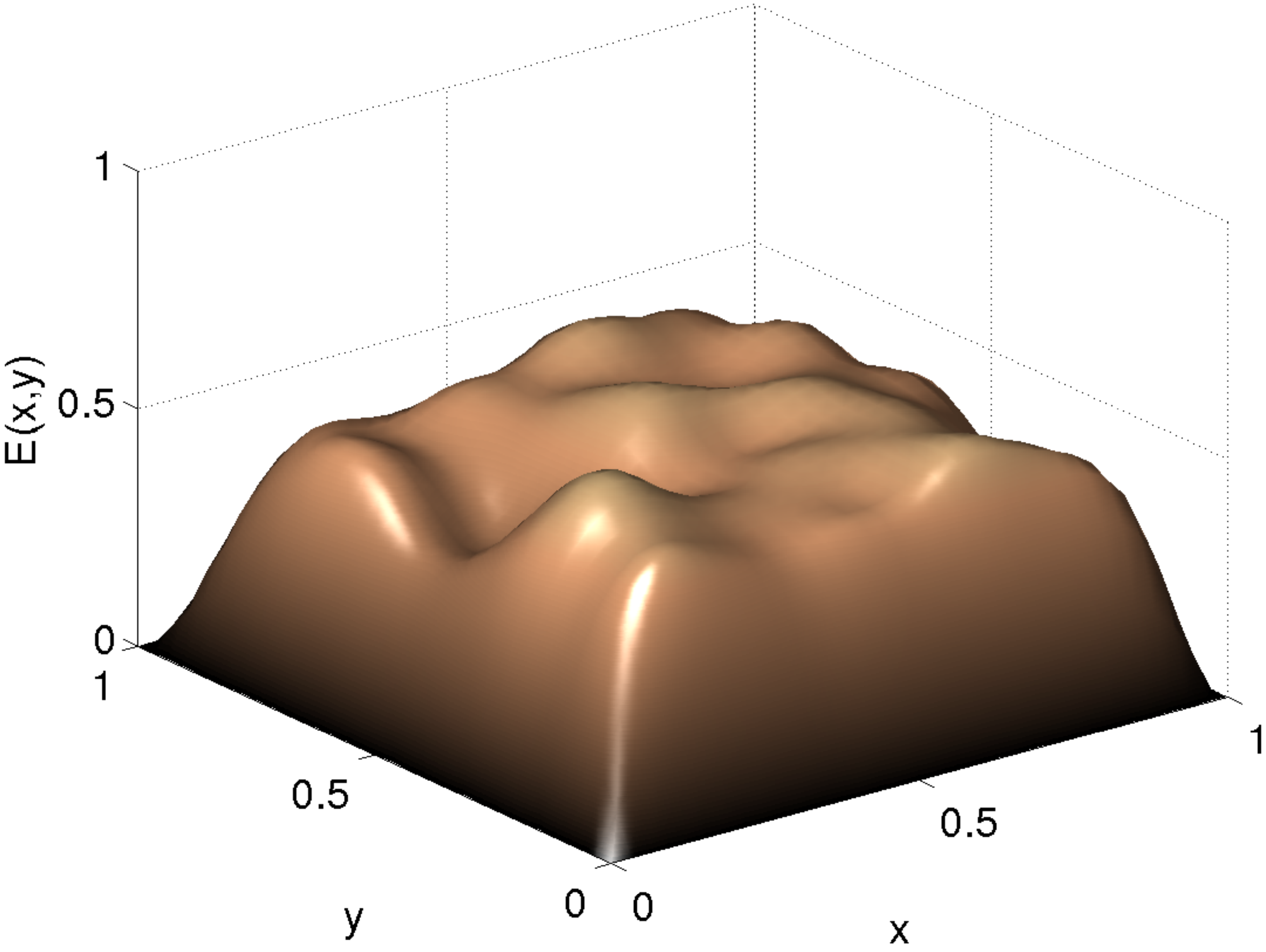
Error after 7 weighted Jacobi iterations

# Smoothing Property



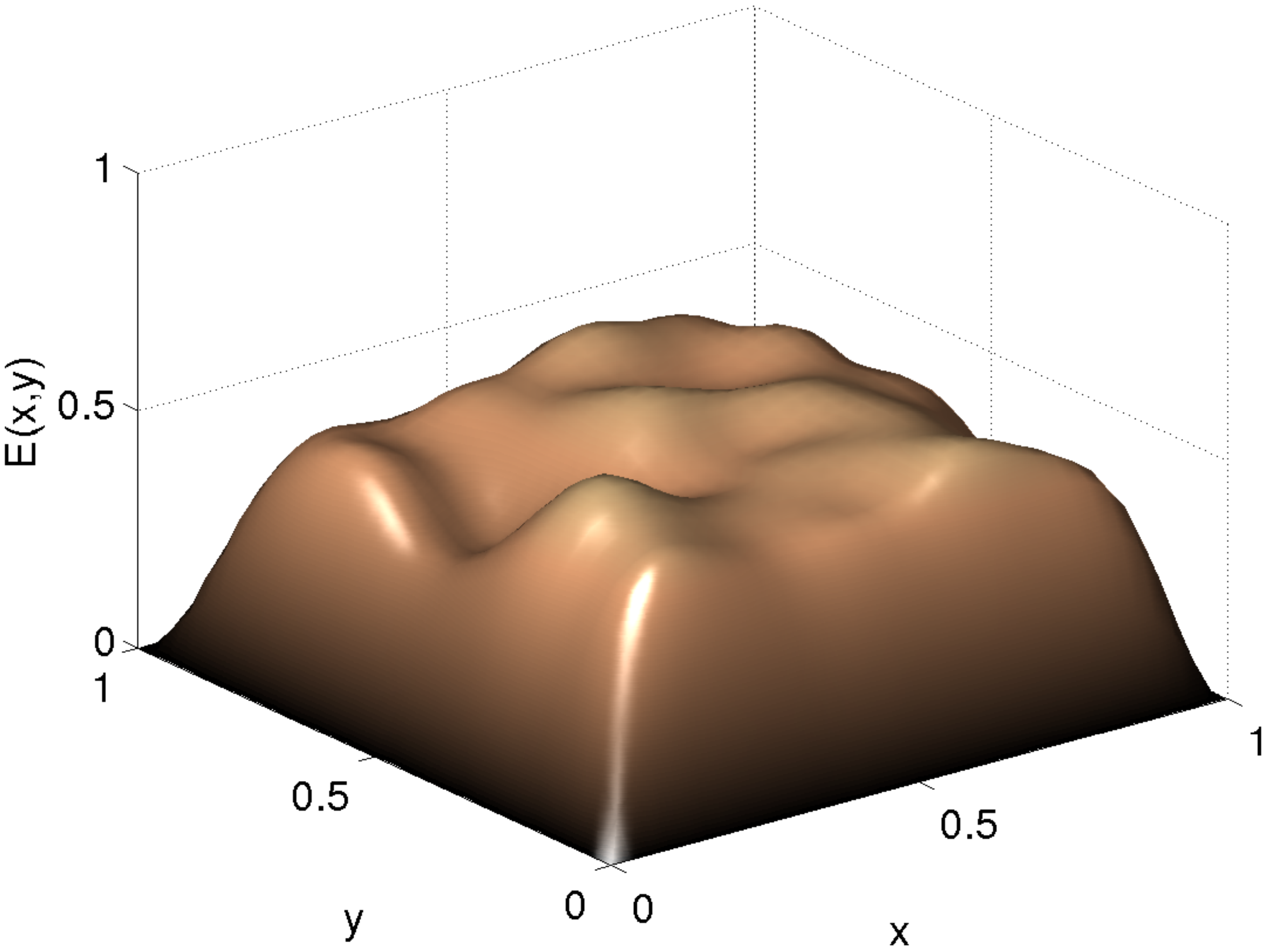
Error after 8 weighted Jacobi iterations

# 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 accurately represented using fewer degrees of freedom

# Coarse Grids

- Sine series representation:

$$f(x) = \sum_{k=1}^{\infty} c_k \sin(k\pi x)$$

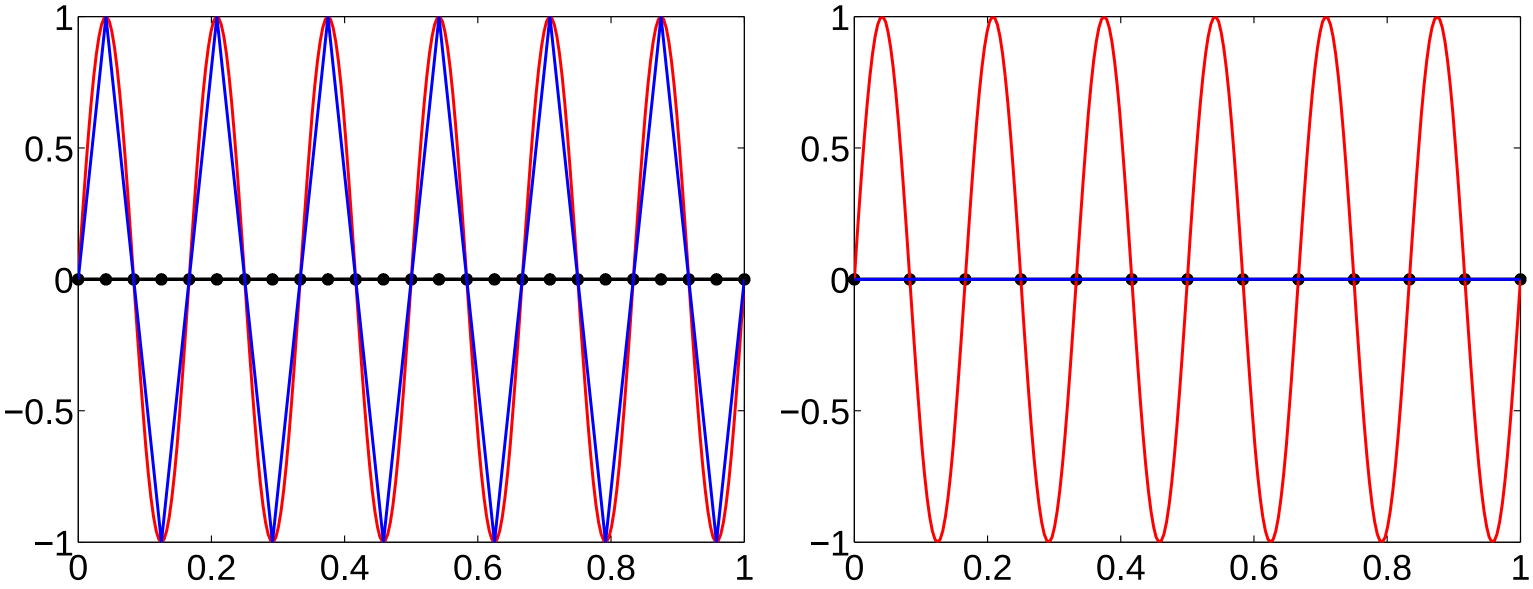
- Discrete problems can only approximate certain modes

# Coarse Grids

- Sine series representation:

$$f(x) = \sum_{k=1}^{\infty} c_k \sin(k\pi x)$$

- Discrete problems can only approximate certain modes



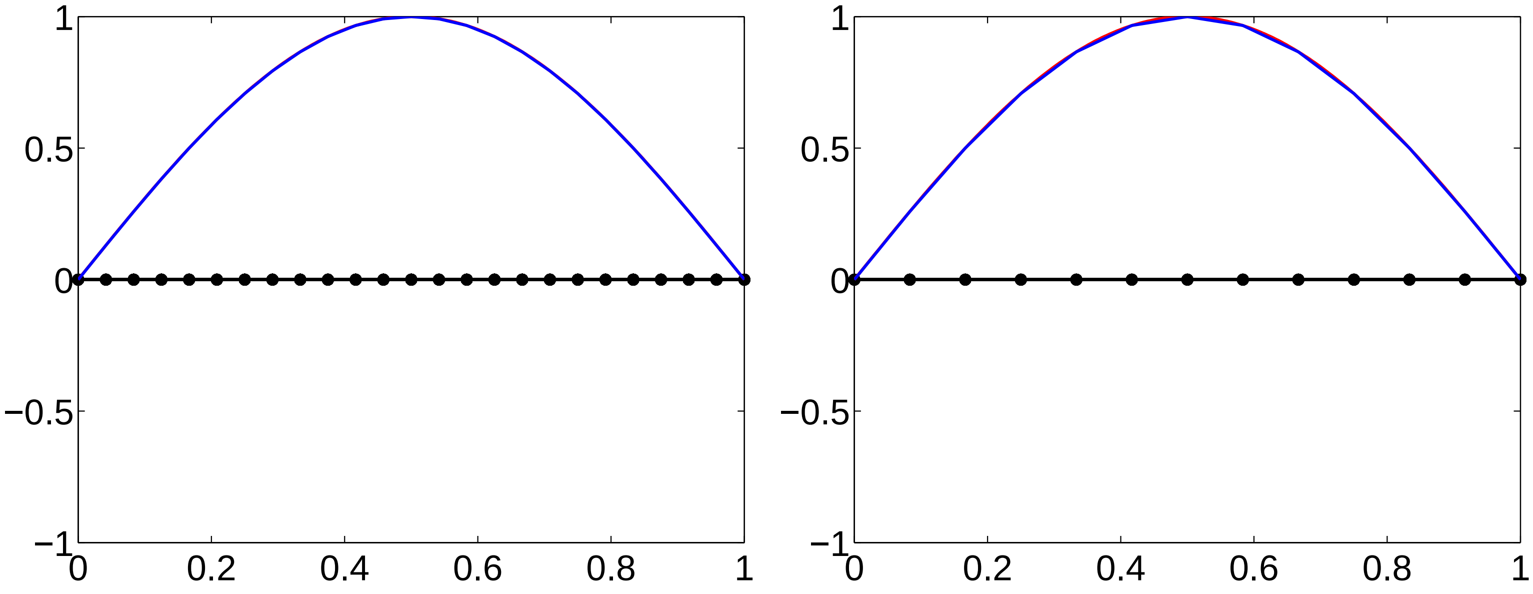
$$\sin(12\pi x)$$

# Coarse Grids

- Sine series representation:

$$f(x) = \sum_{k=1}^{\infty} c_k \sin(k\pi x)$$

- Discrete problems can only approximate certain modes



$\sin(\pi x)$

# 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

- Idea is 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?

# $A$-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:

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

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

# Two-grid cycle

# Two-grid cycle

Multigrid Components

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

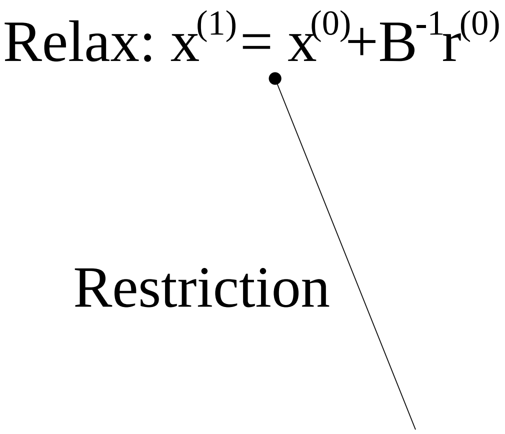
- Relaxation

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

# Two-grid cycle

Multigrid Components

- Relaxation
- Restriction



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

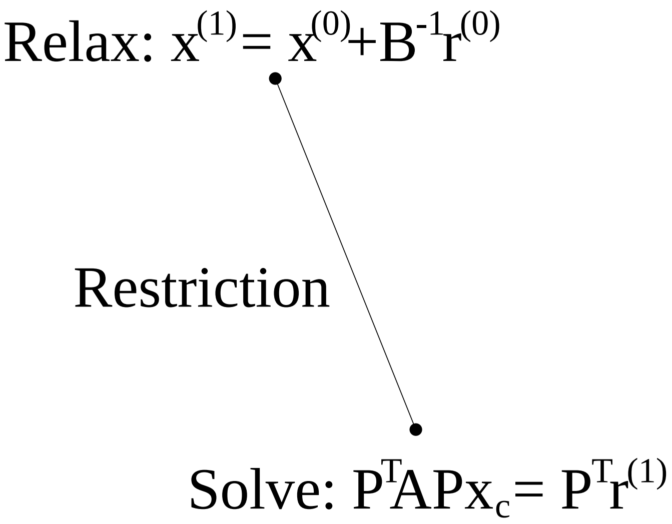
Restriction

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

# Two-grid cycle

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction



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

Restriction

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

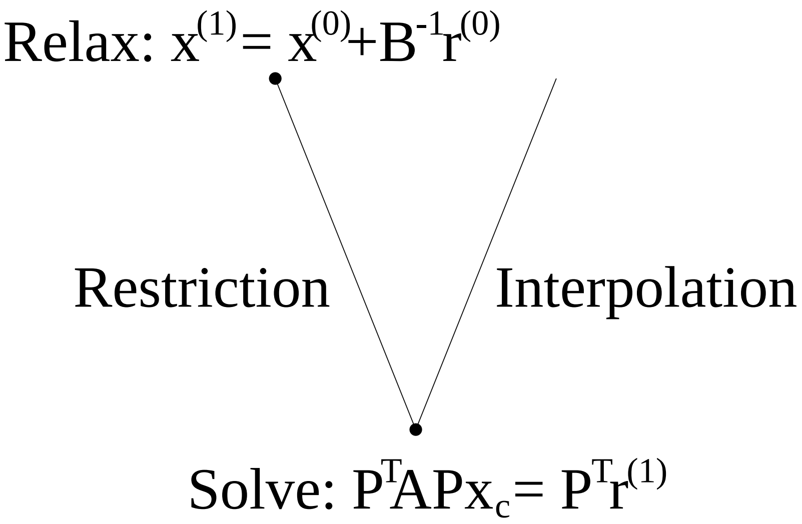
- Use coarse-grid correction to eliminate smooth errors
- Best correction, $x_c$, in terms of $A$-norm satisfies

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

# Two-grid cycle

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation



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

Restriction

Interpolation

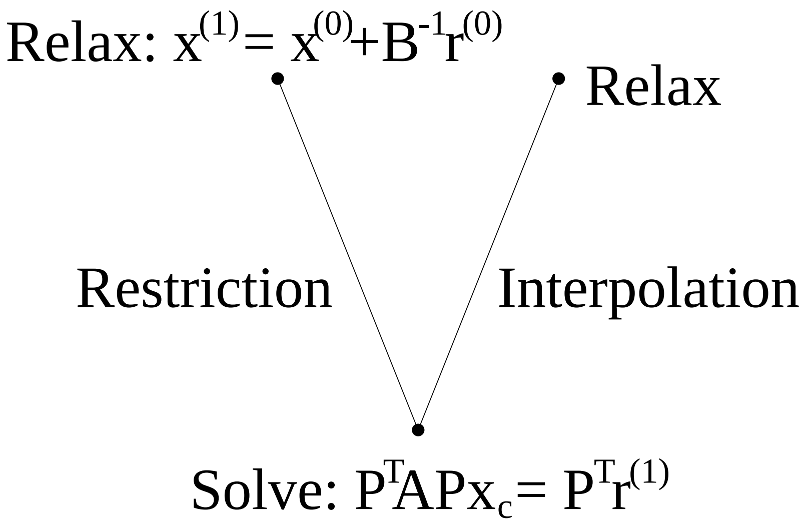
Solve: $P^T A P x_c = P^T r^{(1)}$

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

# Two-grid cycle

Multigrid Components

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



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

Relax

Restriction

Interpolation

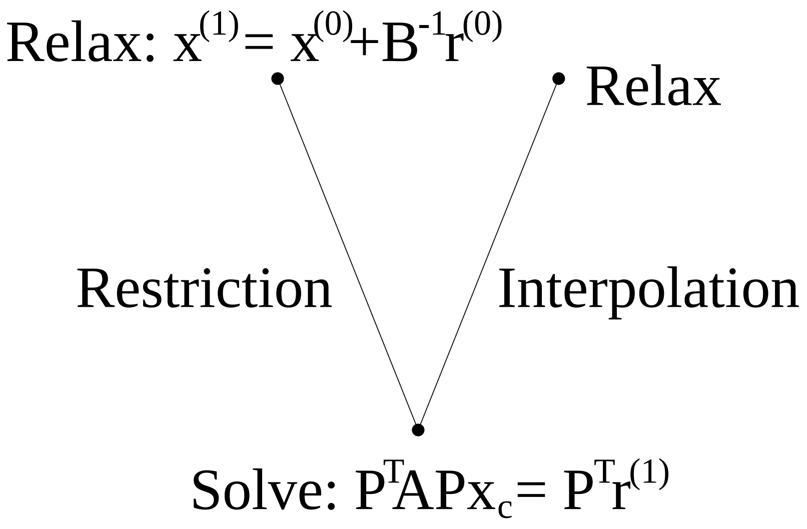
Solve: $P^TAPx_c = P^Tr^{(1)}$

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

# Two-grid cycle

## Multigrid Components

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



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

Relax

Restriction

Interpolation

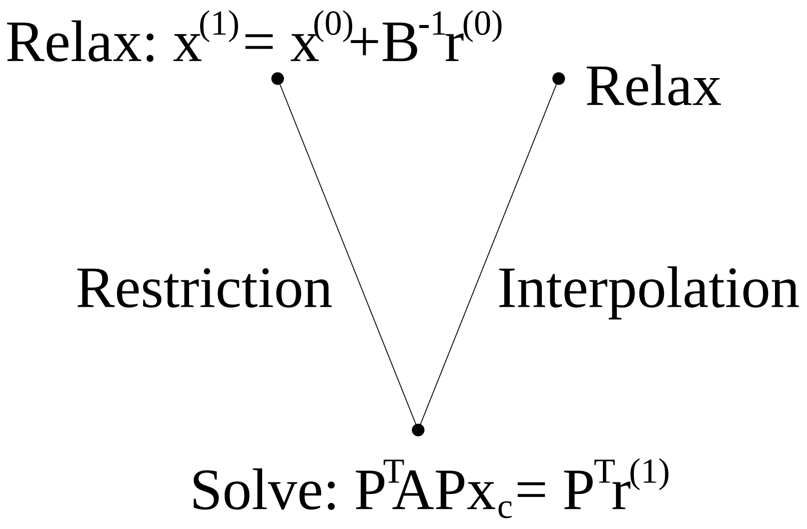
Solve: $P^T A P x_c = P^T r^{(1)}$

Direct solution of coarse-grid problem isn’t practical

# Two-grid cycle

Multigrid Components

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



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

Relax

Restriction

Interpolation

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

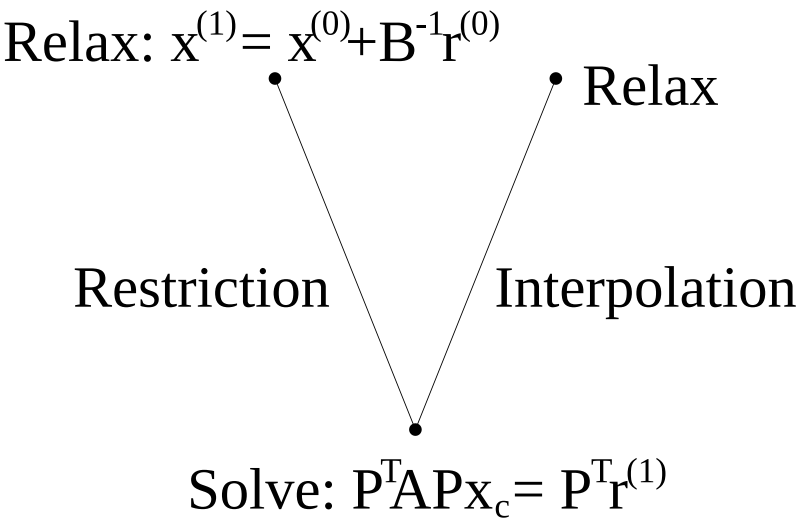
Direct solution of coarse-grid problem isn’t practical

Use an iterative method!

# Two-grid cycle

Multigrid Components

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



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

Relax

Restriction

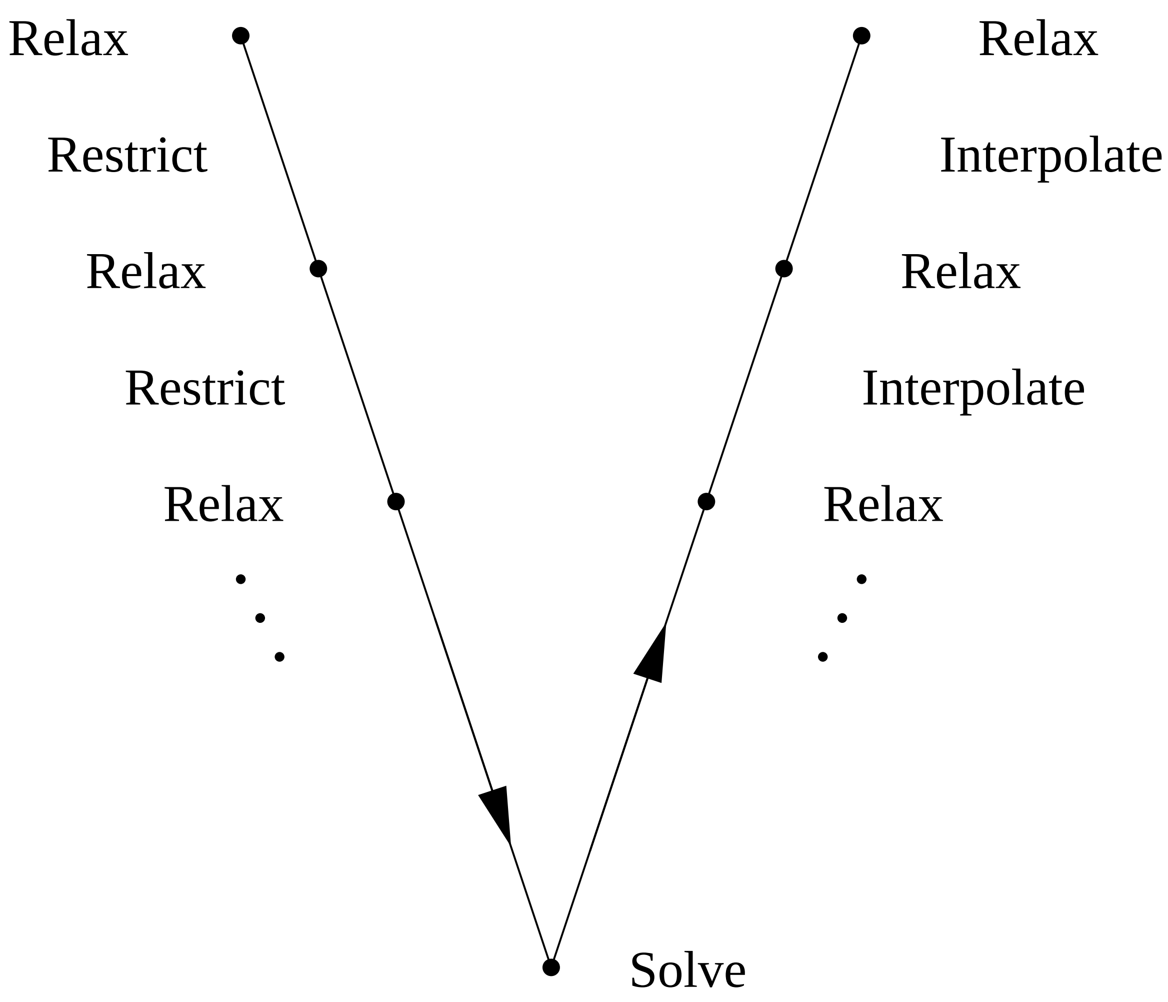
Interpolation

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

Recursion!

Apply same methodology to solve coarse-grid problem

# The Multigrid V-cycle



# Properties of Effective Cycles

- Fast convergence
  - Effective reduction of all error components
  - On each level, coarse-grid correction must effectively reduce exactly those errors that are slow to be reduced by relaxation alone
  - Hierarchy of coarse-grid operators resolves relevant physics at each scale
- Low iteration cost
  - Simple relaxation scheme (cheap computation of $B^{-1}r$ on all levels)
  - Sparse coarse-grid operators (cheap computation of residuals on all levels)
  - Sparse interpolation/restriction operations

# What Haven't I Told You?

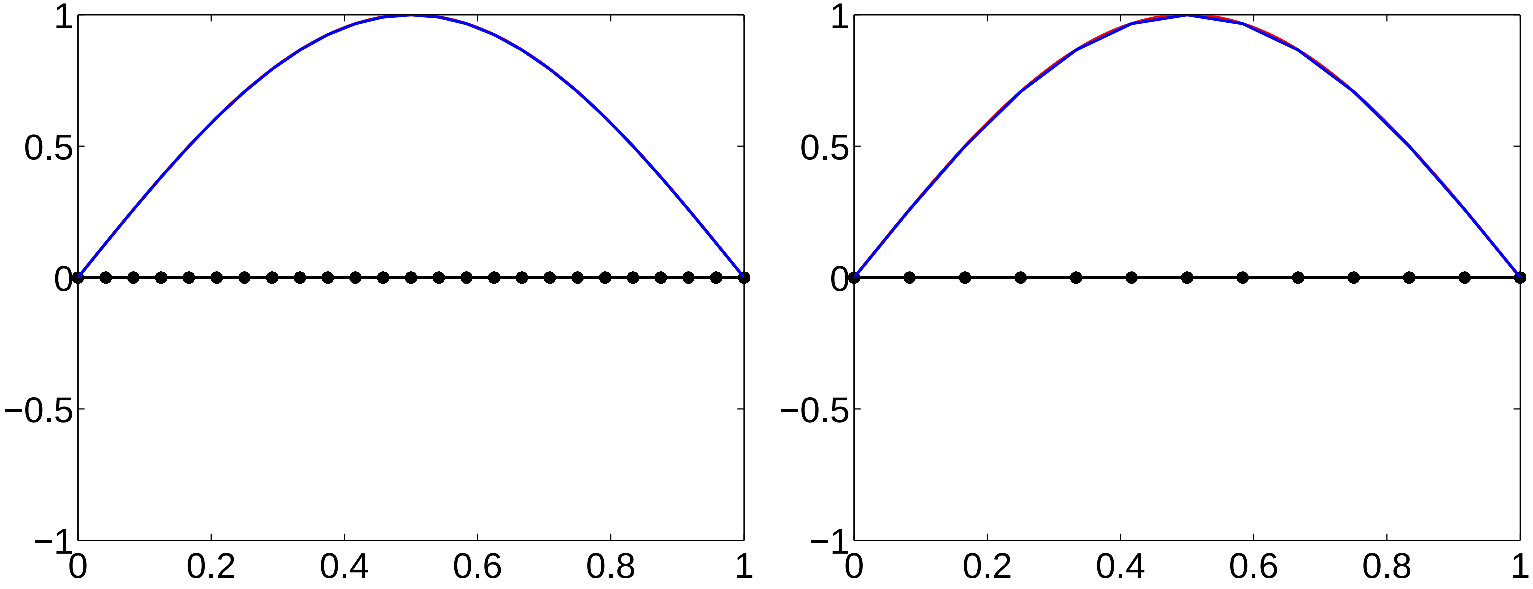
- How do we choose $P$?
  - Number of columns
  - Sparsity structure
  - Non-zero values
- Often consider these independently, but there are dependencies
- These choices *must* be informed by properties of relaxation

# Geometric Interpolation

- For Poisson’s equation, error left after relaxation is smooth
- Low-order geometric interpolation is accurate for smooth functions

# Geometric Interpolation

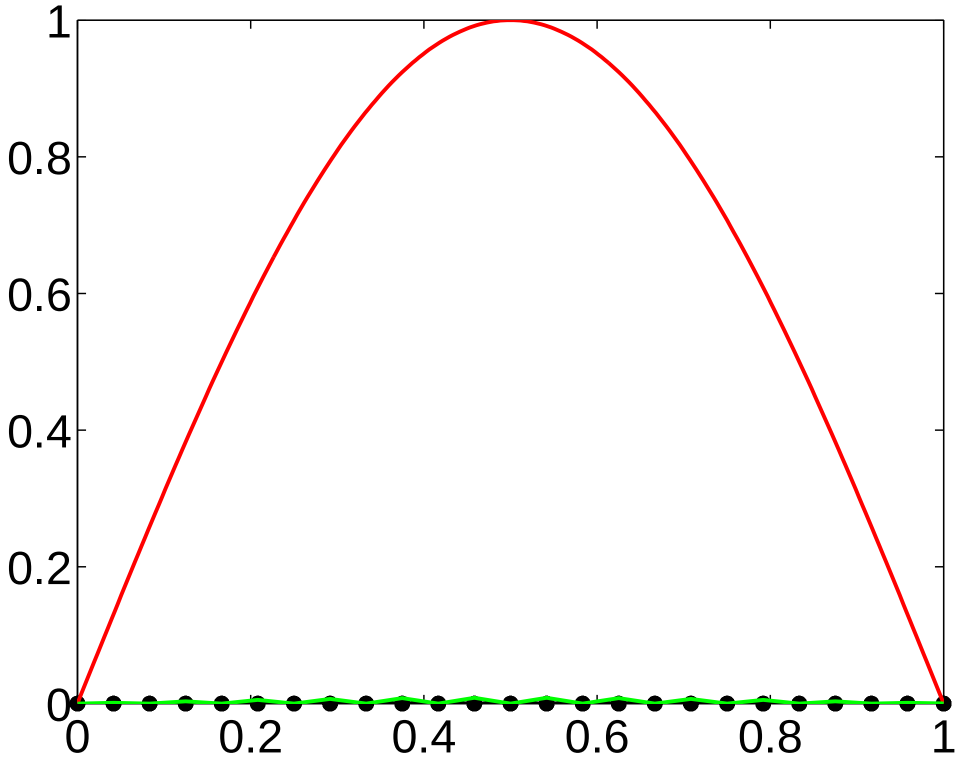
- For Poisson’s equation, error left after relaxation is smooth
- Low-order geometric interpolation is accurate for smooth functions



Approximations of $\sin(\pi x)$

# Geometric Interpolation

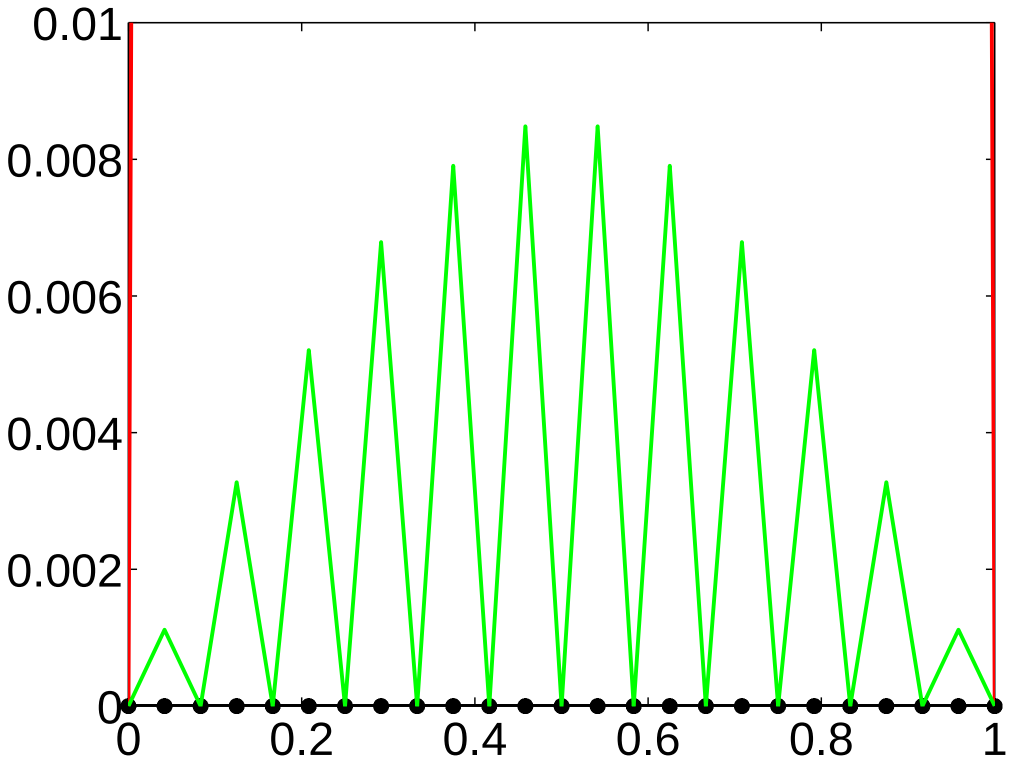
- For Poisson’s equation, error left after relaxation is smooth
- Low-order geometric interpolation is accurate for smooth functions



Interpolation Error for $\sin(\pi x)$

# Geometric Interpolation

- For Poisson’s equation, error left after relaxation is smooth
- Low-order geometric interpolation is accurate for smooth functions



Interpolation Error for $\sin(\pi x)$

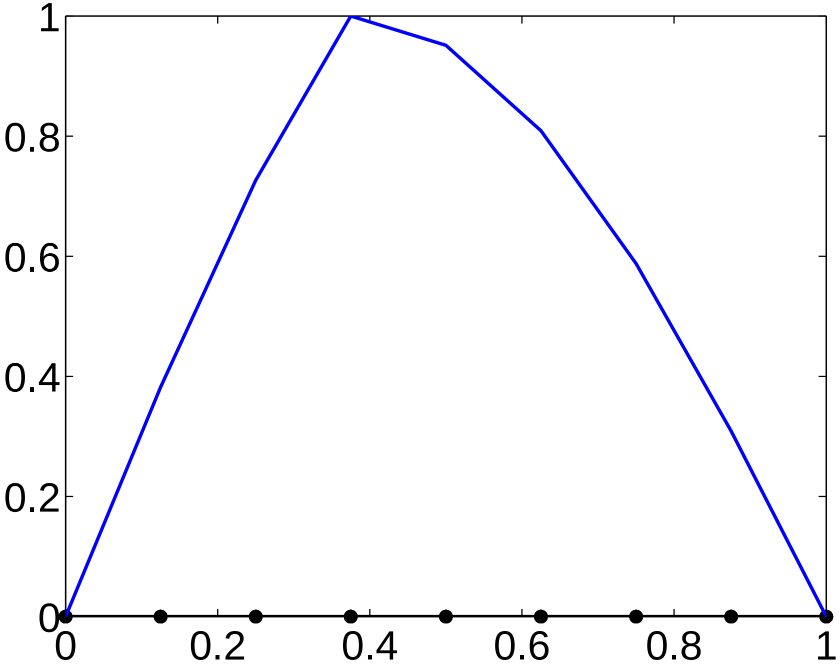
# Geometric Interpolation

- For Poisson’s equation, error left after relaxation is smooth
- Low-order geometric interpolation is accurate for smooth functions
- Linear interpolation works well for problems with smooth, isotropic coefficients when grid geometry is known
    - May not know grid geometry
    - Linear interpolation can make O(1) errors for problems with non-smooth coefficients

# Operator-Induced Interpolation

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

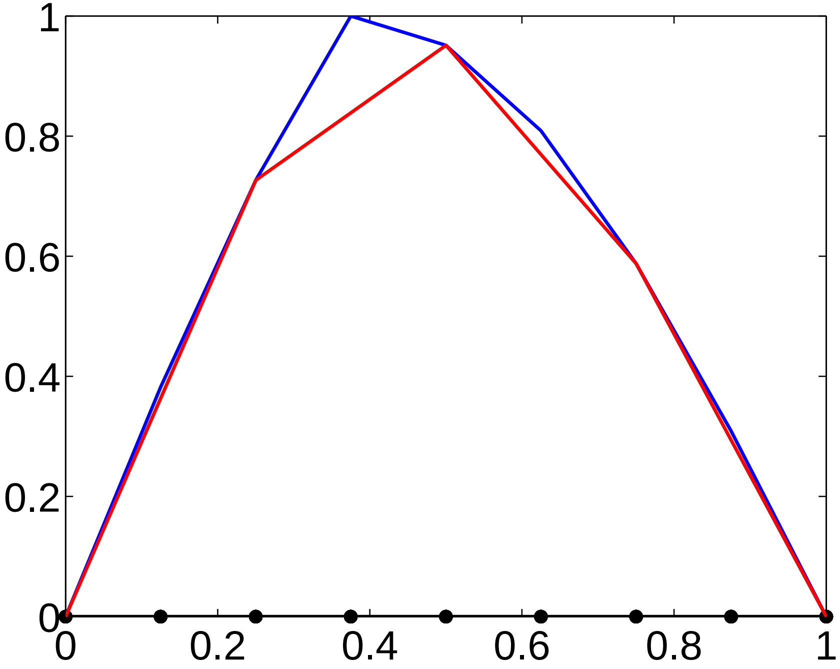


# Operator-Induced Interpolation

- 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



# Operator-Induced Interpolation

- 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

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

# Operator-Induced Interpolation

- 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
- No geometric information used in defining interpolation
- Start with small-residual 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 $k \in C \cap \{k : a_{ik} \neq 0\}$

# Calibrating Interpolation

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

# 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

# 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 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 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 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

# Adaptive Cycling

- Suppose we probe relaxation, design the best interpolation we know how, and the resulting MG cycle still doesn't work.
  - Interpolation was based on a single prototype of slow-to-converge errors
  - May not have enough information to complement all slow-to-converge modes
  - How can we identify a new prototype, distinct from the previous?

Apply the adaptive principles to the multigrid method itself

# 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 $\|B^{-1}A\|$ using Rayleigh Quotients

# Algorithm Overview

- while $\|I - B_{\text{MG}}^{-1}A\|_{\text{est}}$ is large
    - if $\|I - B_{\text{rel}}^{-1}A\|_{\text{est}}$ is increasing
        - iterate on $Ax = 0$ with “relaxation”, $x \leftarrow (I - B_{\text{rel}}^{-1}A)x$
        - 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: $B_{\text{rel}} \leftarrow B_{\text{MG}}$

# Testing Adaptation

- 2-D Finite Element Shifted Laplacian, Dirichlet BCs, $512 \times 512$ grid

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

- $\lambda_{\text{min}} = 6.64 \times 10^{-4}$, 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 |

# Linear Elasticity

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

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

- $\mu$, $\lambda$ are Lamé coefficients, related to Poisson Ratio, $\nu$, and Young modulus, $E$

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

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

# Numerical Results: Linear Elasticity

3D cube, 201,720 degrees of freedom, exponential distribution of $E$

| | Fixed | | | Adaptive | | |
|---|---|---|---|---|---|---|
| $\sigma$ | $\|I - B_{\text{MG}}^{-1}A\|$ | Iterations | CPU (s) | $\|I - B_{\text{MG}}^{-1}A\|$ | Iterations | CPU (s) |
| 2 | 0.1146 | 9 | 25.99 | 0.2141 | 12 | 267.72 |
| 3 | 0.2466 | 14 | 35.68 | 0.3095 | 16 | 275.62 |
| 4 | 0.3948 | 20 | 49.99 | 0.4040 | 21 | 289.39 |
| 5 | 0.5545 | 32 | 73.63 | 0.4966 | 27 | 381.16 |

# Lattice Quantum Chromodynamics

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

# Numerical Results: Lattice QCD

$128 \times 128$ periodic lattice, average residual reduction per iteration

| | Diagonal-PCG | | | | AdaptiveMG-PCG | | | |
|---|---|---|---|---|---|---|---|---|
| $\rho - \rho_{\text{cr}}$ | 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 |

# Choosing Coarse Grids

- Difficult to say what *best* coarse grid is
  - Want enough coarse-grid points so that interpolation is accurate for all slow-to-converge errors
  - Want significantly fewer coarse-grid points than fine-grid points
- Interpolation designed to complement failings of relaxation
  - Coarse-grid selection must make this possible

# Maximal Independent Sets

- Want local interpolation operators
- Idea: Coarsen so that every fine-grid node has at least one coarse-grid neighbour

Problem: not all connections are equal

- Really want every fine-grid node to be somehow strongly coupled to at least one coarse-grid node

# Strong Connections - AMG

- Classical AMG defines the strong connections for each node based directly on the matrix entries:

$$S_i = \left\{ j : -a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\} \right\}$$

- Based on properties of finite difference discretizations
  - Diagonally dominant M-matrices

# Weaknesses

- Definition of strong connections based on “nice” M-matrix properties
- Breaks down if near null space of $A$ is far from the constant
  - Diagonal rescaling,

$$A \rightarrow DAD$$

  - Finite element anisotropy,

$$-u_{xx} - \epsilon u_{yy} \rightarrow \frac{1}{6}\begin{bmatrix} (-1-\epsilon) & (2-4\epsilon) & (-1-\epsilon) \\ (-4+2\epsilon) & (8+8\epsilon) & (-4+2\epsilon) \\ (-1-\epsilon) & (2-4\epsilon) & (-1-\epsilon) \end{bmatrix}$$

- Even for simple problems, size of $a_{ij}$ may not reflect true connection between $i$ and $j$

# What are Strong Connections?

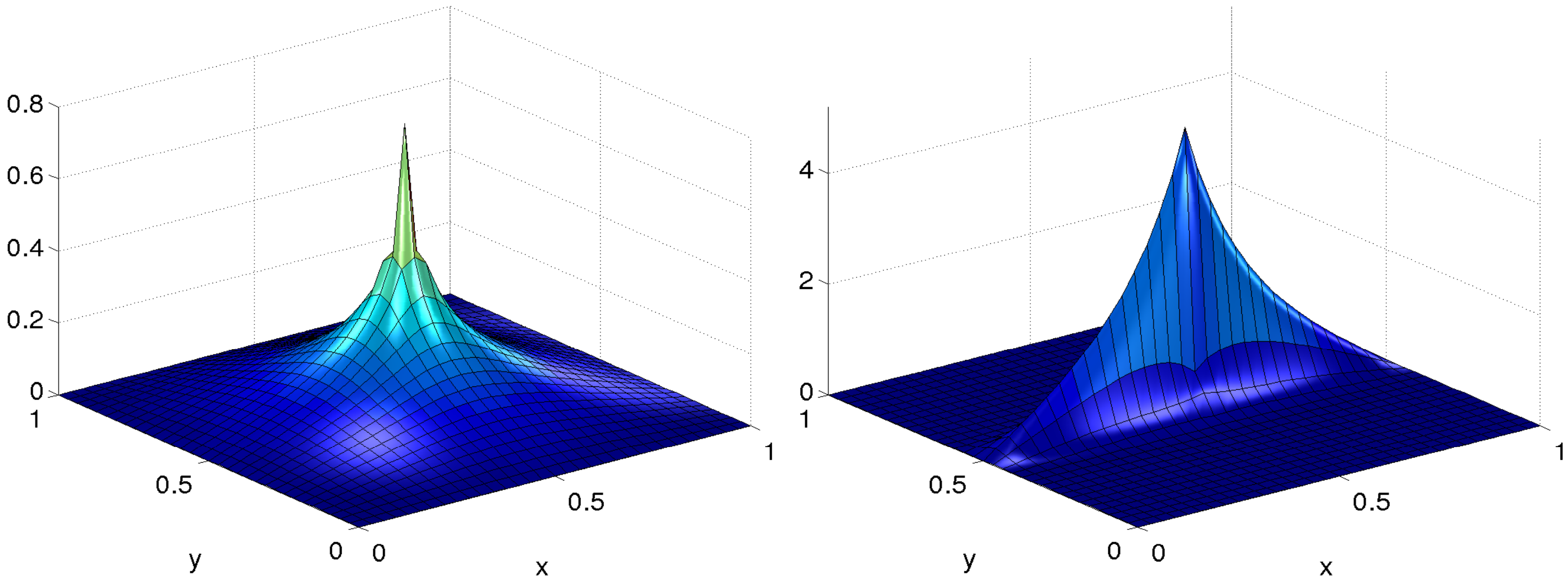
- Point $i$ strongly depends on point $j$ if
  - a change in the right-hand side at point $j$ significantly changes the solution at point $i$.
  - a change in the residual at point $j$ significantly changes the error at point $i$
- Good coarse-grid correction depends on identifying strong connections
  - Interpolation to $i$ is most effective from points that it strongly depends on
  - Corrections from weakly connected points have little effect on the error at $i$

# Inverse-based Strength

- For the discrete linear system, $Ax = b$, the inverse relates changes in $b$ to changes in $x$

$$x = (A)^{-1}\, b$$

- If a change in $b_j$ causes a significant change in $x_i$, then $(A)^{-1}_{ij}$ must be large relative to other values of $(A)^{-1}_{ik}$



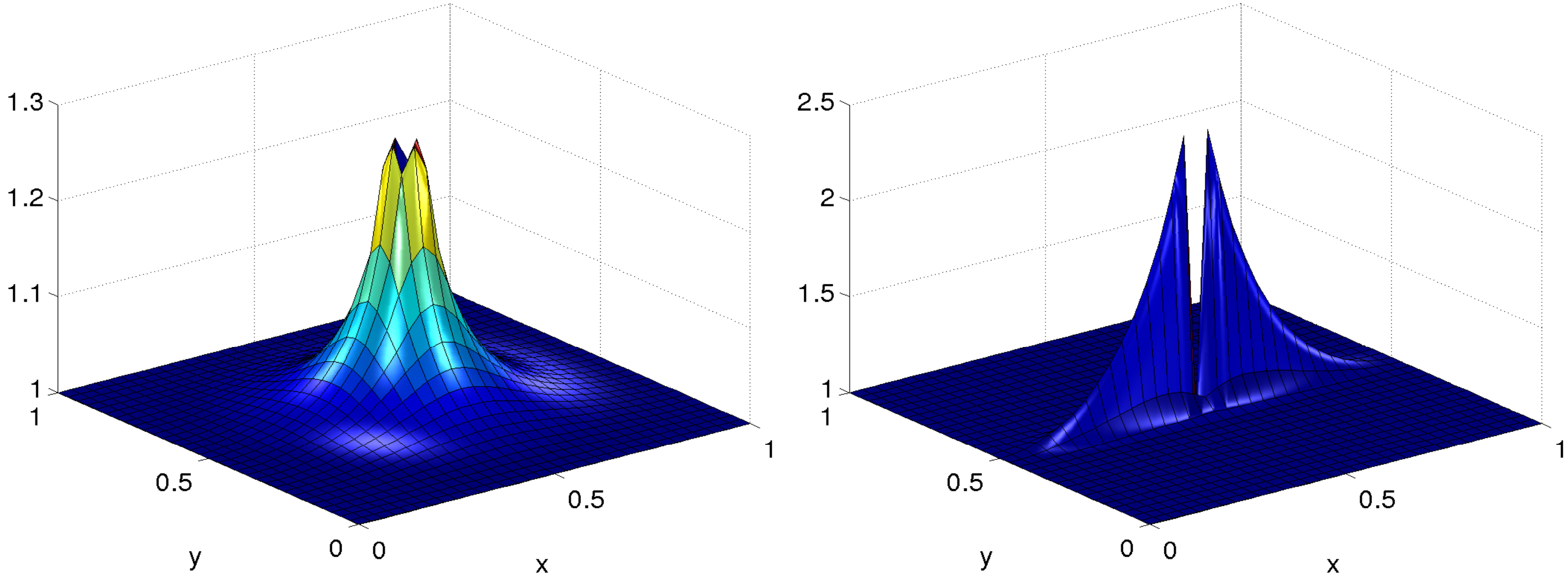
Columns of inverse of Isotropic and Anisotropic Poisson Operators

# Measures of Strong Connections

- Strength of dependence of $i$ on $j$ depends on size of $(A)^{-1}_{ij}$
- How should we measure this size, relative to $(A)^{-1}_{ik}$?

# Measures of Strong Connections

- Strength of dependence of $i$ on $j$ depends on size of $(A)^{-1}_{ij}$
- How should we measure this size, relative to $(A)^{-1}_{ik}$?
- $L^2$ measure: $(A)^{-1}_{ij} \geq \theta \max\limits_{k \neq i} \left\{ (A)^{-1}_{ik} \right\}$
- Energy measure: Let $G^{(i)}_j = (A)^{-1}_{ij}$, $S_{ij} = \dfrac{\|G^{(i)} - G^{(i)}_j e^{(j)}\|_A}{\|G^{(i)}\|_A}$



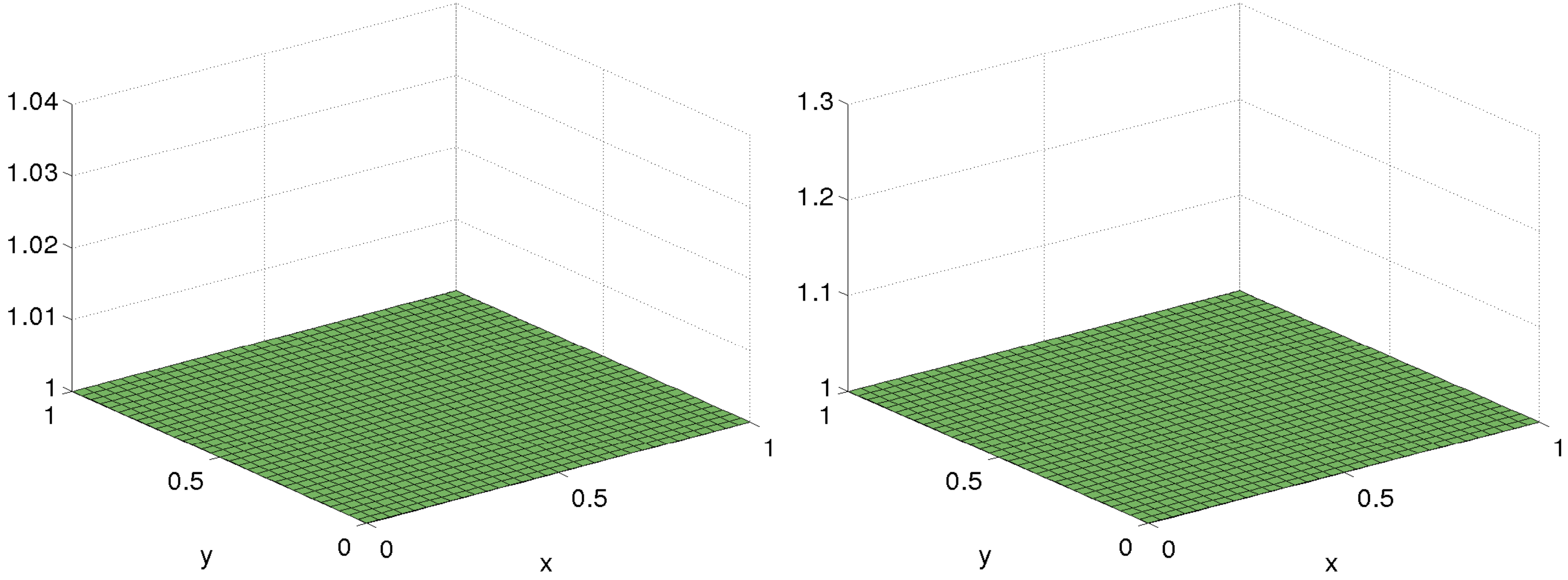
Strength measures for Isotropic and Anisotropic Poisson Operators

# Approximating $S_{ij}$

- Can we get useful, local approximations to $(A)^{-1}_{ij}$ and, thus, $S_{ij}$?
- Apply (localized) relaxation to $AG^{(i)} = e^{(i)}$

# Approximating $S_{ij}$

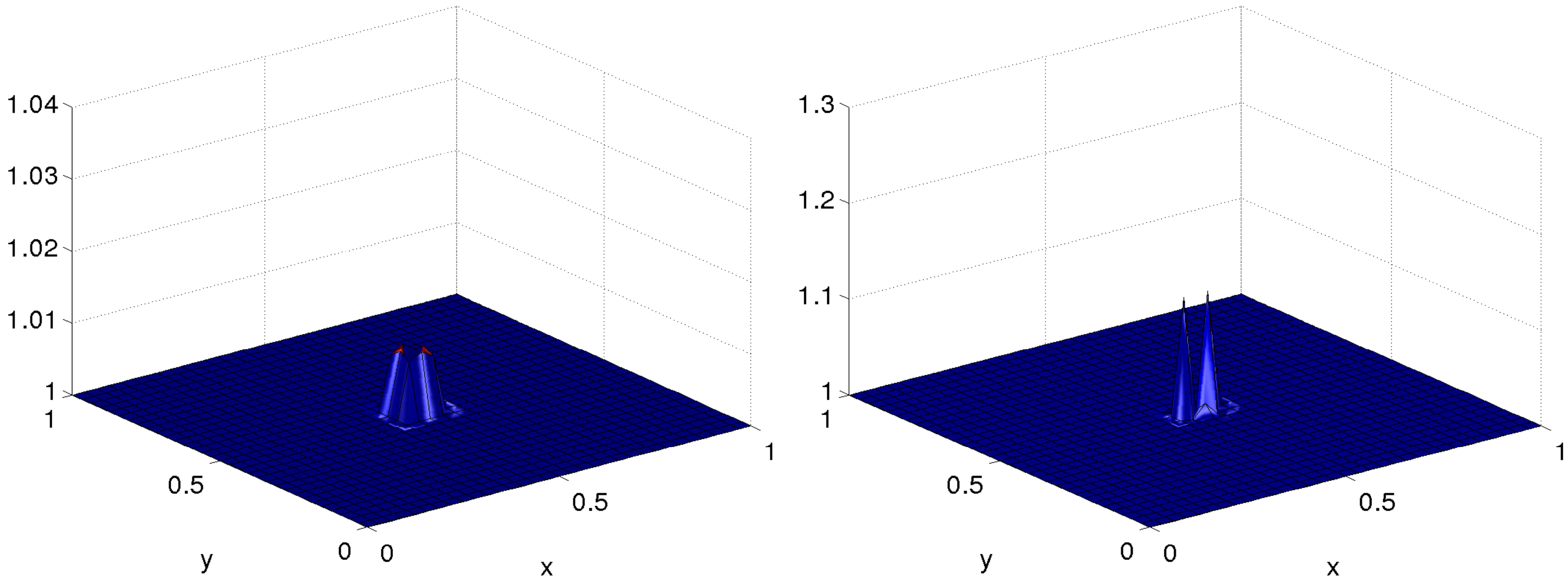
- Can we get useful, local approximations to $(A)^{-1}_{ij}$ and, thus, $S_{ij}$?
- Apply (localized) relaxation to $AG^{(i)} = e^{(i)}$



Approximate $S_{ij}$ after 1 weighted Jacobi Relaxation for Isotropic and Anisotropic Poisson Operators

# Approximating $S_{ij}$

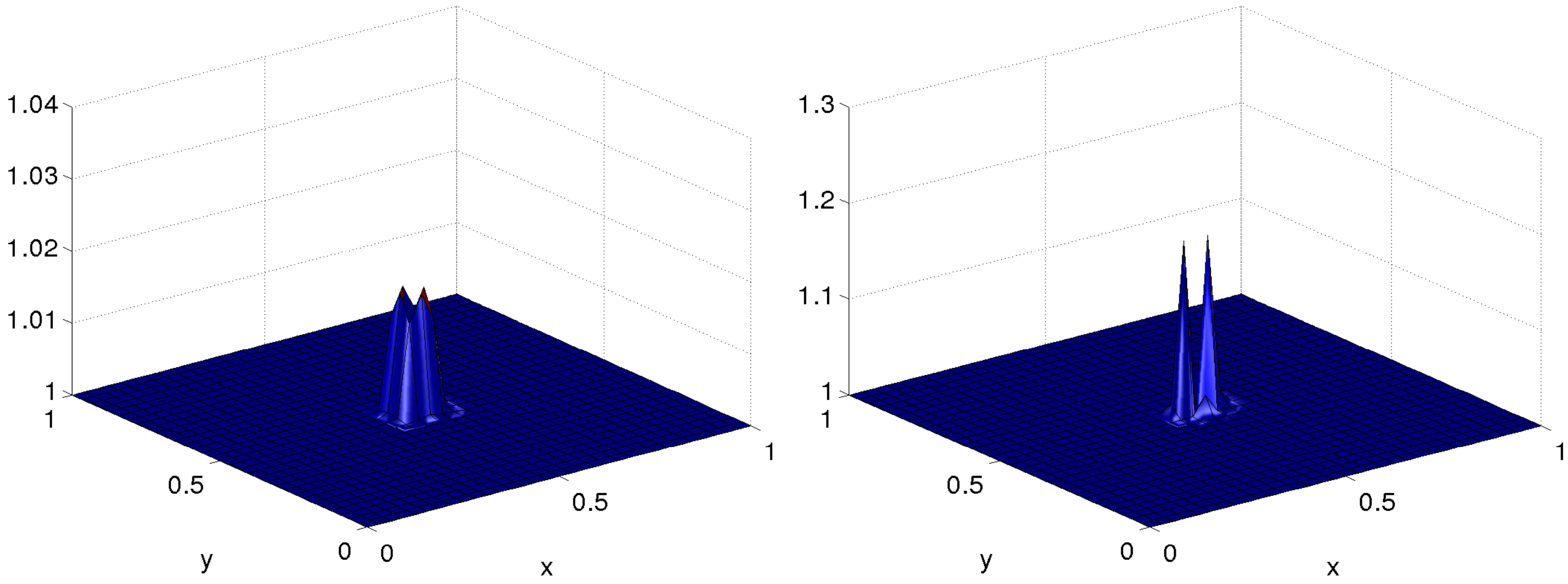
- Can we get useful, local approximations to $(A)_{ij}^{-1}$ and, thus, $S_{ij}$?
- Apply (localized) relaxation to $AG^{(i)} = e^{(i)}$



Approximate $S_{ij}$ after 2 weighted Jacobi Relaxation for Isotropic and Anisotropic Poisson Operators

# Approximating $S_{ij}$

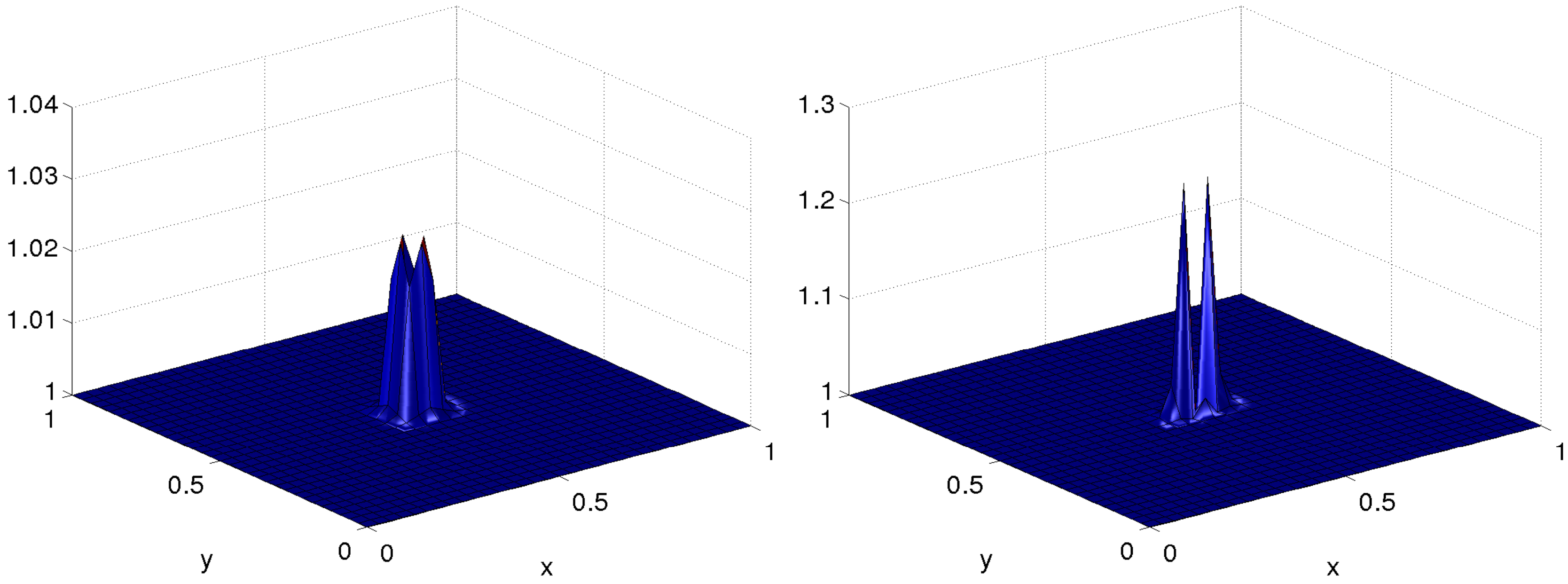
- Can we get useful, local approximations to $(A)^{-1}_{ij}$ and, thus, $S_{ij}$?
- Apply (localized) relaxation to $AG^{(i)} = e^{(i)}$



Approximate $S_{ij}$ after 3 weighted Jacobi Relaxation for Isotropic and Anisotropic Poisson Operators

# Approximating $S_{ij}$

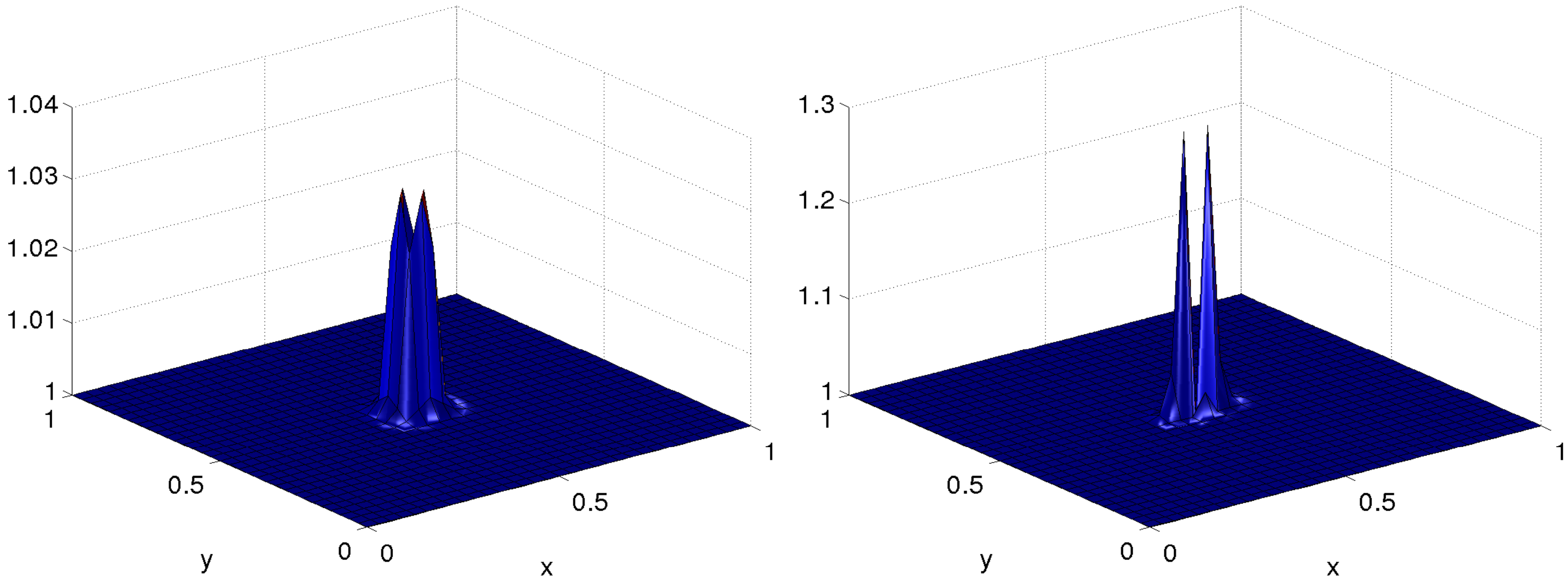
- Can we get useful, local approximations to $(A)^{-1}_{ij}$ and, thus, $S_{ij}$?
- Apply (localized) relaxation to $AG^{(i)} = e^{(i)}$



Approximate $S_{ij}$ after 4 weighted Jacobi Relaxation for Isotropic and Anisotropic Poisson Operators

# Approximating $S_{ij}$

- Can we get useful, local approximations to $(A)^{-1}_{ij}$ and, thus, $S_{ij}$?
- Apply (localized) relaxation to $AG^{(i)} = e^{(i)}$



Approximate $S_{ij}$ after 5 weighted Jacobi Relaxation for Isotropic and Anisotropic Poisson Operators

# Choosing $C$

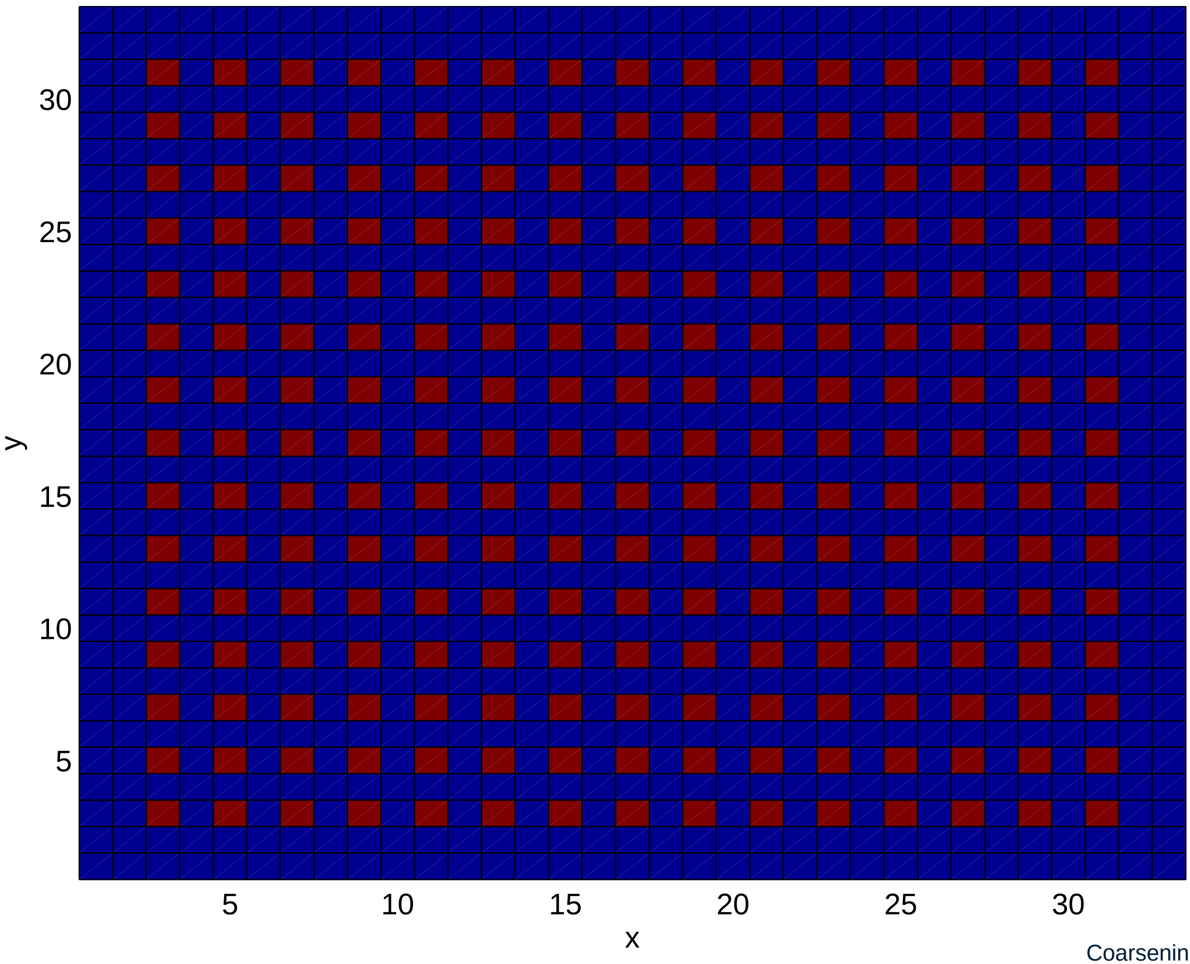
- For point $i$, $\{S_{ij}\}$ are now measures of strengths of connection
- We now say $i$ strongly depends on $j$ if $(A)_{ij} \neq 0$ and

$$S_{ij} - 1 \geq \theta \max_{k \neq i} \{S_{ik} - 1\}$$

- For now, $\theta = 0.25$ seems to work fine
- Coarse grid selection now accomplished by taking a maximal independent subset of the graph of strong connections

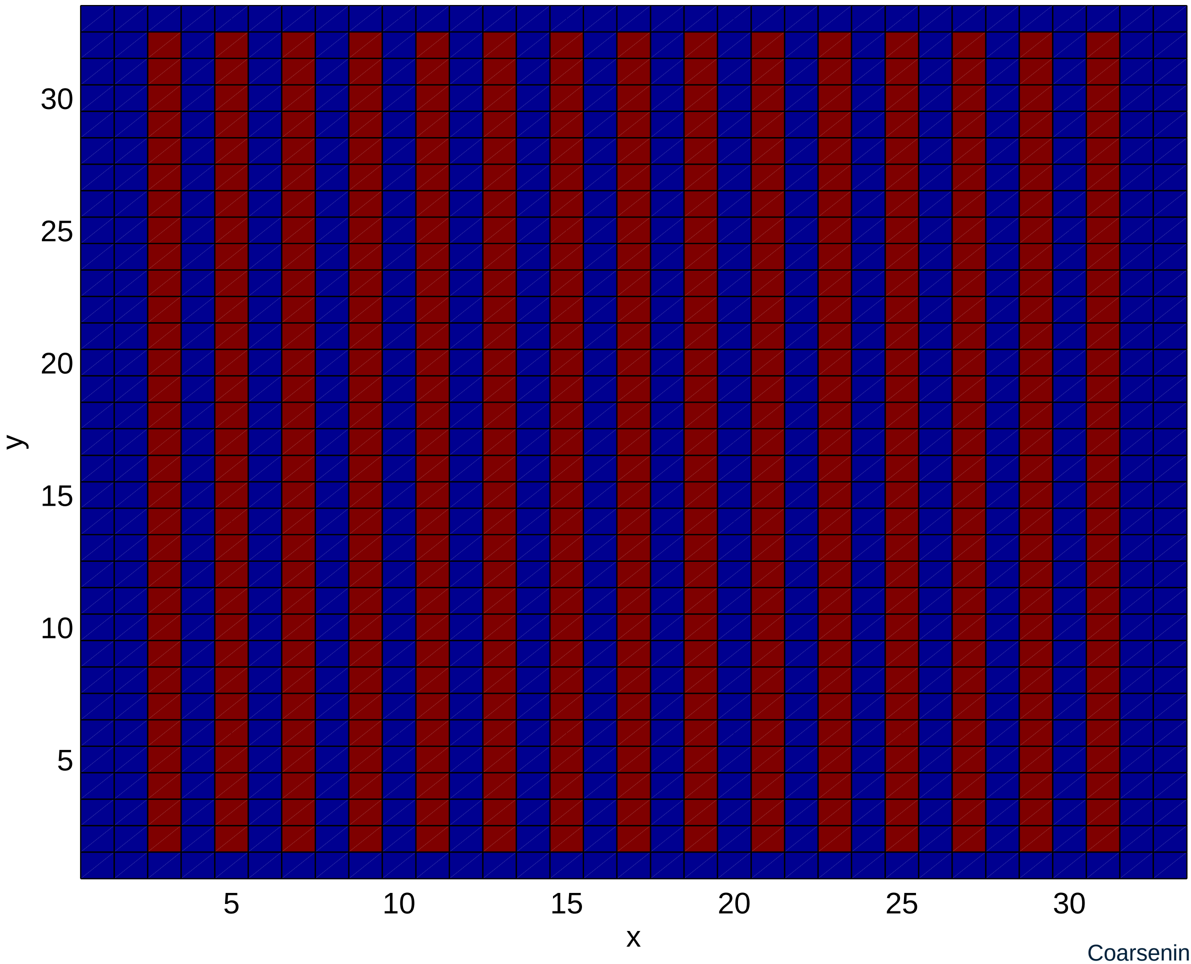
# Choices of coarse grids

- $-u_{xx} - u_{yy} = f$, Dirichlet BCs
- $32 \times 32$ bilinear finite element grid
- 2 Steps Weighted Jacobi to determine $S_{ij}$



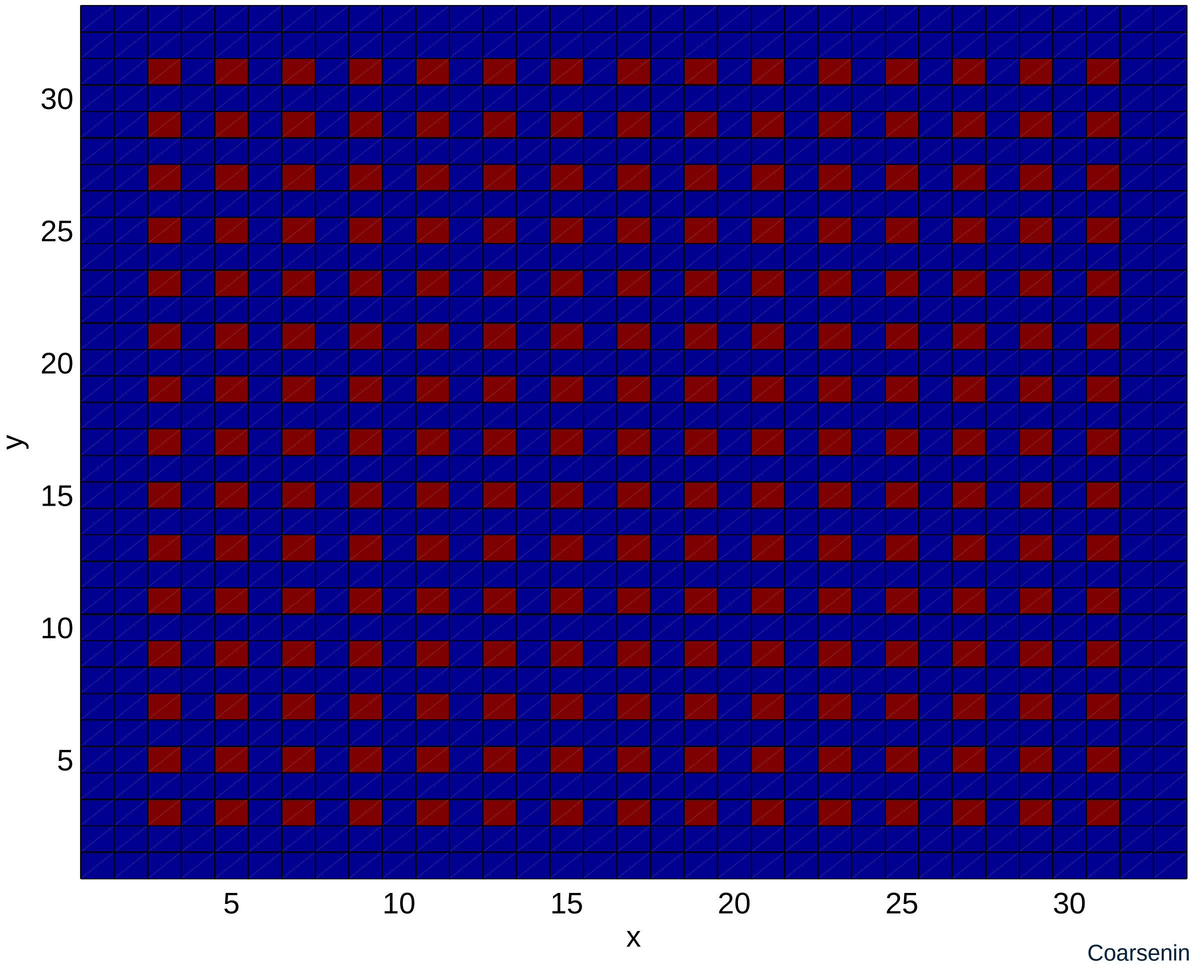
# Choices of coarse grids

- $-u_{xx} - 0.01u_{yy} = f$, Dirichlet BCs
- $32 \times 32$ bilinear finite element grid
- 2 Steps Weighted Jacobi to determine $S_i$



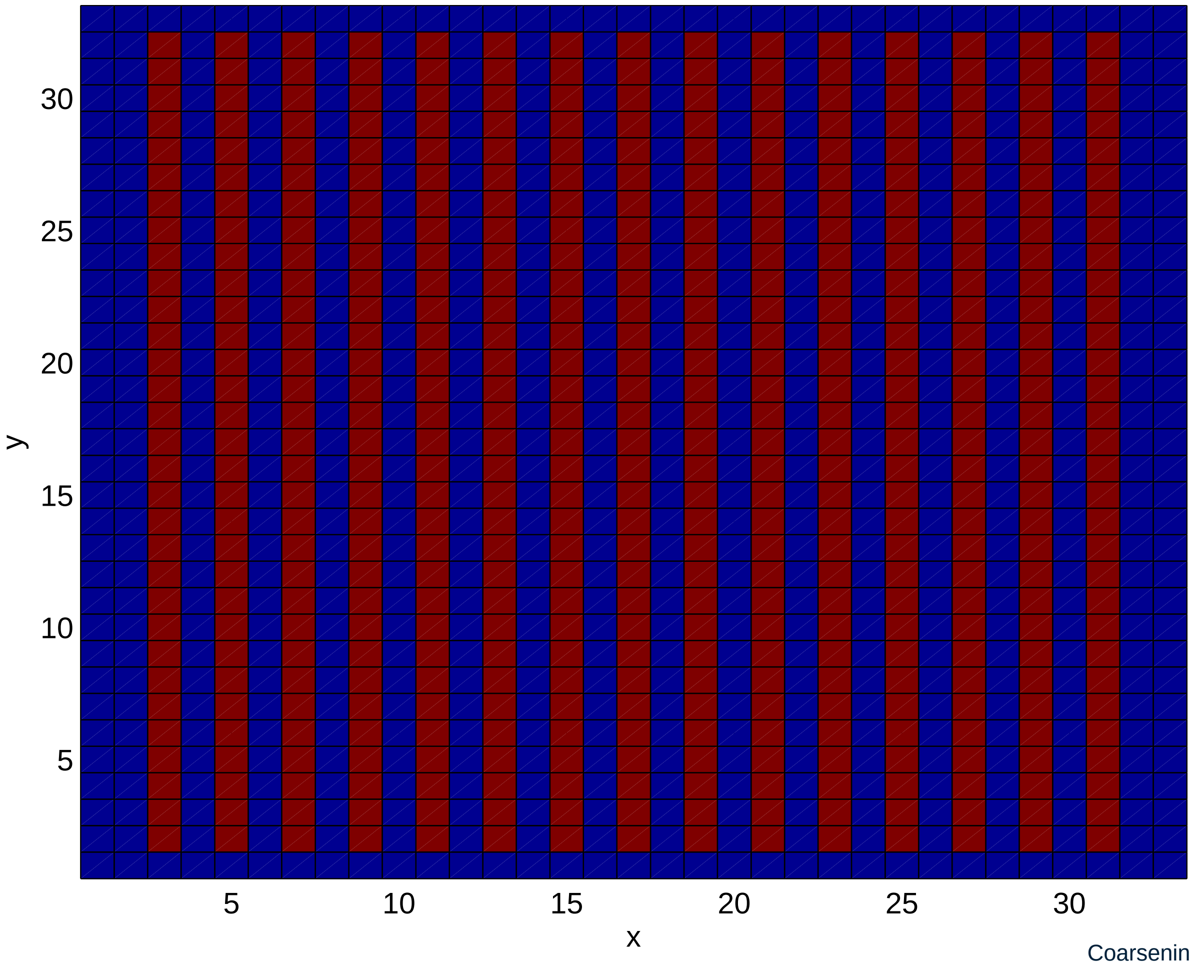
# Choices of coarse grids

- $-u_{xx} - u_{yy} = f$, Dirichlet BCs
- $32 \times 32$ bilinear finite element grid, $A \rightarrow DAD$, $d_{ii} = 10^{5r_i}$
- 2 Steps Weighted Jacobi to determine $S_i$



# Choices of coarse grids

- $-u_{xx} - 0.01u_{yy} = f$, Dirichlet BCs
- $32 \times 32$ bilinear finite element grid, $A \rightarrow DAD$, $d_{ii} = 10^{5r_i}$
- 2 Steps Weighted Jacobi to determine $S_i$



# Algorithm

- Given $A$,$b$
- Relax $\nu_0$ times on $A\mathbf{x} = \mathbf{0}$ with a random initial guess
- On each level
  - Determine local strong connections by $\mu$ relaxations on $A\mathbf{G}^{(i)} = \mathbf{I}^{(i)}$
  - Choose coarse grid by colouring algorithm
  - Relax $\nu_1$ times on $A\mathbf{x} = \mathbf{0}$ to improve representation of algebraically smooth error
  - Form interpolation, $P$, based on $\mathbf{x}$
  - Compute $A_c = P^T AP$, inject $\mathbf{x}_c = (\mathbf{x})_c$
- Examples have fixed $\nu_0 = \nu_1 = 15$, $\mu = 2$

# Numerical Results

Convergence Factors of Resulting V(1,1) Cycles

| grid | Laplace | Scaled Laplace | Anisotropic | Scaled Anisotropic |
|---|---|---|---|---|
| $32 \times 32$ | 0.06 | 0.06 | 0.10 | 0.10 |
| $64 \times 64$ | 0.07 | 0.07 | 0.10 | 0.10 |
| $128 \times 128$ | 0.07 | 0.07 | 0.10 | 0.10 |
| $256 \times 256$ | 0.07 | 0.07 | 0.10 | 0.10 |
| $512 \times 512$ | 0.07 | 0.07 | 0.10 | 0.10 |

Good convergence factors, but setup cost is now high

# Compatible Relaxation

- Heuristics seem to work well and can be made robust
- Alternative: choose coarse grids so that we know that interpolation can be chosen to complement relaxation

Theory of Compatible Relaxation says that if relaxation on the fine-grid submatrix is fast to converge, then there is an interpolation operator which yields a multigrid method with small convergence factor

# Compatible Relaxation

- Heuristics seem to work well and can be made robust
- Alternative: choose coarse grids so that we know that interpolation can be chosen to complement relaxation

Theory of Compatible Relaxation says that if relaxation on the fine-grid submatrix is fast to converge, then there is an interpolation operator which yields a multigrid method with small convergence factor

- Fine-grid relaxation: $I - \omega B_{ff}^{-1} A_{ff}$ is efficient if

$$c_1 \langle B_{ff} x_f, x_f \rangle \leq \langle A_{ff} x_f, x_f \rangle \leq c_2 \langle B_{ff} x_f, x_f \rangle$$

  for reasonable $c_1$,$c_2$

- Under right assumptions, can show multigrid convergence is bounded less than 1, with bound dependent on $\frac{c_2}{c_1}$

# Satisfying the Theory

- Want to turn these results into a practical algorithm
  - Idea: Choose partition so that we know fine-grid relaxation converges quickly
  - Weighted Jacobi relaxation on $A_{ff}$ converges quickly when $A_{ff}$ is diagonally dominant

We can guarantee good 2-level convergence factors by choosing $A_{ff}$ to be diagonally-dominant

# NP-completeness

- Define $\theta$-dominance of $A_{ff}$ as

$$a_{ii} \geq \theta \sum_{j \in F} |a_{ij}|$$

- Know theory can be satisfied as long as $A_{ff}$ is $\theta$-dominant
- So, choose $A_{ff}$ to be the largest submatrix of $A$ that is $\theta$-dominant

# NP-completeness

- Define $\theta$-dominance of $A_{ff}$ as

$$a_{ii} \geq \theta \sum_{j \in F} |a_{ij}|$$

- Know theory can be satisfied as long as $A_{ff}$ is $\theta$-dominant
- So, choose $A_{ff}$ to be the largest submatrix of $A$ that is $\theta$-dominant

This is an NP-complete problem

# Greedy Algorithm Approach

- Want an $O(n)$ coarse-grid selection algorithm, not an NP-complete one
  - Initialize all points to be in $U$; $F$, $C$ to be empty
  - For each point $i$, compute diagonal dominance measure

    $$\hat{\theta}_i = \frac{|a_{ii}|}{\sum\limits_{j \in F \cup U} |a_{ij}|}$$

    If $\hat{\theta}_i \geq \theta$, put $i$ into $F$, remove it from $U$
  - While $U$ is non-empty
    - Find $j = \underset{i \in U}{\operatorname{argmin}} \, \hat{\theta}_i$
    - Remove $j$ from $U$, put it in $C$
    - For each neighbouring point $i$ of $j$, update $\hat{\theta}_i$

      If $\hat{\theta}_i \geq \theta$, put $i$ into $F$, remove it from $U$

# Test Problems

- Test problems based on finite element discretizations of $-\nabla \cdot K(x,y)\nabla p(x,y)$
    - Laplace equation, $K(x,y) = 1$
    - Smooth coefficient, $K(x,y) = 10^{-8} + 10(x^2 + y^2)$
    - Randomly chosen coefficient, $K(x,y) = 10^{-8}$ on 20% of the cells, chosen randomly, $K(x,y) = 1$ otherwise
    - Anisotropic coefficient, $K(x,y) = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$
- Algorithm:
    - Greedy algorithm to select coarse grids
    - Classical AMG to define interpolation
    - Usual AMG-V(1,1) cycles with Gauss-Seidel relaxation

# Multilevel AMG results

| Coefficient | Grid | $c_A$ | $t_{\text{setup}}$ | $t_{\text{solve}}$ | # iters. | $\rho$ |
|---|---|---|---|---|---|---|
| $K(x, y) = 1$ | $512 \times 512$ | 1.33 | 1.3 | 0.7 | 5 | 0.13 |
| | $1024 \times 1024$ | 1.33 | 5.1 | 2.5 | 5 | 0.14 |
| | $2048 \times 2048$ | 1.33 | 21.9 | 10.5 | 5 | 0.14 |
| smooth $K(x, y)$ | $512 \times 512$ | 1.33 | 1.3 | 0.6 | 5 | 0.13 |
| | $1024 \times 1024$ | 1.33 | 5.1 | 2.5 | 5 | 0.14 |
| | $2048 \times 2048$ | 1.33 | 21.7 | 10.4 | 5 | 0.14 |
| random $K(x, y)$ | $512 \times 512$ | 2.06 | 2.3 | 1.2 | 6 | 0.35 |
| | $1024 \times 1024$ | 2.08 | 9.6 | 4.8 | 6 | 0.40 |
| | $2048 \times 2048$ | 2.10 | 41.0 | 19.8 | 6 | 0.46 |
| anisotropic $K(x, y)$ | $512 \times 512$ | 2.39 | 1.5 | 1.0 | 5 | 0.13 |
| | $1024 \times 1024$ | 2.41 | 6.2 | 4.1 | 5 | 0.20 |
| | $2048 \times 2048$ | 2.43 | 25.8 | 17.7 | 5 | 0.20 |

# Summary

- Single-level iterative methods slow to resolve low-energy modes of $B^{-1}A$
- Structure of these modes allows definition of efficient multigrid solvers
- If slow-to-converge error cannot be characterized beforehand, adaptive multigrid techniques can recover good performance at cost of extra setup
- Better heuristics allow more robust coarse-grid selection, but cost questions are important
- Theoretically motivated coarsening gives encouraging results
  - Competitive algorithmic cost
  - Good experimental results
  - Robustness and parallelism questions

# Support and Collaboration

- Initial work was 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 in collaboration with Steve McCormick, Tom Manteuffel, John Ruge, Marian Brezina at CU-Boulder, and Rob Falgout from CASC-LLNL.
    - Energy-based coarsening in collaboration with Steve McCormick, Tom Manteuffel, John Ruge, Marian Brezina, and James Brannick from CU-Boulder
- Theoretically driven coarsening in collaboration with Yousef Saad at UMN, supported by NSF-ACIR