Improving and Understanding Algebraic Multigrid Convergence

Scott MacLachlan
scott.maclachlan@gmail.com

Delft Institute of Applied Mathematics, TU-Delft
Centrum voor Wiskunde en Informatica, Amsterdam
Department of Mathematics, Tufts University

September 6, 2007

Support and Collaboration

- Research supported by the DOE SciDAC TOPS program, the Center for Applied Scientific Computing at Lawrence Livermore National Lab, and by the European Community's Sixth Framework Programme, through a Marie Curie International Incoming Fellowship, MIF1-CT-2006-021927

- Primarily in collaboration with
  - Steve McCormick, Tom Manteuffel, John Ruge, Marian Brezina at CU-Boulder, and Rob Falgout from CASC-LLNL.
  - James Brannick from Penn State University

Big Picture

AMG is a nice algorithm

- Efficiently solves many problems
- Good algorithmic and parallel scalability
- Somewhat mature technology

Big Picture

AMG is a nice algorithm

- Efficiently solves many problems
- Good algorithmic and parallel scalability
- Somewhat mature technology

AMG isn't perfect

- Sensitive to parameter choices
- Requires some expert knowledge
- Convergence isn't well understood

Big Picture

AMG is a nice algorithm

- Efficiently solves many problems
- Good algorithmic and parallel scalability
- Somewhat mature technology

AMG isn't perfect

- Sensitive to parameter choices
- Requires some expert knowledge
- Convergence isn't well understood

When AMG works, it is often the best solver

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 difference/element models of elliptic systems
- Matrices are
  - sparse
  - symmetric
  - positive definite

Solving $Ax = b$

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

Algorithmic Goals

Efficient and robust solver for heterogeneous models

Efficient: Optimal scalability, both algorithmic and parallel

- Cost of solve linearly proportional to number of unknowns
- Natural parallelism; most calculations should be data-local

Robust: Consistent performance with few parameters

- Predictable performance based on simple characteristics
- Not expert software; no magic parameters

Efficiency First

Heterogeneity is an added complication, but not fundamental

Still need techniques to handle

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

Robustness without efficiency is EASY!

Gaussian Elimination

Efficiency First

Heterogeneity is an added complication, but not fundamental

Still need techniques to handle

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

Robustness without efficiency is EASY!
Gaussian Elimination

- Start with an efficient solver for homogeneous models
  - Geometric multigrid
- Look for where robustness and heterogeneity play 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 $M^{-1} \approx A^{-1}$
- Take $x^{(1)} = x^{(0)} + M^{-1}r^{(0)}$

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

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

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

$$e^{(2)} = (I - M^{-1}A)e^{(1)}$$

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

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

$$e^{(2)} = (I - M^{-1}A)^2 e^{(0)}$$

Stationary Iterative Methods

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

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

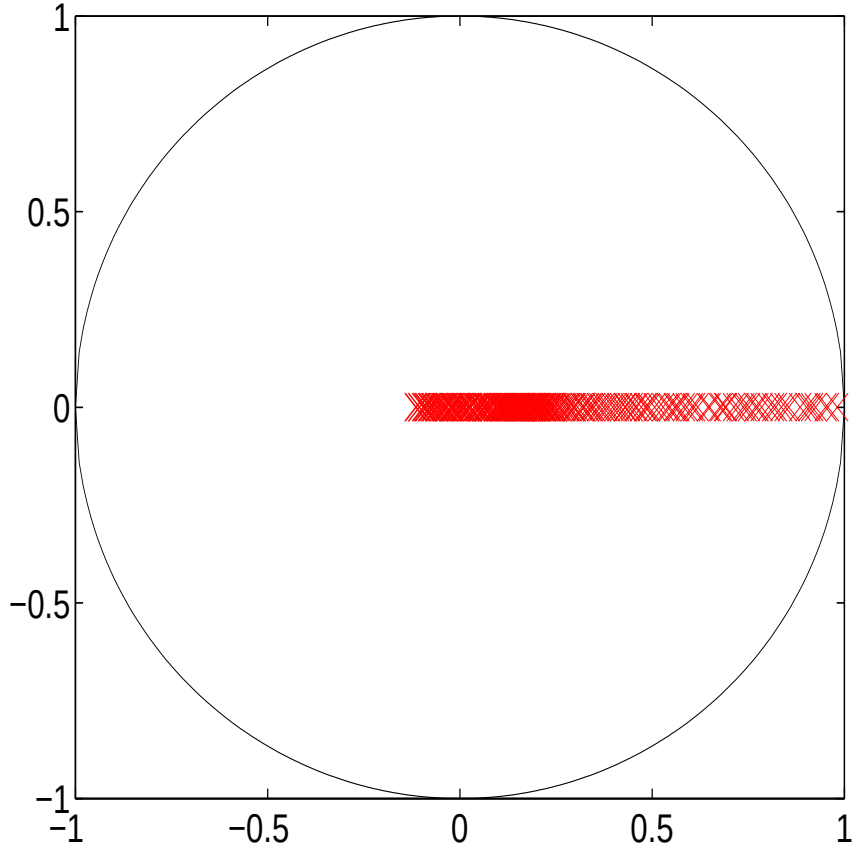
- Choose $M^{-1} \approx A^{-1}$
- Take $x^{(1)} = x^{(0)} + M^{-1}r^{(0)}$

Error propagation form:

$$\begin{aligned} e^{(1)} &= (I - M^{-1}A)e^{(0)} \\ e^{(2)} &= (I - M^{-1}A)^2 e^{(0)} \\ &\vdots \\ e^{(n)} &= (I - M^{-1}A)^n e^{(0)} \end{aligned}$$

Convergence of Stationary Iterations

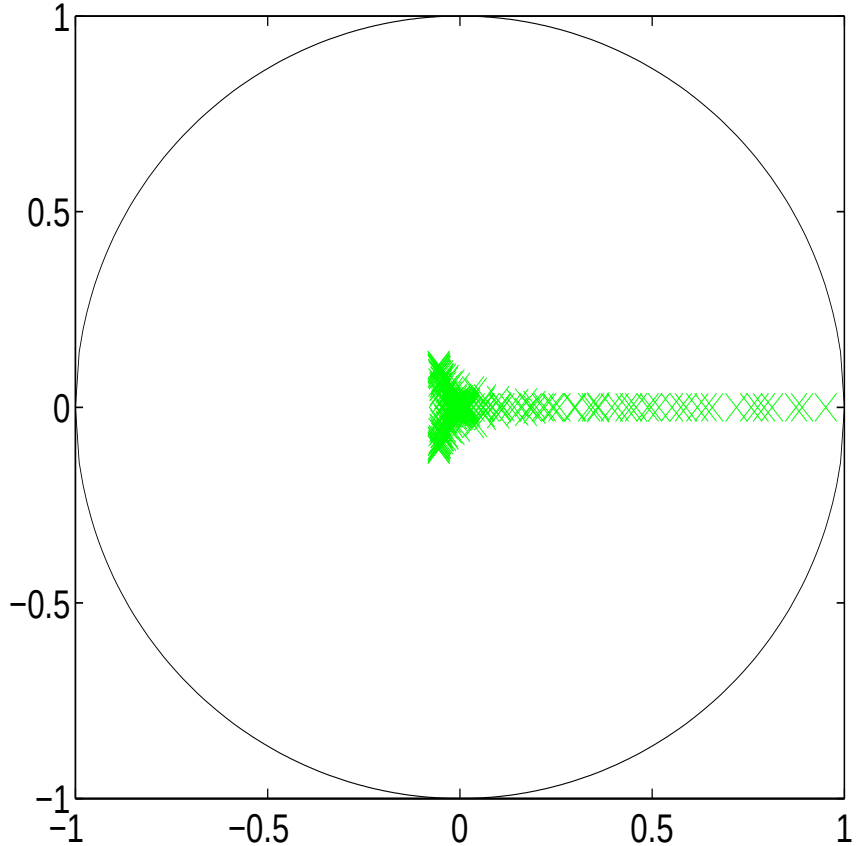
Convergence depends on spectrum of $I - M^{-1}A$



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

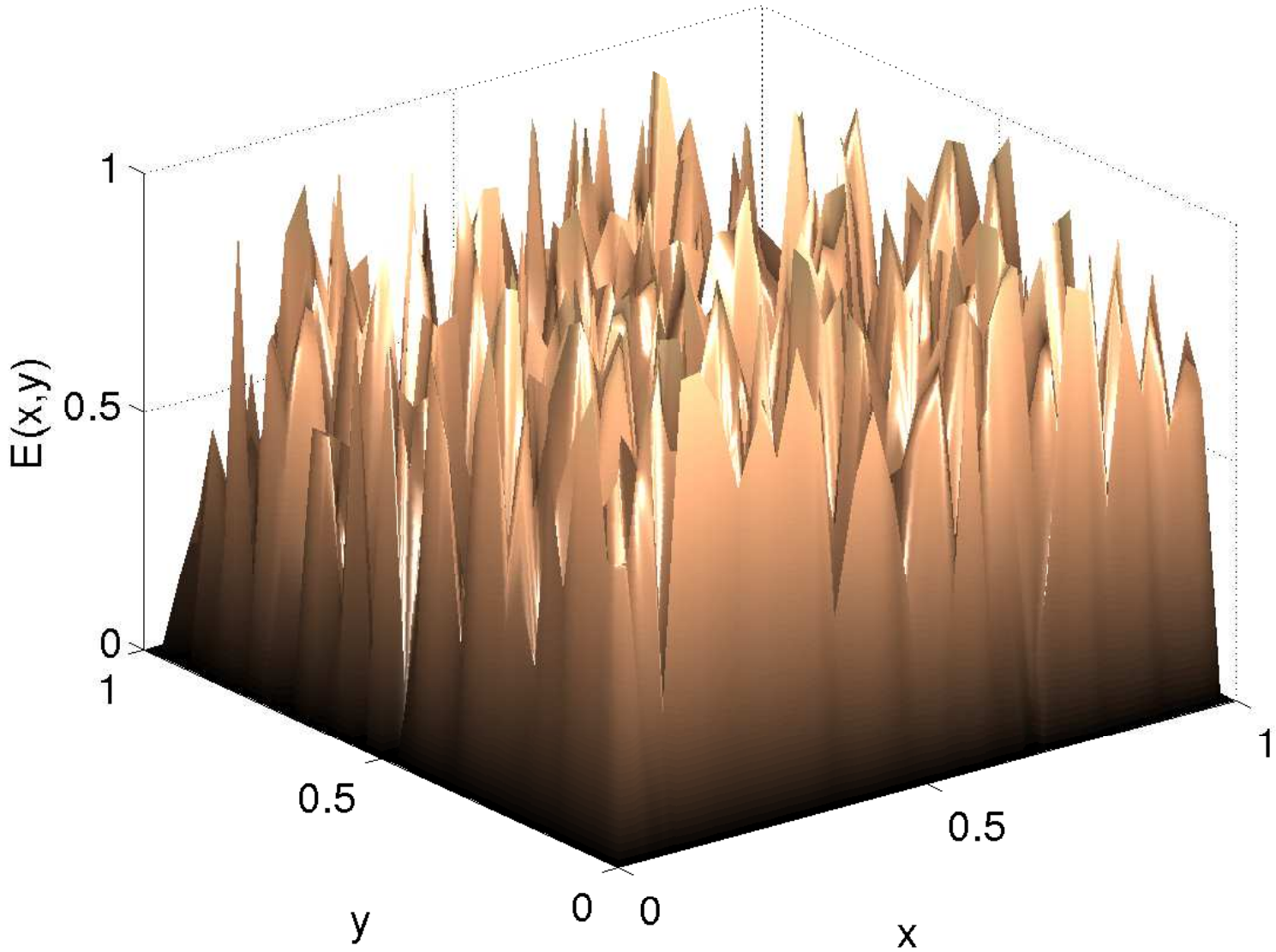
Convergence of Stationary Iterations

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



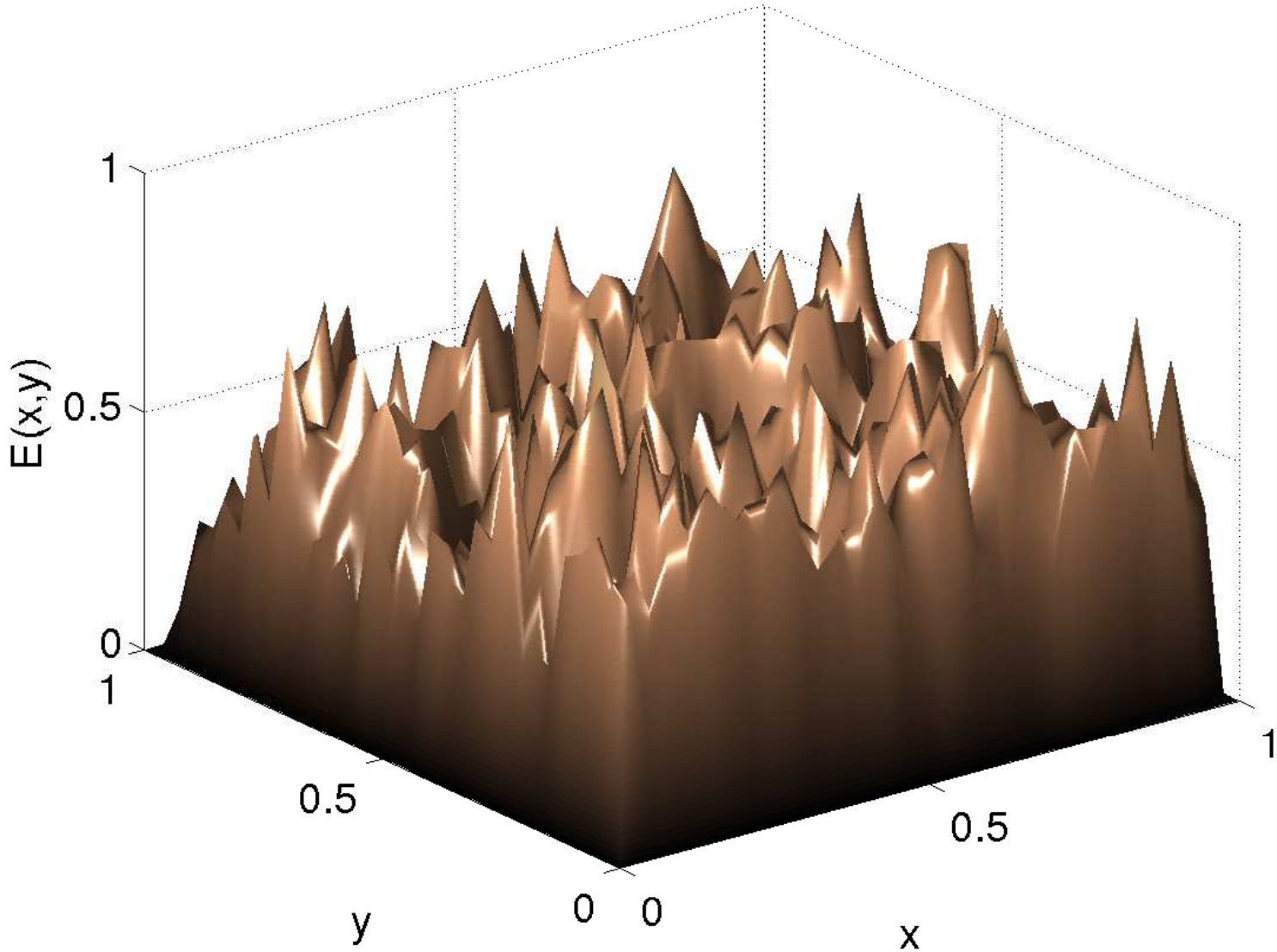
Gauss-Seidel Iteration: $e^{(n)} = (I - L^{-1}A)^n e^{(0)}$

Smoothing Property



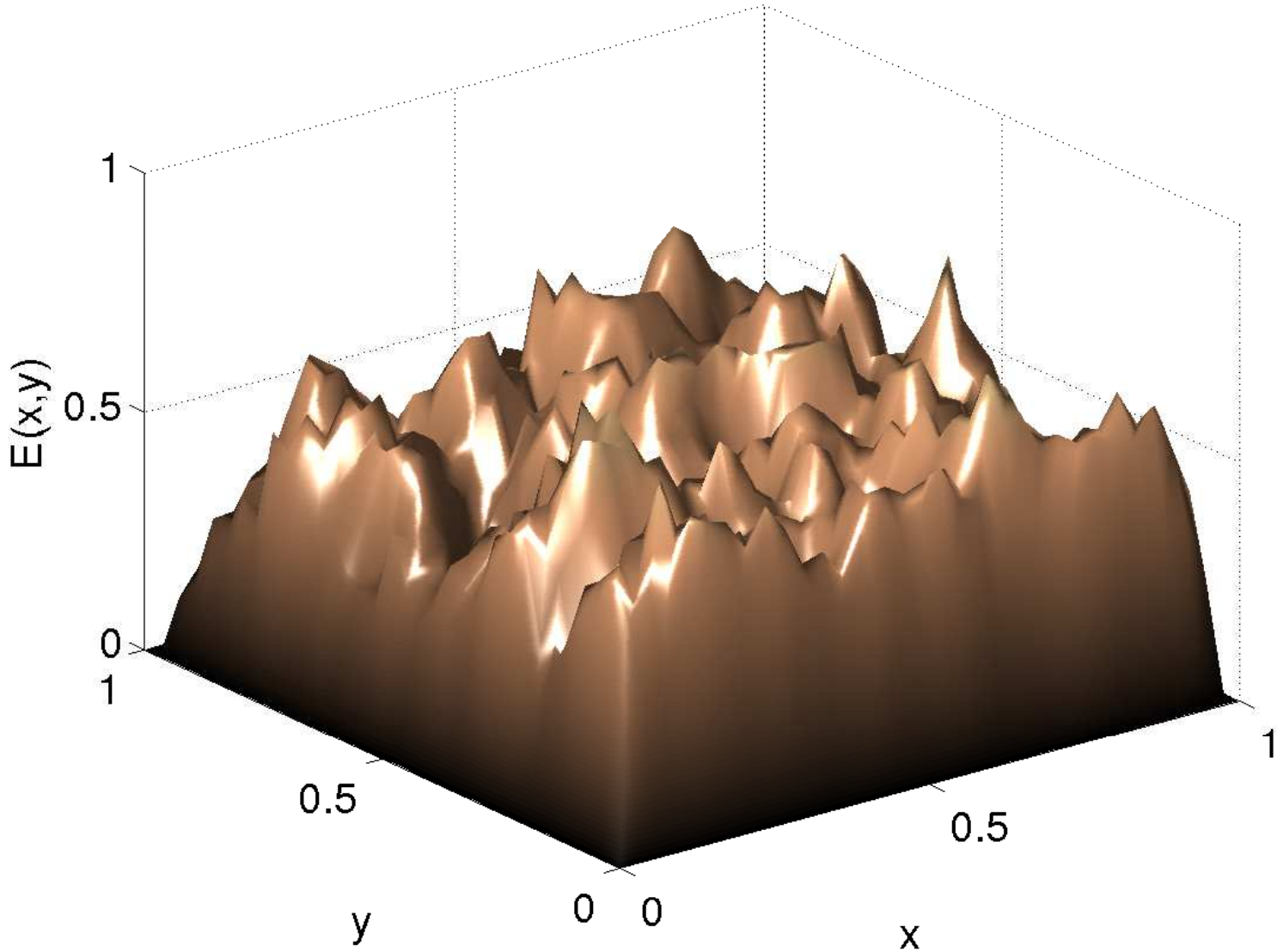
Random initial error

Smoothing Property



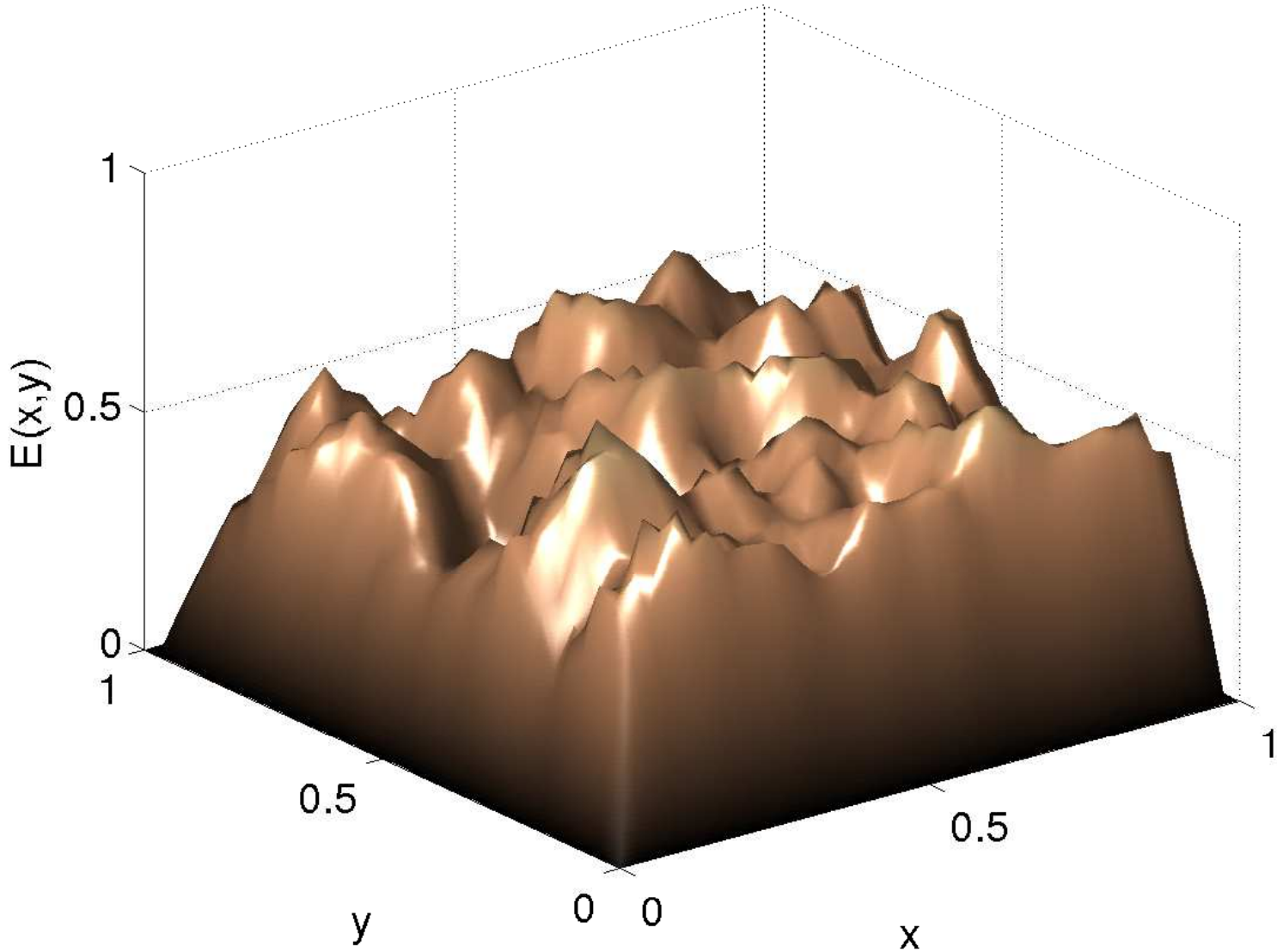
Error after 1 weighted Jacobi iteration

Smoothing Property



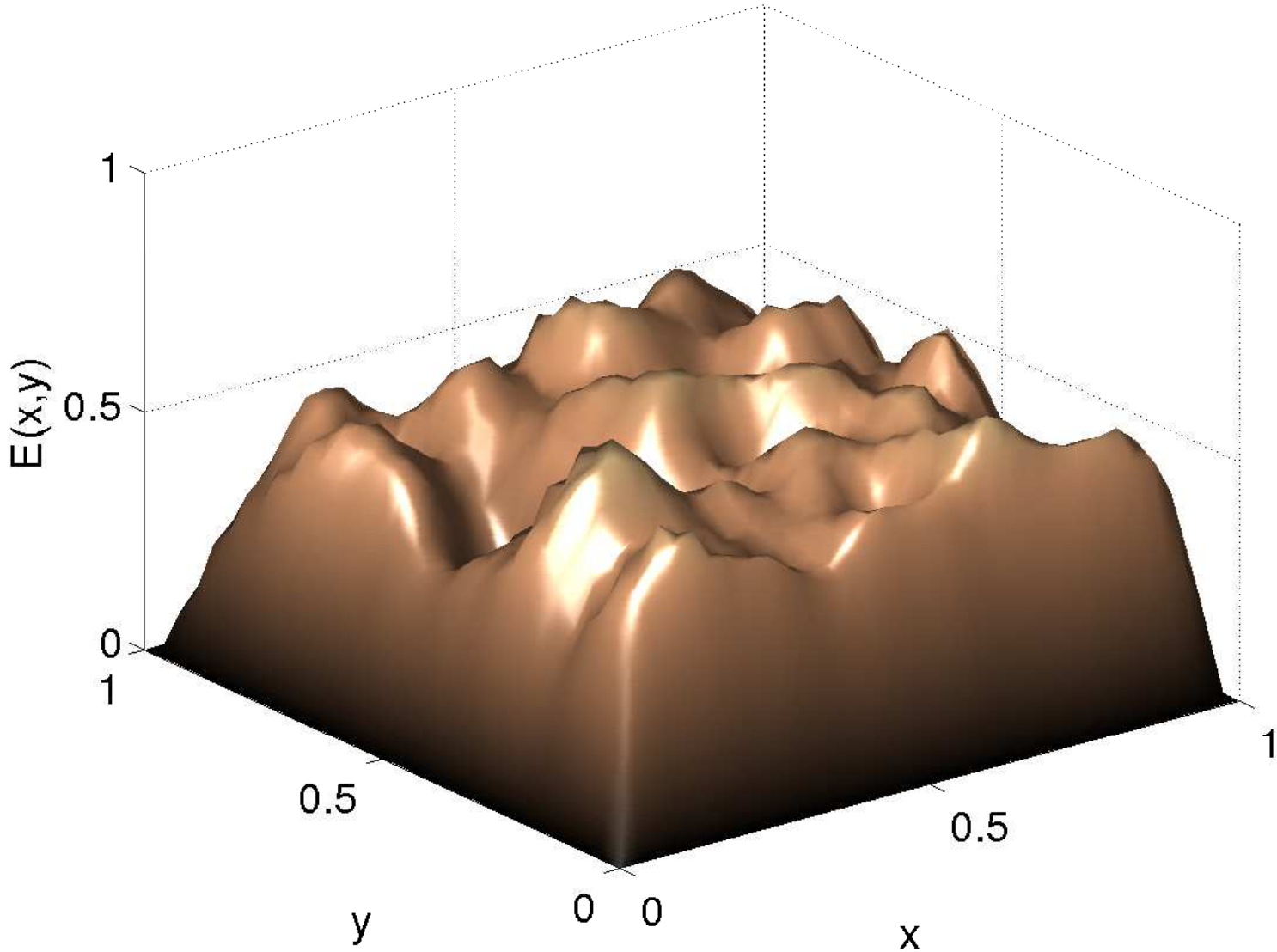
Error after 2 weighted Jacobi iterations

Smoothing Property



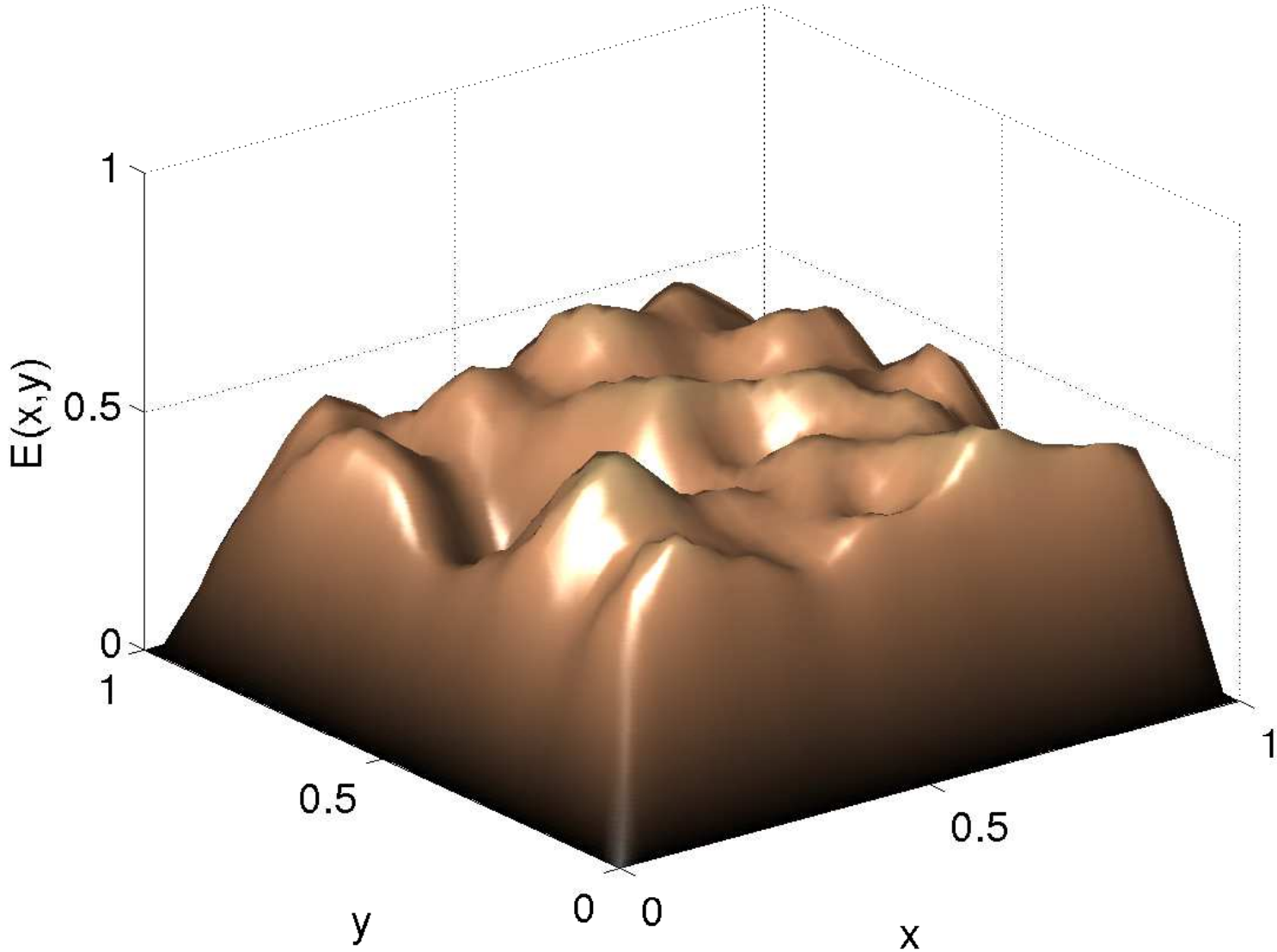
Error after 3 weighted Jacobi iterations

Smoothing Property



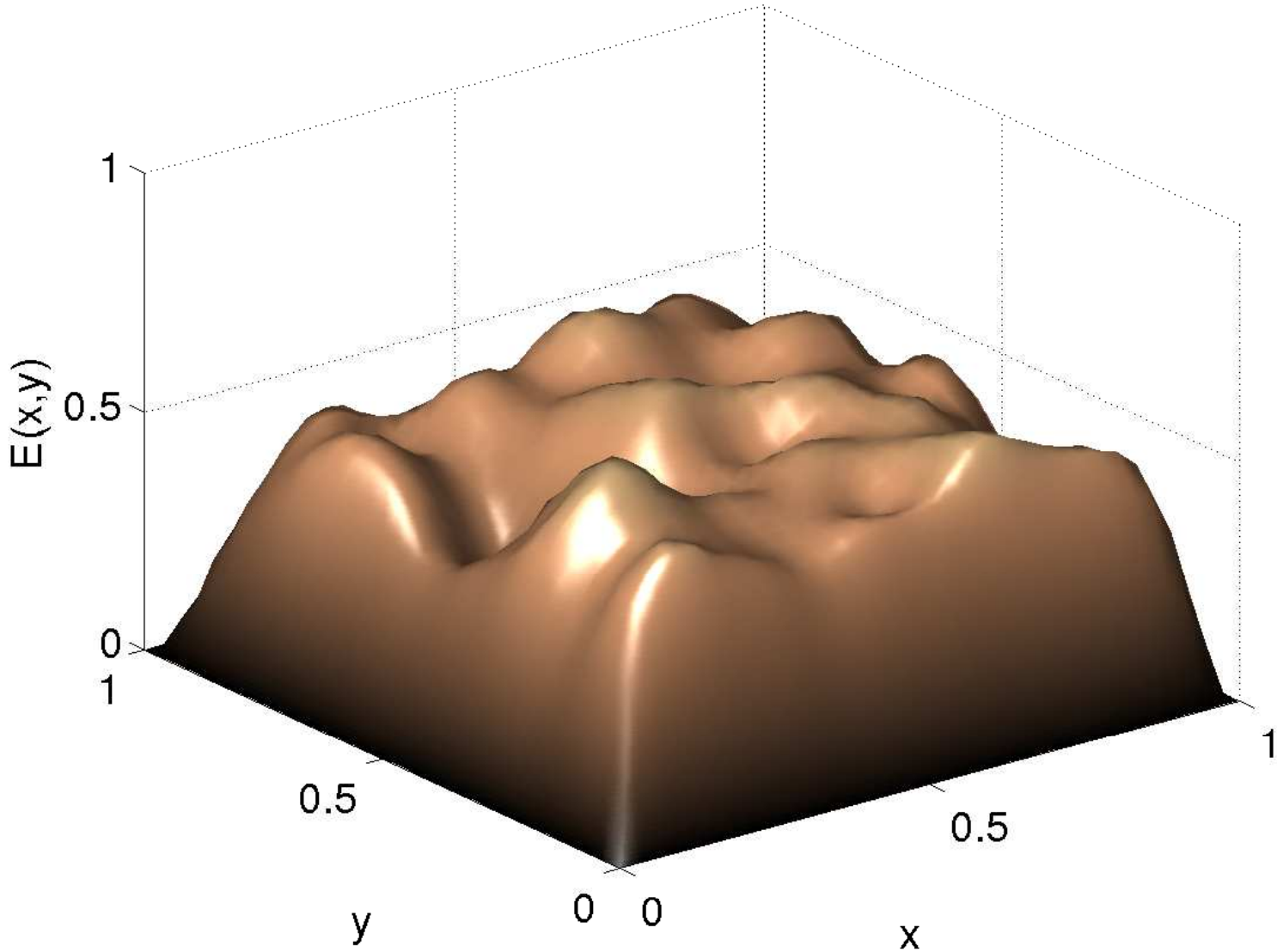
Error after 4 weighted Jacobi iterations

Smoothing Property



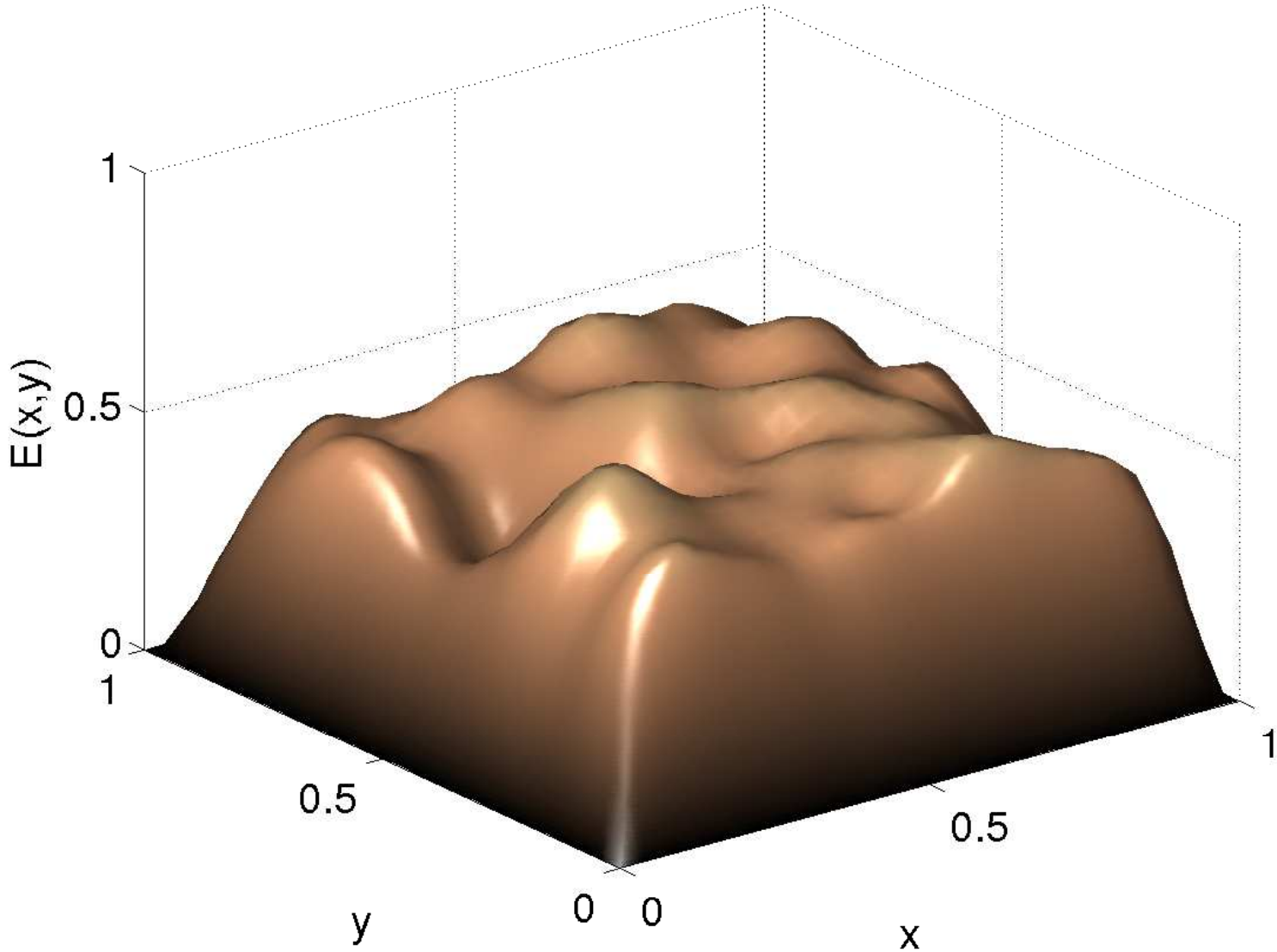
Error after 5 weighted Jacobi iterations

Smoothing Property



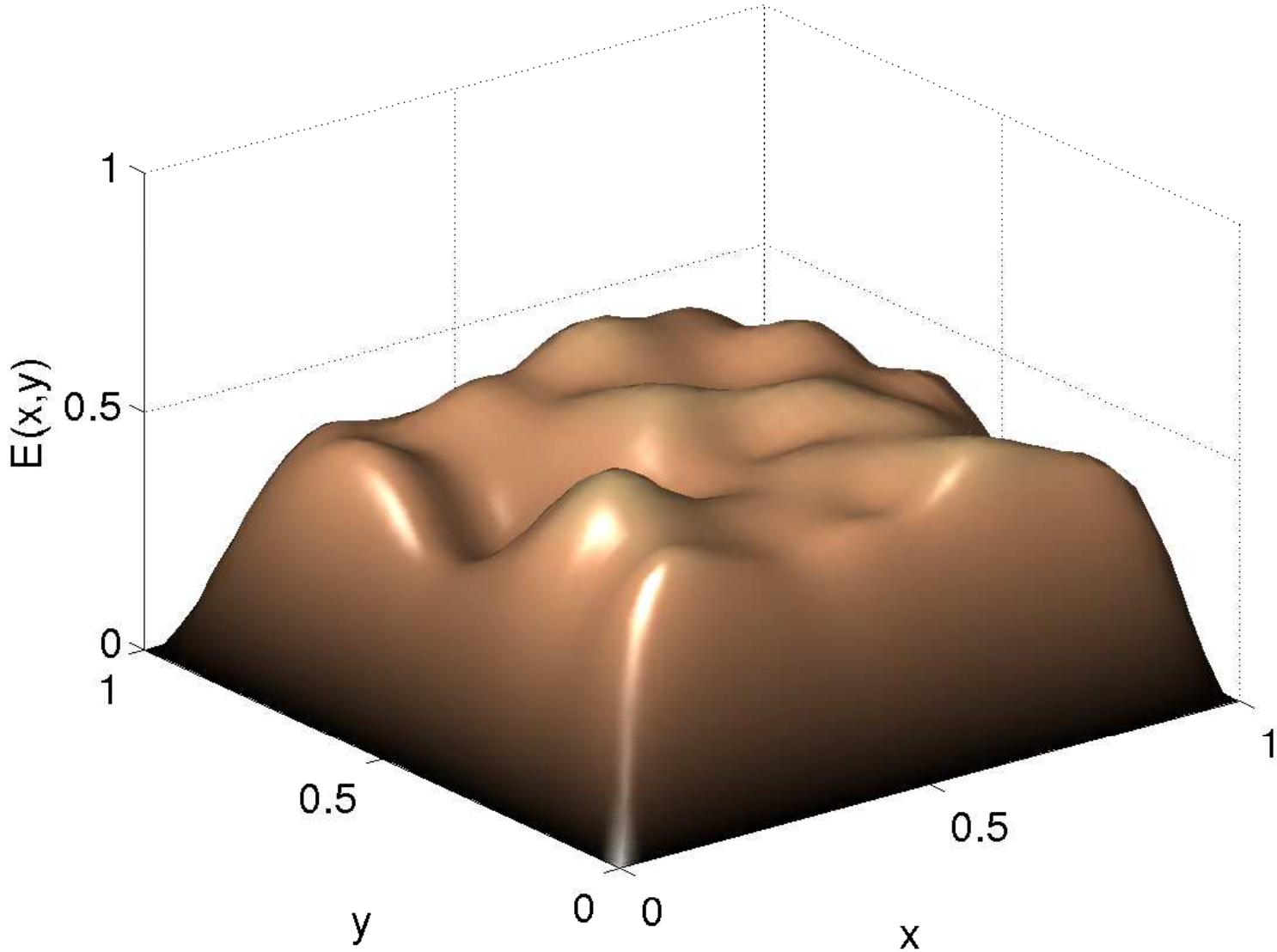
Error after 6 weighted Jacobi iterations

Smoothing Property



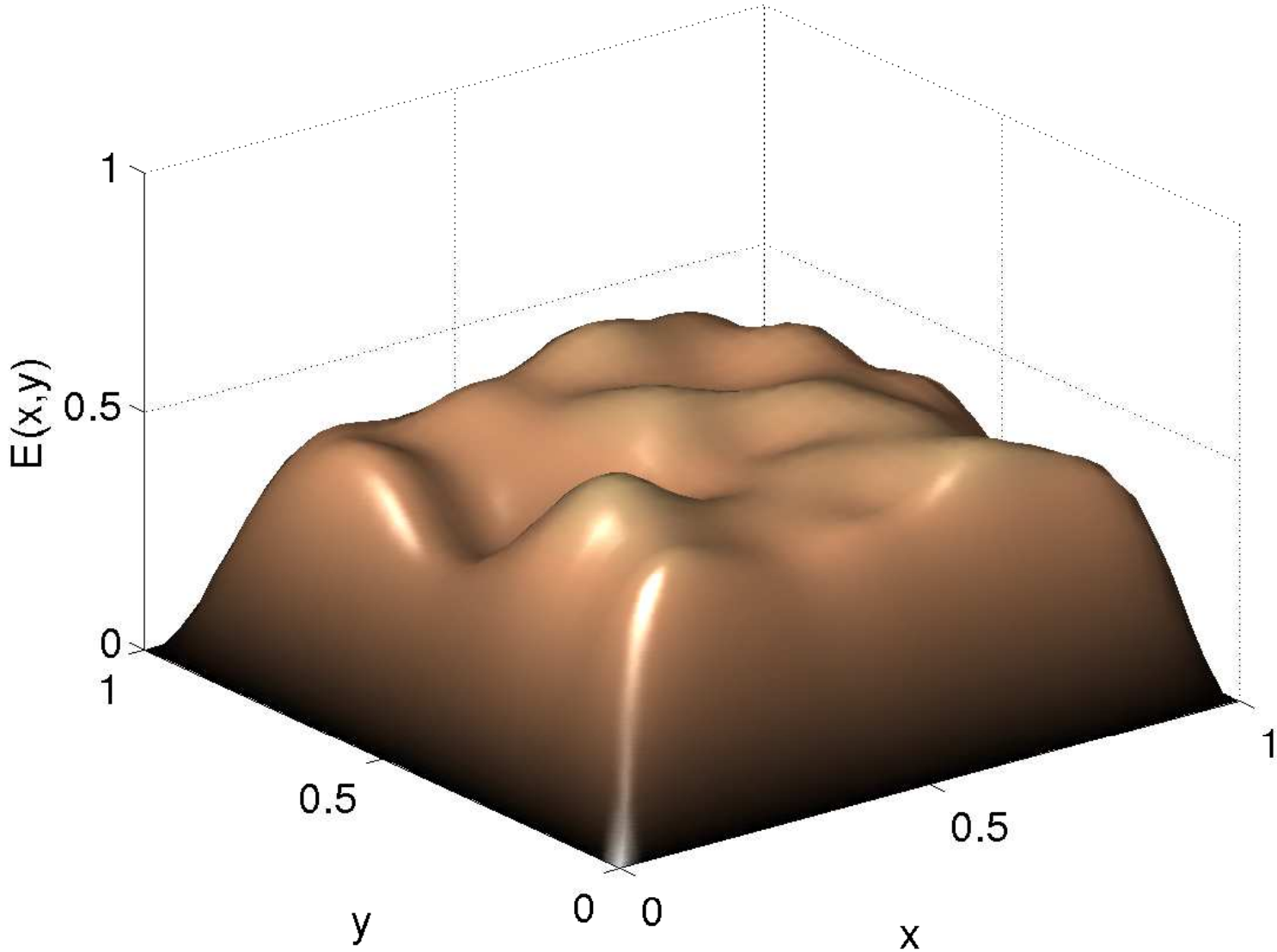
Error after 7 weighted Jacobi iterations

Smoothing Property



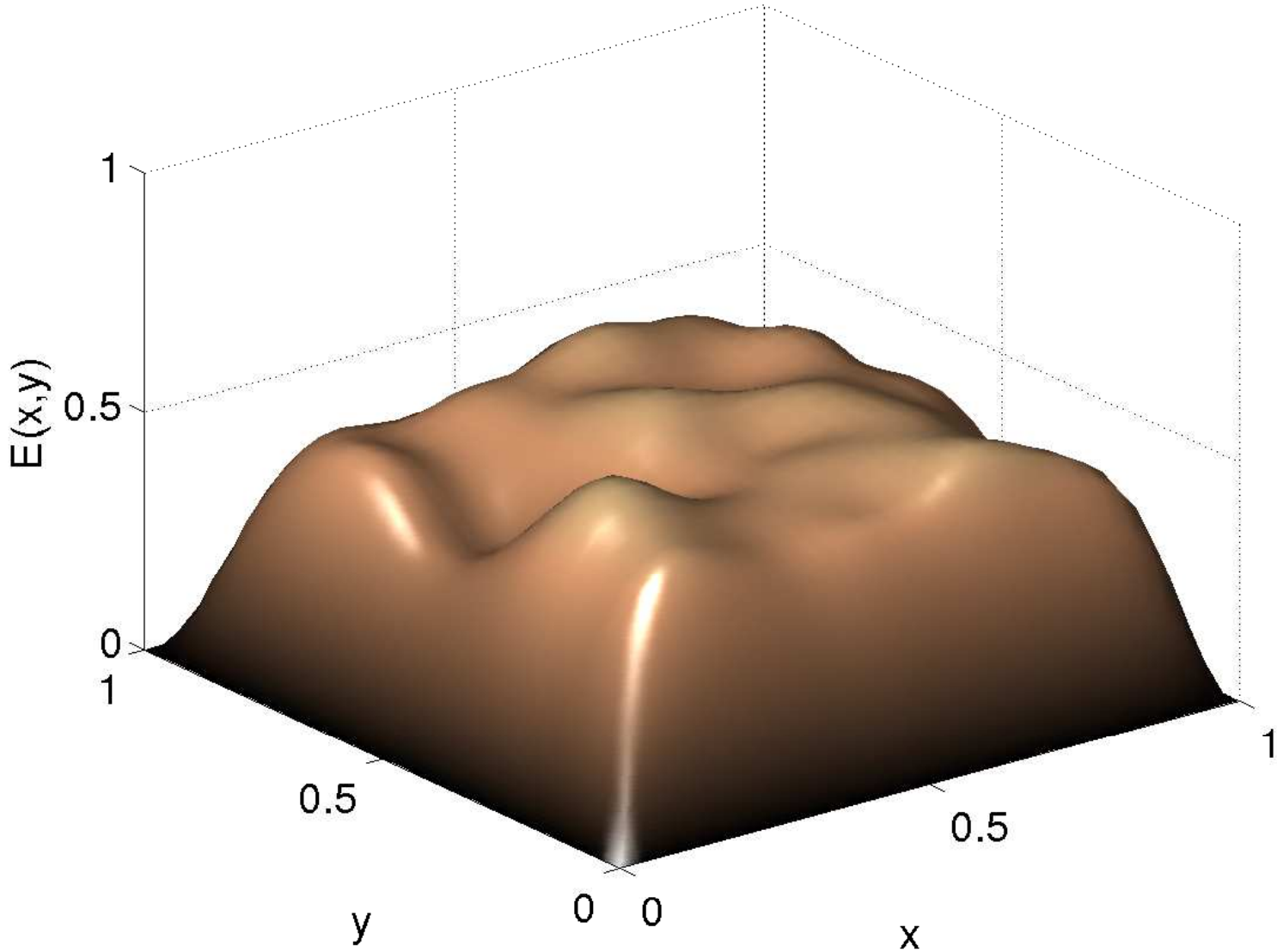
Error after 8 weighted Jacobi iterations

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 Grids

- Sine series representation:

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

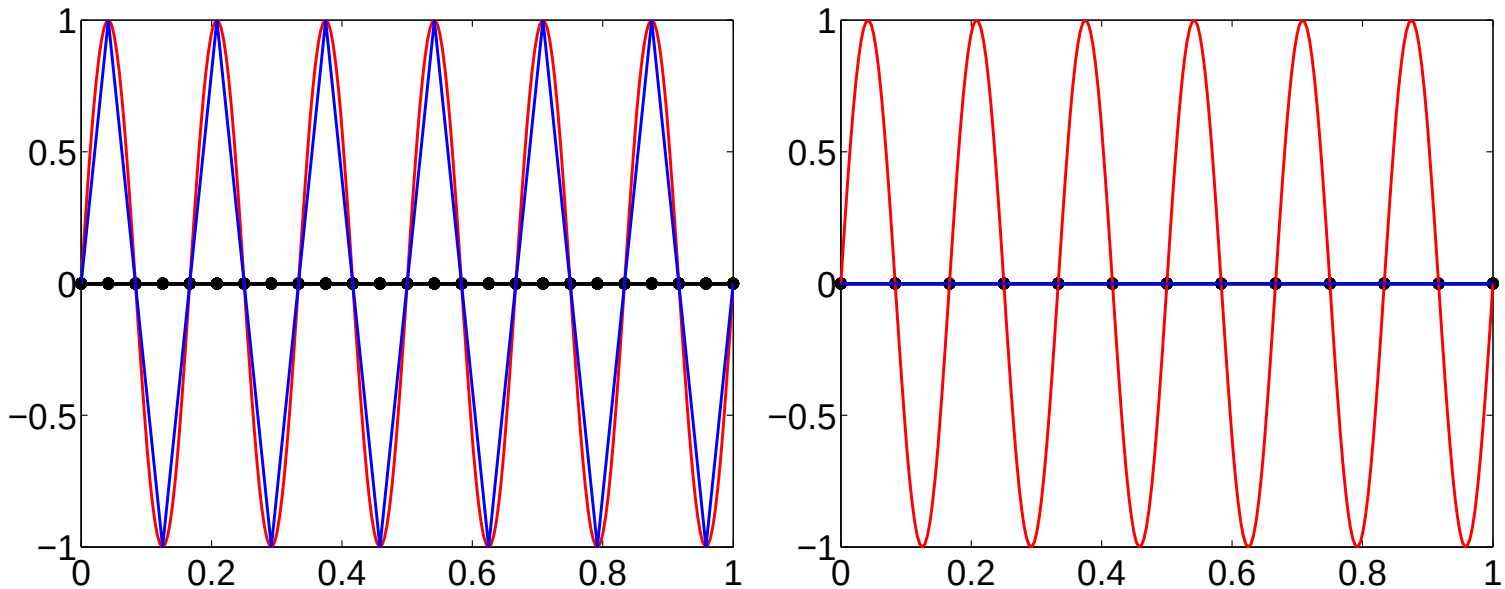
- Discrete problems can only approximate certain modes

Coarse Grids

- Sine series representation:

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

- Discrete problems can only approximate certain modes



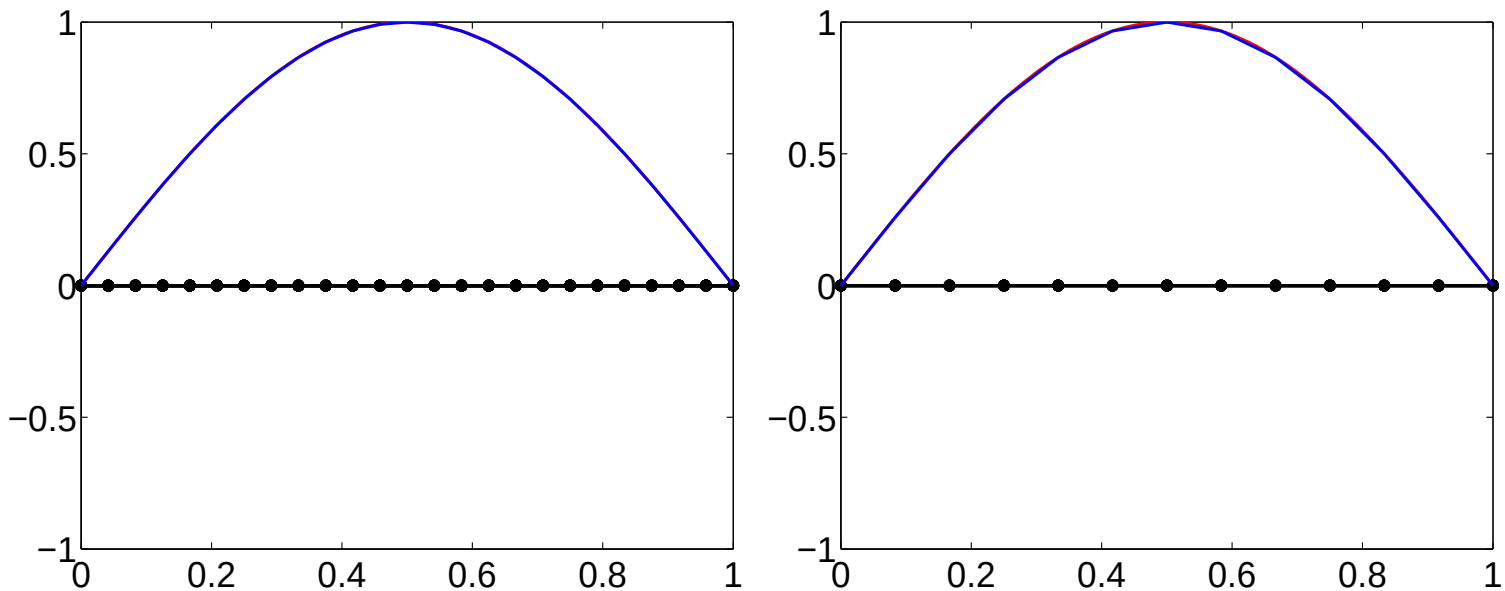
$\sin(12\pi x)$

Coarse Grids

- Sine series representation:

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

- Discrete problems can only approximate certain modes



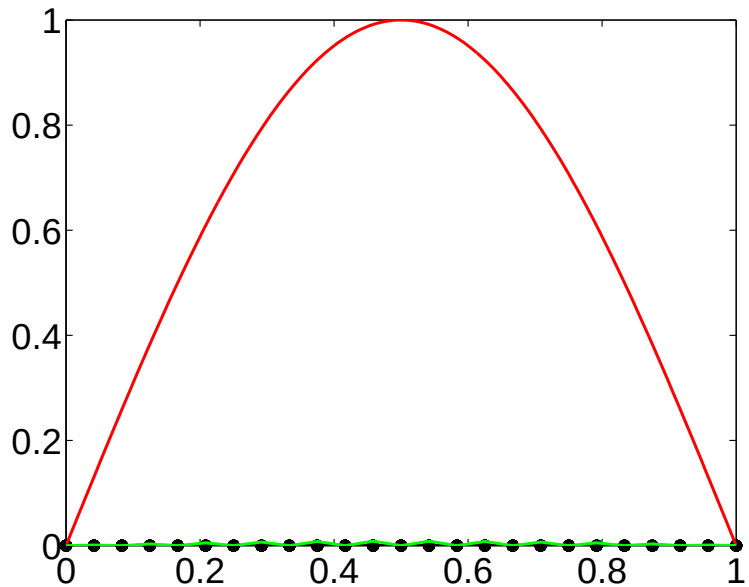
$\sin(\pi x)$

Coarse Grids

- Sine series representation:

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

- Discrete problems can only approximate certain modes



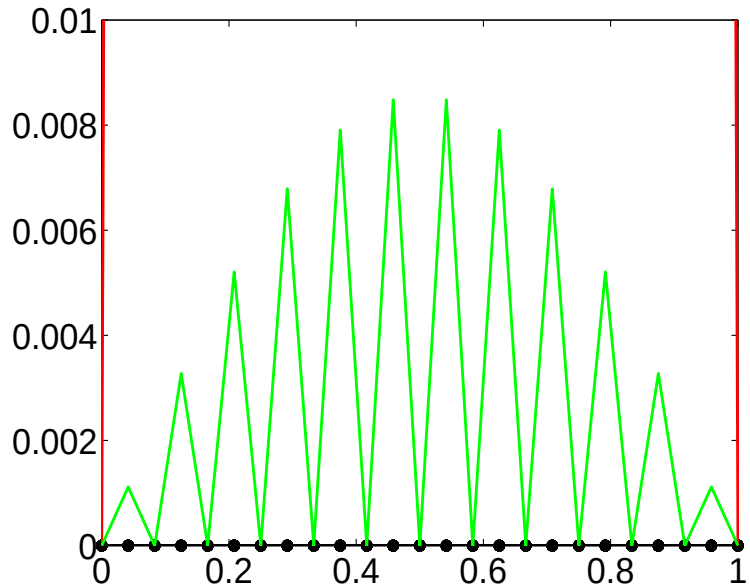
Error in coarse-grid representation of $\sin(\pi x)$

Coarse Grids

- Sine series representation:

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

- Discrete problems can only approximate certain modes



Error in coarse-grid representation of $\sin(\pi x)$

Coarse Grids

- Sine series representation:

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

- Discrete problems can only approximate certain modes

Coarse grids accurately represent low-frequency modes

Natural complement to relaxation

Complementarity

Use two complementary processes to efficiently damp all errors

Relaxation: Damp high-frequency error by stationary iteration

Coarse-grid correction: Eliminate low-frequency error by relaxation on coarse grids

Complementarity

Use two complementary processes to efficiently damp all errors

Relaxation: Damp high-frequency error by stationary iteration

Coarse-grid correction: Eliminate low-frequency error by relaxation on coarse grids

Key realization: Solve for coarse-grid representation of error

- At any stage, error is reflected in residual:

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

- Don't transfer $Ax = b$ to coarse grid, transfer $Ae = r$

Multigrid

Multigrid Components

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

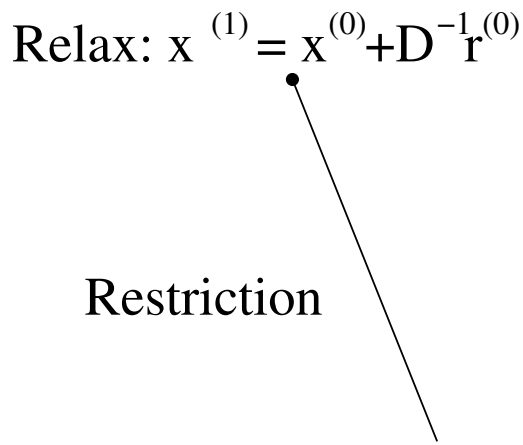
- Relaxation

- 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

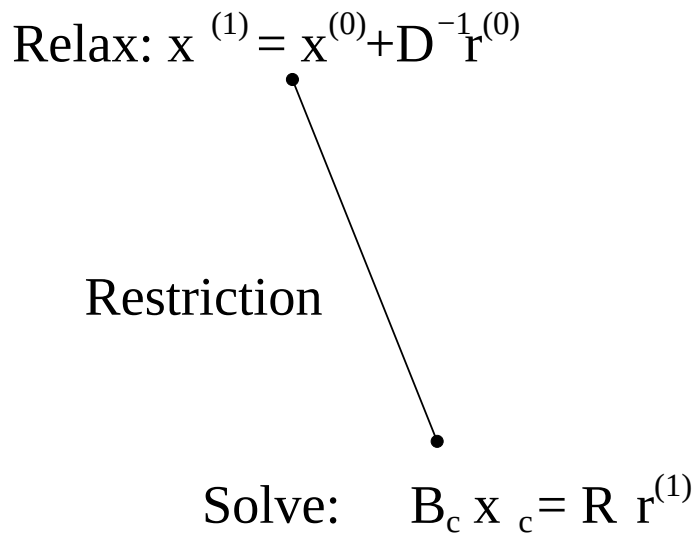


- Transfer residual to coarse grid
- Compute $Rr^{(1)}$

Multigrid

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction



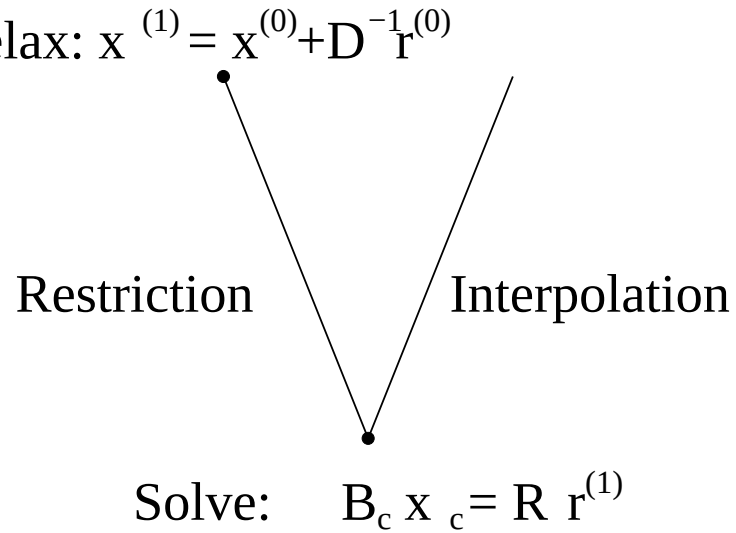
- Use coarse-grid correction to eliminate smooth errors

$$B_c x_c = R r^{(1)}$$

Multigrid

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation

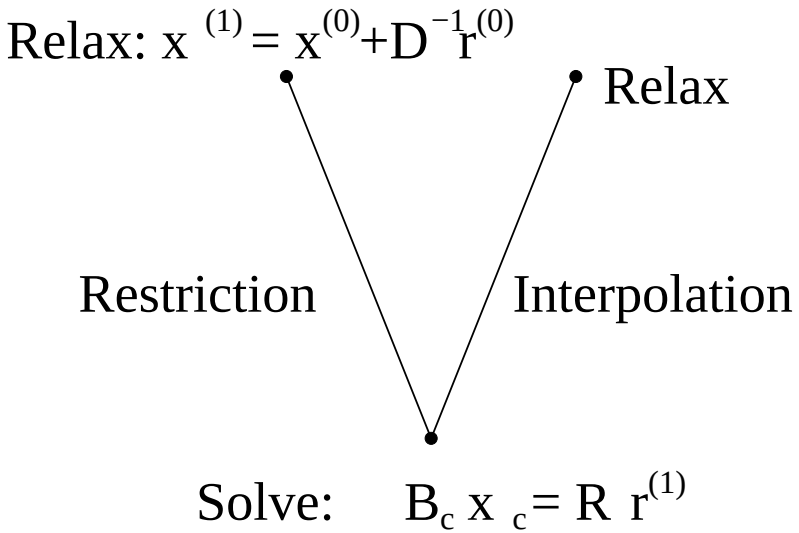


- Transfer correction to fine grid
- Compute $x^{(2)} = x^{(1)} + P x_c$

Multigrid

Multigrid Components

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

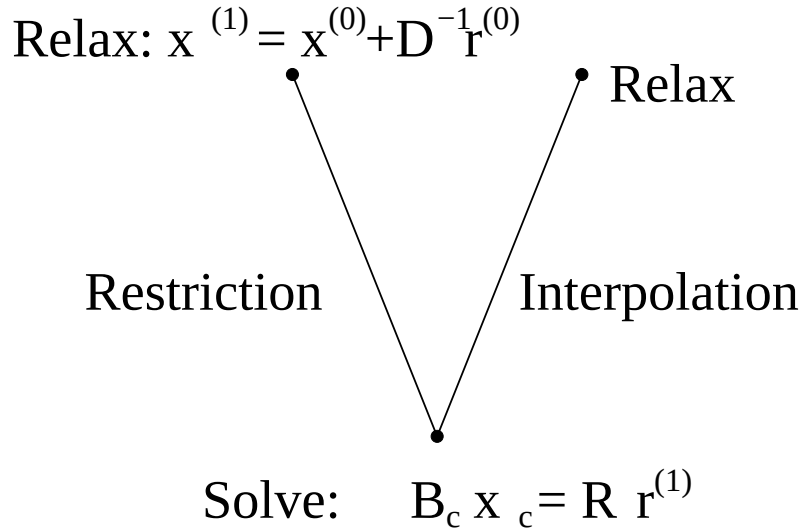


- Relax once again to remove oscillatory error introduced in coarse-grid correction

Multigrid

Multigrid Components

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



Direct solution of coarse-grid problem isn't practical

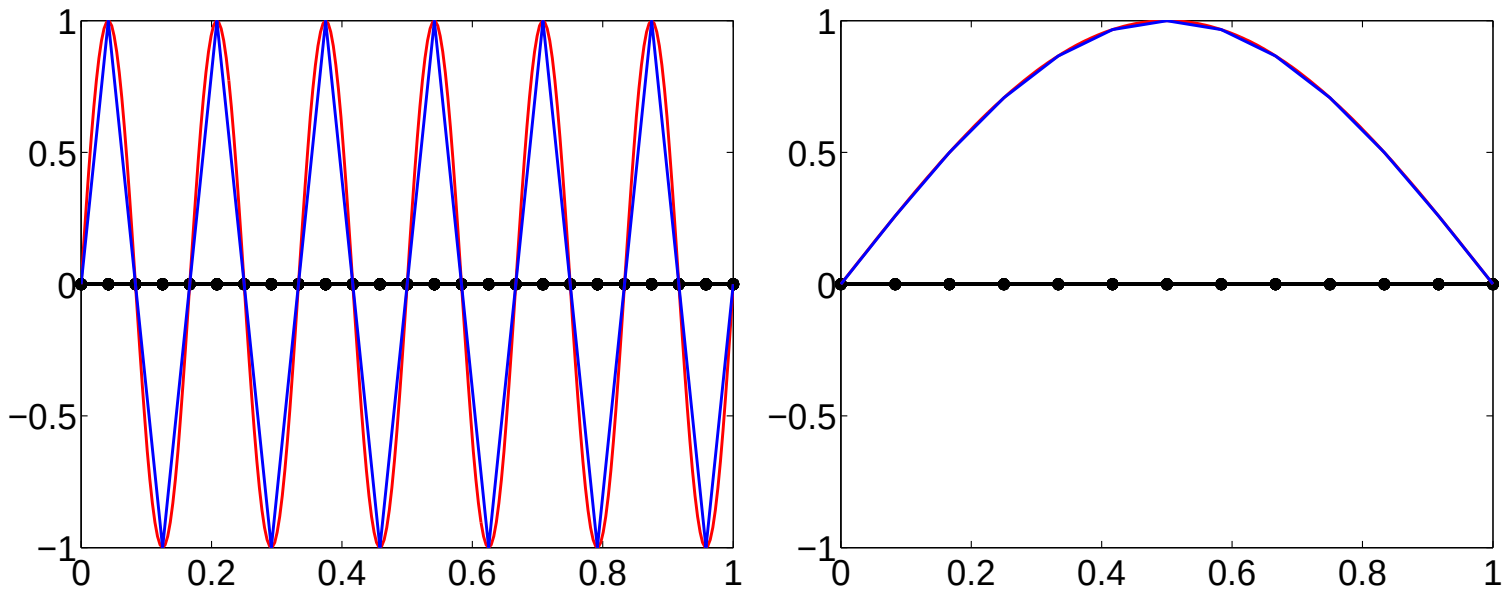
Recursion!

Apply same methodology to solve coarse-grid problem

Key to Success

Effective multigrid comes from complementarity

- Fixed relaxation effectively reduces certain types of error
- Coarse-grid correction must properly damp all complementary modes

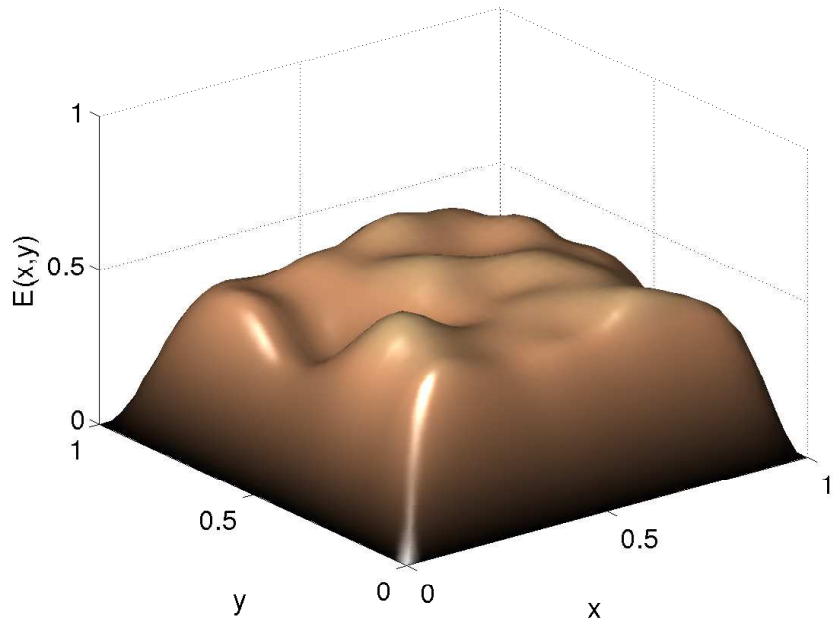


Knowledge of problem leads to good multigrid performance

Geometric Multigrid

For homogeneous operators, relaxation is predictable

- Jacobi/Gauss-Seidel relaxation
- Regular coarsening
- Linear interpolation



Fully explained by local mode (Fourier) analysis

Limitations of Geometric Approach

Geometric multigrid requires several assumptions on

- Problem geometry
- Form of operator
- Performance of relaxation

Limitations of Geometric Approach

Geometric multigrid requires several assumptions on

- Problem geometry
- Form of operator
- Performance of relaxation

These assumptions may be difficult to satisfy

- Heterogeneous coefficients
- Unstructured geometry
- Time-dependence
- Monte-Carlo simulations

Try to generalize algorithm to allow for heterogeneous coefficients and unstructured grids

Multigrid Without Grids

The essence of multigrid has nothing to do with grids!

Complementarity is key:

- Fix choice of relaxation
- For any A, some errors are slow to converge
- These errors must be corrected some other way

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

Multigrid Without Grids

The essence of multigrid has nothing to do with grids!

Complementarity is key:

- Fix choice of relaxation
- For any A, some errors are slow to converge
- These errors must be corrected some other way

Coarse-grid correction:

$$x \leftarrow x + PB_c^{-1}Rr$$
$$e \leftarrow e - PB_c^{-1}Rr$$

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

Variational Coarsening

Coarse-grid correction,

$$I - PB_c^{-1}RA,$$

can only correct errors in the range of P

Choosing $R = P^T$ and $B_c = P^T AP$ exactly eliminates errors in this space.

Complementarity is key:

- Errors reduced by relaxation and coarse-grid correction
- Errors that relaxation reduces slowly must be in range(P)

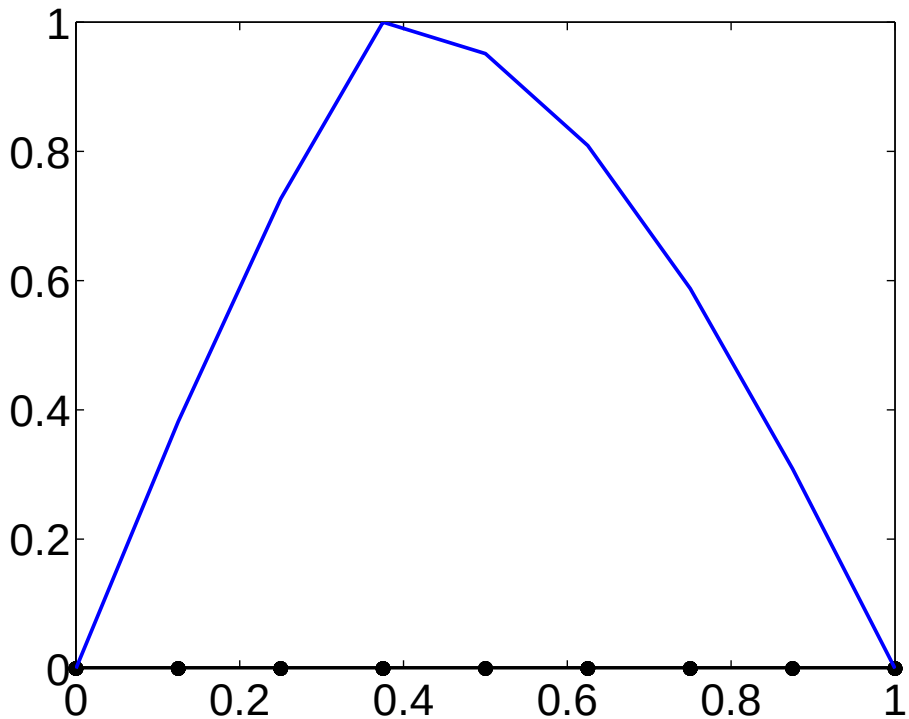
R. Nicolaides, Math. Comp. 1977, **31**:892-906

"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

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



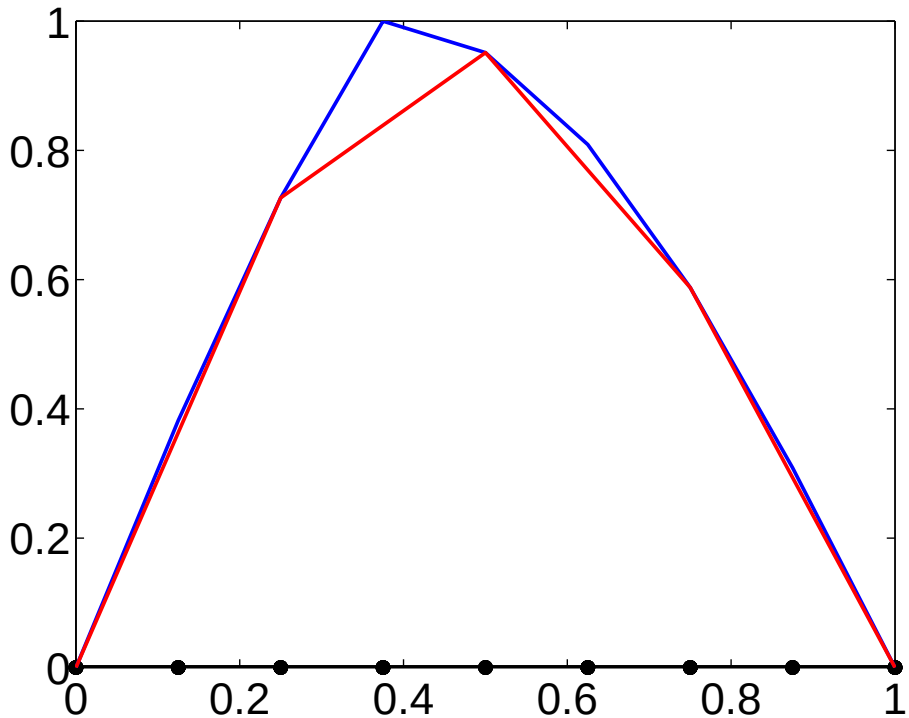
"Smooth" Errors

- 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

Algebraically Smooth Error

Slow to converge errors of relaxation replace smooth modes within AMG

Design interpolation to accurately represent these modes

- Assume these errors give small residuals, $Ae \approx 0$
- Expand residual equation:

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

- Use assumption on character of these errors to eliminate connections to $k \notin C$

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

Graph-based coarsening

Goal: Choose coarse-grid nodes to allow easy elimination of fine-fine connections

- Filter to eliminate small matrix entries
- Create graph of filtered matrix
- Greedy algorithm to choose maximal independent subset

Maximal independent subset ensures

- every fine-fine connection is "close" to a coarse-grid point
- coarse grid is small, but not too small

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

AMG working well

Bilinear finite element discretizations of $-\nabla \cdot \mathcal{K} \nabla p$

- Problem 1: $\mathcal{K} = 1$, Dirichlet BCs
- Problem 2: $\mathcal{K} = 1$, Neumann BCs
- Problem 3: $\mathcal{K}(x) = \begin{cases} 10^{-8} & x \in [\frac{1}{3}, \frac{2}{3}]^2, \\ 1 & \text{otherwise.} \end{cases}$
- Problem 4: $\mathcal{K}(x) = 10^{-8}$ on 20% of elements, chosen randomly, $\mathcal{K} = 1$ elsewhere

Asymptotic AMG V-cycle convergence factors

	128×128	256×256	512×512	1024×1024
Problem 1	0.115	0.124	0.131	0.137
Problem 2	0.069	0.070	0.071	0.071
Problem 3	0.122	0.130	0.136	0.141
Problem 4	0.212	0.233	0.290	0.375

AMG working badly

Same matrices, but symmetrically diagonally scaled by matrix, D, where

$$d_{ii} = 10^{5r_i}$$

for $\{r_i\}$ uniformly distributed on $[0, 1]$

Asymptotic AMG V-cycle convergence factors

	128×128	256×256	512×512	1024×1024
Problem 1r	0.997	0.996	0.996	0.996
Problem 2r	0.993	0.993	0.993	0.992
Problem 3r	0.997	0.996	0.996	0.996
Problem 4r	0.996	0.996	0.996	0.995

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; SISC 2006, **27**:1261-1286

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; SISC 2006, **27**:1261-1286

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

M. Brezina et al., SISC 2004, **25**:1896-1920; SISC 2006, **27**:1261-1286

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

M. Brezina et al., SISC 2004, **25**:1896-1920; SISC 2006, **27**:1261-1286

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

---

M. Brezina et al., SISC 2004, **25**:1896-1920; SISC 2006, **27**:1261-1286

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 $\|I - B_{\text{MG}}^{-1}A\|_{\text{est}}$ 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, 512×512 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

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

2D Dirac-Wilson normal equations
128×128 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

J. Brannick et al., *Proc. DD16*, 2007

Why does it work?

In principle, adaptive AMG is obvious:

- Fix relaxation, coarse-grid correction must complement
- Find out how relaxation fails, then build appropriate hierarchy

In practice, difficult to analyse

- Adaptive AMG interpolation depends nonlinearly on prototype
- Very dependent on coarse/fine partition

Simplifying AMG

Analyse simpler algorithm than full-blown AMG

- Ignore partition (assume properties of partition)
- Linearise dependence of adaptivity on prototype
- Directly link relaxation and interpolation

Start with partitioned matrix,

$$A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix},$$

derive and analyse AMG variant, then adaptivity

Reduction-Based AMG

Suppose we can partition the grid, $\Omega = F \cup C$, so that

$$x_f^T M_{ff} x_f \leq x_f^T A_{ff} x_f \leq \lambda_{\max} x_f^T M_{ff} x_f$$

and that $\begin{bmatrix} M_{ff} & -A_{fc} \\ -A_{cf} & A_{cc} \end{bmatrix}$ is positive semi-definite. Choose

Relaxation: $I - \frac{2}{1+\lambda_{\max}} \begin{bmatrix} M_{ff}^{-1} & 0 \\ 0 & 0 \end{bmatrix} A$

Coarse-grid correction: variational with $P = \begin{bmatrix} M_{ff}^{-1} A_{fc} \\ I \end{bmatrix}$

Then

$$\rho_{\mathrm{MG}} \leq \left(1 - \left(\frac{2}{\lambda_{\max} + 1}\right)^2\right)^{\frac{1}{2}}$$

M. Ries, U. Trottenberg, G. Winter, J. Lin. Alg. Applic., 1983
S. MacLachlan, T. Manteuffel, S. McCormick, Numer. Linear Algebra Appl. 2006.

Additive Multigrid

Theory for additive preconditioners has similar conditions.

Let

- $B = \begin{bmatrix} I & 0 \\ -A_{cf}M_{ff}^{-1} & I \end{bmatrix} \begin{bmatrix} M_{ff} & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & -M_{ff}^{-1}A_{fc} \\ 0 & I \end{bmatrix}$
- $\begin{bmatrix} M_{ff} & -A_{fc} \\ -A_{cf} & Acc \end{bmatrix}$ be positive semi-definite
- $x_f^T M_{ff} x_f \leq \lambda_{\min} x_f^T M_{ff} x_f \leq x_f^T A_{ff} x_f \leq \lambda_{\max} x_f^T M_{ff} x_f$
- $\nu_{\min} x_c^T S x_c \leq x_c^T (A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}) x_c \leq \nu_{\max} x_c^T S x_c$

Then,

$$\kappa(B^{-\frac{1}{2}} A B^{-\frac{1}{2}}) \leq \left(1 + \sqrt{1 - \frac{1}{\lambda_{\max}}}\right)^2 \frac{\lambda_{\max}^2 \nu_{\max}}{\min(\nu_{\min}, \lambda_{\min})}.$$

O. Axelsson, *Iterative Solution Methods*, 1994

Y. Saad and B. Suchomel, Numer. Linear Algebra Appl. 2002, **9**:359-378

Y. Notay, Numer. Linear Algebra Appl. 2005, **12**:419-451

Adaptive AMGr

- Key to success in AMGr is spectral equivalence

$$x_f^T M_{ff} x_f \leq x_f^T A_{ff} x_f \leq \lambda_{\max} x_f^T M_{ff} x_f$$

- Control cycle cost by controlling sparsity in M_{ff}

Adaptive AMGr:

- Fix sparsity of M_{ff}; e.g., take M_{ff} to be diagonal
- Adaptively choose entries in M_{ff} to ensure equivalence
  - Upper bound is easy (Gerschgorin)
  - Lower bound is difficult (algebraic smoothness)

S. MacLachlan, T. Manteuffel, S. McCormick, Numer. Linear Algebra Appl. 2006.

Adaptive setup algorithm

Adaptive stage in AMGr aims to compute lowest energy mode

- Multigrid approximation property says interpolation must be very accurate for this mode
- Good match needed for good spectral equivalence

Setup Algorithm:

1. Relax on $Ax = 0$
2. Define P such that $Px_c = x$
3. Compute $x^{(\text{new})} = P\left(\text{argmin}_{y_c} RQ(Py_c)\right)$.

S. MacLachlan, T. Manteuffel, S. McCormick, Numer. Linear Algebra Appl. 2006.

Adaptive Convergence

Adaptive iteration is nonlinear, so global theory is complicated

- Special case: $n_c = 2$, uniform global convergence
  - Convergence dependent on initial guess
- General case:

Suppose $A_{fc} \neq 0$, A_{ff} dominated by diagonal M_{ff}, SPSD A has one-dimensional null space, r.

Define Λ such that $\Lambda^{-1} A_{fc} x_f = A_{ff}^{-1} A_{fc} x_f$, use $P = \begin{bmatrix} \Lambda^{-1} A_{fc} \\ I \end{bmatrix}$ plus exact coarse-grid RQ minimization.

Then adaptive setup map is a contraction in a neighbourhood of $x = r$.

S. MacLachlan, T. Manteuffel, S. McCormick, Numer. Linear Algebra Appl. 2006.

Summary

- Effective multigrid arises by complementing relaxation with appropriate coarse-grid correction
- For simple problems, use simple corrections
- When details are complicated or unknown, AMG often helps
- AMG implicitly assumes certain properties of relaxation
- When these assumptions are wrong, adaptive AMG can restore good performance
- Good performance for difficult problems
- AMGr is a new theoretical framework
  - simpler than AMG
  - allows analysis of adaptive process

Current directions

- AMGr is nice, but it isn't AMG
  - Extend AMGr-type theory to AMG itself, or closer variants
- Use insight from AMGr to improve coarsening/relaxation in AMG
  - Adaptive AMG focuses on interpolation, but theory gives other good insights
- Apply and tune adaptive AMG for specific applications
  - QCD application challenges even best adaptive solvers
- Bring adaptive and algebraic ideas back to simpler multigrid solvers
  - Can we get the benefits without the costs?