Multigrid solvers for quantum dynamics - a first look

Scott MacLachlan

Delft Institute of Applied Mathematics, TU-Delft
and
Centrum voor Wiskunde en Informatica, Amsterdam

scott.maclachlan@gmail.com

April 26, 2007

Support

This work is in collaboration with

- K. Oosterlee of TU Delft and CWI
- J. Brannick and L. Zikatanov from Penn State
- R. Brower, M. Clark, J. Osborne, and C. Rebbi of Boston University
- M. Brezina, C. Ketelsen, T. Manteuffel, S. McCormick, and J. Ruge of CU-Boulder
- R. Falgout of CASC-LLNL
- A. Bessen and D. Keyes from Columbia

This research was supported by the European Community's Sixth Framework Programme, through a Marie Curie International Incoming Fellowship, MIF1-CT-2006-021927.

What is QCD?

Quantum Chromodynamics is the theory of the strong interaction

- Part of the Standard model of particle physics
- Describes interactions between quarks and gluons
- Consistent with particle accelerator experiments

We generally believe QCD to be true,
but look to make consistent predictions to check it

What is QCD?

Quantum Chromodynamics is the theory of the strong interaction

- Part of the Standard model of particle physics
- Describes interactions between quarks and gluons
- Consistent with particle accelerator experiments

We generally believe QCD to be true,
but look to make consistent predictions to check it

- Accelerators measure particle masses, decay rates, ...
- Use QCD theory to make predictions
- QCD not amenable to asymptotic analysis
- Need high-precision numerical simulations

What's a Quark?

Quarks are elementary particles

- Protons/Neutron composed of three quarks
- Each quark has a flavour
  - Up, Down, Strange, Charm, Top, Bottom
- Quarks bound together by strong force
  - Force between quarks grows as they move apart

We characterize quarks by their

- Spin (4 degrees of freedom)
- Colour (3 degrees of freedom)

QCD Resources

Accelerators:

- Fermilab Tevatron: \$120 million (in 1983)
- Large Hadron Collider: €2 billion (in 2007)
- International Linear Collider: est'd \$6.65 billion

Computational:

- QCDOC: 3 systems at 10 Teraflops each
- APEnext: 7 TeraFlops
- Commodity clusters in USQCD: $\sim$ 10 Teraflops
- Blue Gene, Cray XT3

The Continuum Challenge

Compute:

$$\langle \mathcal{O}(A, \psi, \overline{\psi}) \rangle = \frac{1}{Z} \int \mathcal{O}(A, \psi, \overline{\psi}) e^{-S_{pg} - S_F} d\psi d\overline{\psi} dA$$

where

- $\langle \mathcal{O} \rangle$ is the expected value of $\mathcal{O}$
- $A = A_\mu(\mathbf{x}) \in \mathbb{C}^{3\times 3}$ is the gauge potential
- $\psi(\mathbf{x}), \overline{\psi}(\mathbf{x})$ are Grassman-valued fermion fields
- $Z = \int e^{-S_{pg} - S_F} d\psi d\overline{\psi} dA$
- $S_{pg} = S_{pg}(A)$ is the "pure gauge" action
- $S_F = \int_{\mathbf{x},\mathbf{y}} \overline{\psi}(\mathbf{x}) M(A) \psi(\mathbf{y})$ is the fermionic action

The Dirac Operator

M is block-structured in colour-spin

$$M(A) = \sum_{\mu=1}^{4} (\gamma_\mu \otimes (I_3 \partial_\mu - \imath A_\mu)) - m I_{12}$$

where

- $\mu = 1, \ldots, 4$ represent space-time directions
- γ_μ are unitary 4×4 matrices
- I_3 and I_{12} are the 3×3 and 12×12 identities
- ∂_μ is a regular partial derivative
- $A_\mu(\mathbf{x}) \in \mathbb{C}^{3\times 3}$ is the gauge potential
- m is a mass term

Block Form

Writing $D_\mu = I_3 \partial_\mu - \imath A_\mu$,

$$M = \begin{pmatrix} -mI_3 & 0 & \imath D_3 - D_4 & \imath D_1 - D_2 \\ 0 & -mI_3 & \imath D_1 + D_2 & -\imath D_3 - D_4 \\ -\imath D_3 - D_4 & -\imath D_1 + D_2 & -mI_3 & 0 \\ -\imath D_1 - D_2 & \imath D_3 - D_4 & 0 & -mI_3 \end{pmatrix}$$

Notice that ${D_\mu}^\star = -D_\mu$, so

$$\begin{pmatrix} I_3 & 0 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & -I_3 & 0 \\ 0 & 0 & 0 & -I_3 \end{pmatrix} M \text{ is a Hermitian operator}$$

Perturbative Methods

Early success in quantum field theory came from asymptotic analysis

Quantum Electrodynamics (QED)

- Predictions accurate to 13 digits!

QCD resists this approach

Perturbative Methods

Early success in quantum field theory came from asymptotic analysis

Quantum Electrodynamics (QED)

- Predictions accurate to 13 digits!

QCD resists this approach

Discretize

- Space-time becomes discrete lattice
- Gauge potential, A_μ, integrated over lattice links

$$A_\mu(\mathbf{x}) \rightarrow U(\mathbf{x}, \mu) = e^{-\imath g A_\mu(\mathbf{x}_j)}$$

The Discrete Challenge

Compute:

$$\langle \mathcal{O}(U,\psi,\overline{\psi})\rangle = \frac{1}{Z}\int \mathcal{O}(U,\psi,\overline{\psi}) e^{-S_{pg}-S_F}\, d\psi d\overline{\psi} dU$$

where

- $\langle \mathcal{O} \rangle$ is the expected value of $\mathcal{O}$
- $U(\mathbf{x},\mu) = e^{-\imath g A_\mu(\mathbf{x})}$ is the lattice gauge field
- $\psi(\mathbf{x}), \overline{\psi}(\mathbf{x})$ are Grassman-valued fermion fields
- $Z = \int e^{-S_{pg}-S_F}\, d\psi d\overline{\psi} dU$
- $S_{pg} = S_{pg}(U)$ is the "pure gauge" action
- $S_F = \sum_{\mathbf{x},\mathbf{y}} \overline{\psi}(\mathbf{x}) M(U) \psi(\mathbf{y})$ is the fermionic action

Simplifying the integrals

"Easy" to simplify Z: $\int e^{-S_F} d\psi d\overline{\psi} = \det(M)$

$$Z = \int \det(M(U)) e^{-S_{pg}(U)} dU = \int e^{-S_{pg}^{eff}(U)} dU$$

for $S_{pg}^{eff} = S_{pg}(U) + Tr(\log(M(U)))$

Simplifying the integrals

"Easy" to simplify Z: $\int e^{-S_F} d\psi d\overline{\psi} = \det(M)$

$$Z = \int \det(M(U)) e^{-S_{pg}(U)} dU = \int e^{-S_{pg}^{eff}(U)} dU$$

for $S_{pg}^{eff} = S_{pg}(U) + Tr(\log(M(U)))$

Similar simplification for certain $\mathcal{O}(U, \psi, \overline{\psi})$:

$$\langle \psi(\mathbf{y}) \overline{\psi}(\mathbf{x}) f(U) \rangle = \frac{1}{Z} \int e^{-S_{pg}^{eff}(U)} f(U) M^{-1}(\mathbf{y}, \mathbf{x}) dU$$

In general, integrate out fermion fields:

$$\langle \mathcal{O}(U, \psi, \overline{\psi}) \rangle = \frac{1}{Z} \int e^{-S_{pg}^{eff}} \mathcal{O}^{eff}(U, M^{-1}(U)) dU$$

Integrating over U

Use Monte-Carlo to integrate over U:

- Assume we can generate gauge fields with given probability

$$P(U_k) = \frac{1}{Z} e^{-S_{pg}^{eff}(U_k)}$$

Then,

$$\langle \mathcal{O}(U, \psi, \overline{\psi}) \rangle \approx \frac{1}{N} \sum_{k=1}^{N} \mathcal{O}^{eff}(U_k, M^{-1}(U_k))$$

Two challenges:

- How do we pick $\{U_k\}_{k=1}^{N}$?
- How do we evaluate $\mathcal{O}^{eff}(U_k, M^{-1}(U_k))$?

Discretization

Discretization of M must preserve several properties for consistency within Monte-Carlo scheme

Several choices for $D_\mu = I_3\partial_\mu - \imath A_\mu$:

$$
\begin{aligned}
D_\mu\psi^\nu(\mathbf{x}) &\approx \frac{1}{h}\left(U(\mathbf{x},\mu)\psi^\nu(\mathbf{x}+h\hat{\mu}) - \psi^\nu(\mathbf{x})\right) \\
&\approx \frac{1}{h}\left(\psi^\nu(\mathbf{x}) - U^\star(\mathbf{x}-h\hat{\mu},\mu)\psi^\nu(\mathbf{x}-h\hat{\mu})\right) \\
&\approx \frac{1}{2h}\left(U(\mathbf{x},\mu)\psi^\nu(\mathbf{x}+h\hat{\mu}) - U^\star(\mathbf{x}-h\hat{\mu},\mu)\psi^\nu(\mathbf{x}-h\hat{\mu})\right)
\end{aligned}
$$

where $\hat{\mu}$ is the unit-vector in the μ-direction

Wilson Matrix

Choosing central differences for $D_\mu \psi^\nu(\mathbf{x})$ leads to instability
(Analogy: Nodal vs. staggered discretizations of Stokes)

Modify M by adding

$$\frac{-h}{2} \sum_\mu D_\mu^2 \psi^\nu(\mathbf{x}) \approx \frac{1}{2h} \sum_\mu (-U(\mathbf{x} + h\hat{\mu}, \mu)\psi^\nu(\mathbf{x} + h\hat{\mu}) + 2\psi^\nu(\mathbf{x}) - U^\star(\mathbf{x} - h\hat{\mu}, \mu)\psi^\nu(\mathbf{x} - h\hat{\mu}))$$

to its diagonal, $M \longrightarrow \widetilde{M}$

Stabilized matrix, $\widetilde{M}$, is known as the Dirac-Wilson operator

Quark Propagators

M^{-1} is called a quark propagator

Typically, approximating $\mathcal{O}^{eff}(U_k, M^{-1}(U_k))$ requires computing several/many entries in $\widetilde{M}^{-1}(U_k)$

Quark Propagators

M^{-1} is called a quark propagator

Typically, approximating $\mathcal{O}^{eff}(U_k, M^{-1}(U_k))$ requires computing several/many entries in $\widetilde{M}^{-1}(U_k)$

After discretization, $\widetilde{M}$ has dimension $12 n_x^3 n_t$

- $n_x \times n_x \times n_x$ spatial lattice
- n_t points in time (4th space dimension)
- 3 colour indices per lattice point
- 4 spin indices per colour/lattice point

Numerical Linear Algebra Challenge

Solve $\widetilde{M}\psi_j = \eta_j$ for some collection, $\{\eta_j\}$

Size of $\{\eta_j\}$ varies with application

- May be only a few RHS
- May want all (or most) of $\widetilde{M}^{-1}$

Our interest: when $\{\eta_j\}$ is small enough that iterative methods are appropriate, but large enough that multigrid setup costs may be amortized

Challenges in Lattice QCD

Numerical challenges arise because

- Need large n_x, n_t for physical accuracy
- $\widetilde{M}$ is large, $12n_x^3 n_t$ degrees of freedom
- Gauge field, U, is very disordered

Challenges in Lattice QCD

Numerical challenges arise because

- Need large n_x, n_t for physical accuracy
- $\widetilde{M}$ is large, $12n_x^3 n_t$ degrees of freedom
- Gauge field, U, is very disordered

Assets in Lattice QCD

Opportunity for fast solvers exists because

- $\widetilde{M}$ is sparse, 108 nonzeros per row
- $\widetilde{M}$ is easily permuted to be Hermitian
- Many right-hand sides for each realization of U

Extreme Simplification

Consider

- 2D spatial lattice, instead of 4D space-time
- Single colour/spin per lattice site ($\Rightarrow$ scalar U)
- "Cold" Gauge field, $U(\mathbf{x}, \mu) \equiv 1$
- Mass term, $m = 0$

Extreme Simplification

Consider

- 2D spatial lattice, instead of 4D space-time
- Single colour/spin per lattice site ($\Rightarrow$ scalar U)
- "Cold" Gauge field, $U(\mathbf{x}, \mu) \equiv 1$
- Mass term, $m = 0$

Result:

2D Poisson equation:

$$\widetilde{M} \longrightarrow \begin{bmatrix} & \frac{-1}{h^2} & \\ \frac{-1}{h^2} & \frac{4}{h^2} & \frac{-1}{h^2} \\ & \frac{-1}{h^2} & \end{bmatrix}$$

Extreme Simplification

Consider

- 2D spatial lattice, instead of 4D space-time
- Single colour/spin per lattice site ($\Rightarrow$ scalar U)
- "Cold" Gauge field, $U(\mathbf{x}, \mu) \equiv 1$
- Mass term, $m = 0$

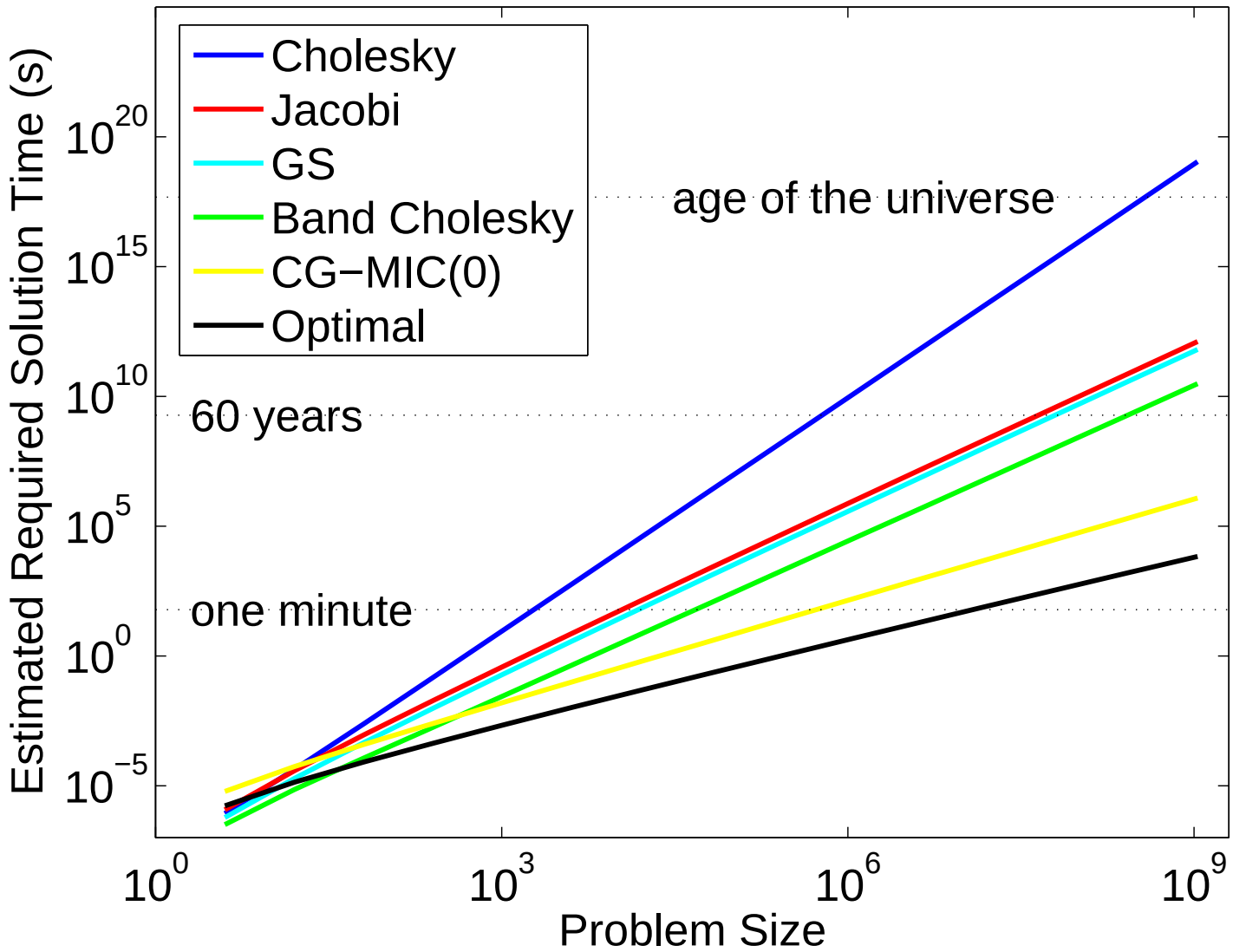
Result:

2D Poisson equation:

$$\widetilde{M} \rightarrow \begin{bmatrix} & \frac{-1}{h^2} & \\ \frac{-1}{h^2} & \frac{4}{h^2} & \frac{-1}{h^2} \\ & \frac{-1}{h^2} & \end{bmatrix}$$

Start search for good QCD solver with good Poisson solver

Scalability



Stationary Iterative Methods

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

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

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

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

Stationary Iterative Methods

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

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

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

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

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

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

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

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

Stationary Iterative Methods

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

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

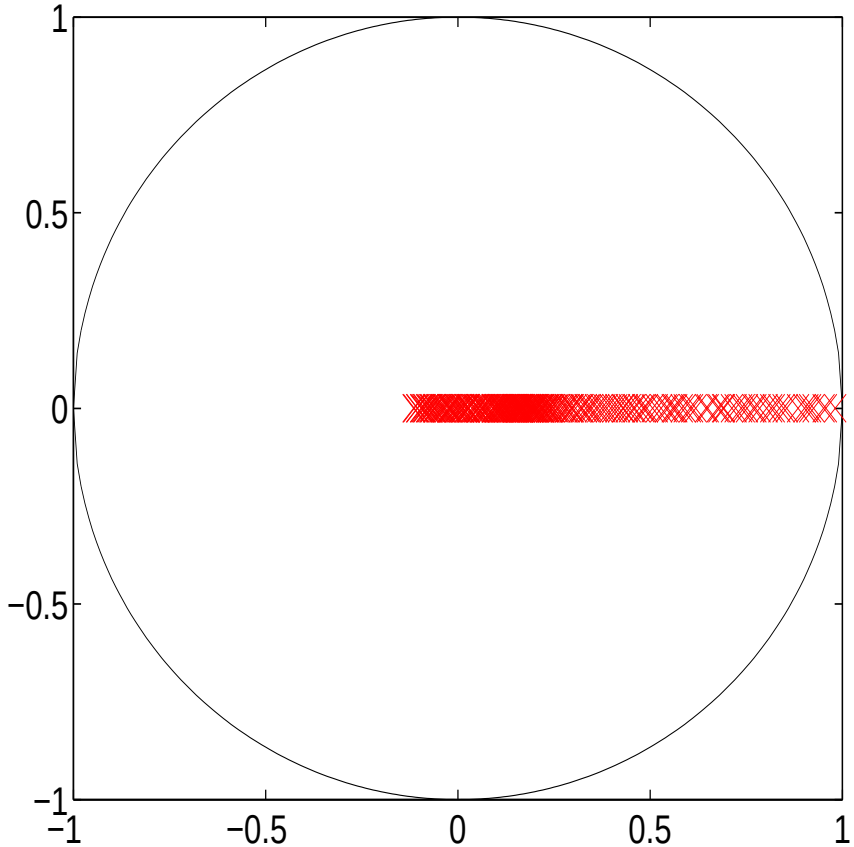
- Choose $B^{-1} \approx A^{-1}$
- Take $x^{(1)} = x^{(0)} + B^{-1}r^{(0)}$

Error propagation form:

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

Convergence of Stationary Iterations

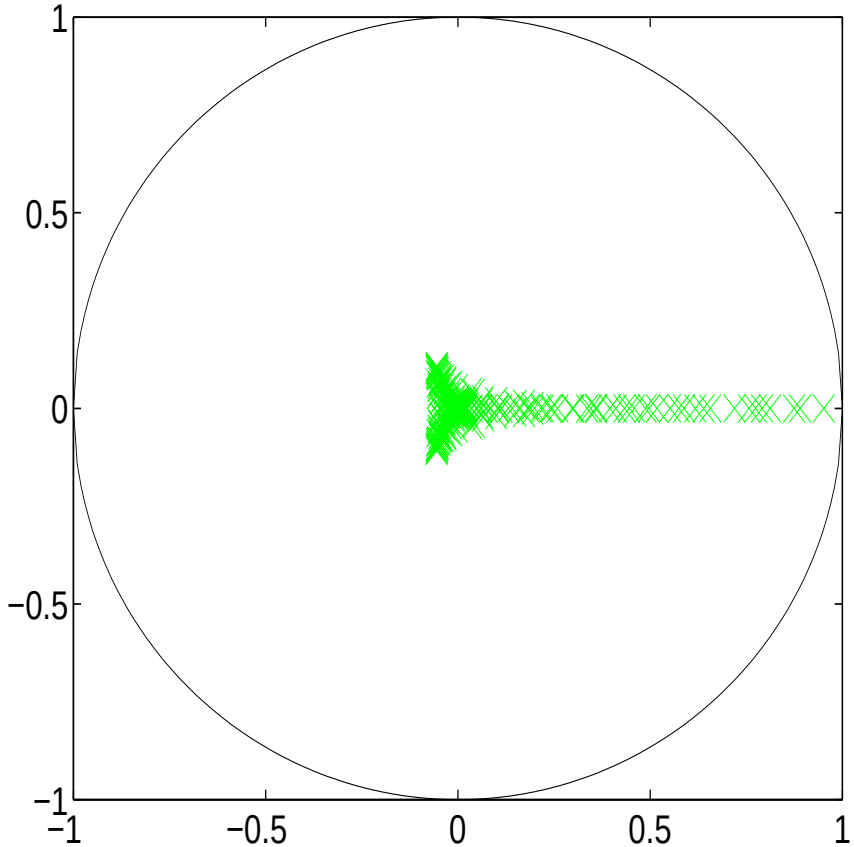
Convergence depends on spectrum of $I - B^{-1}A$



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

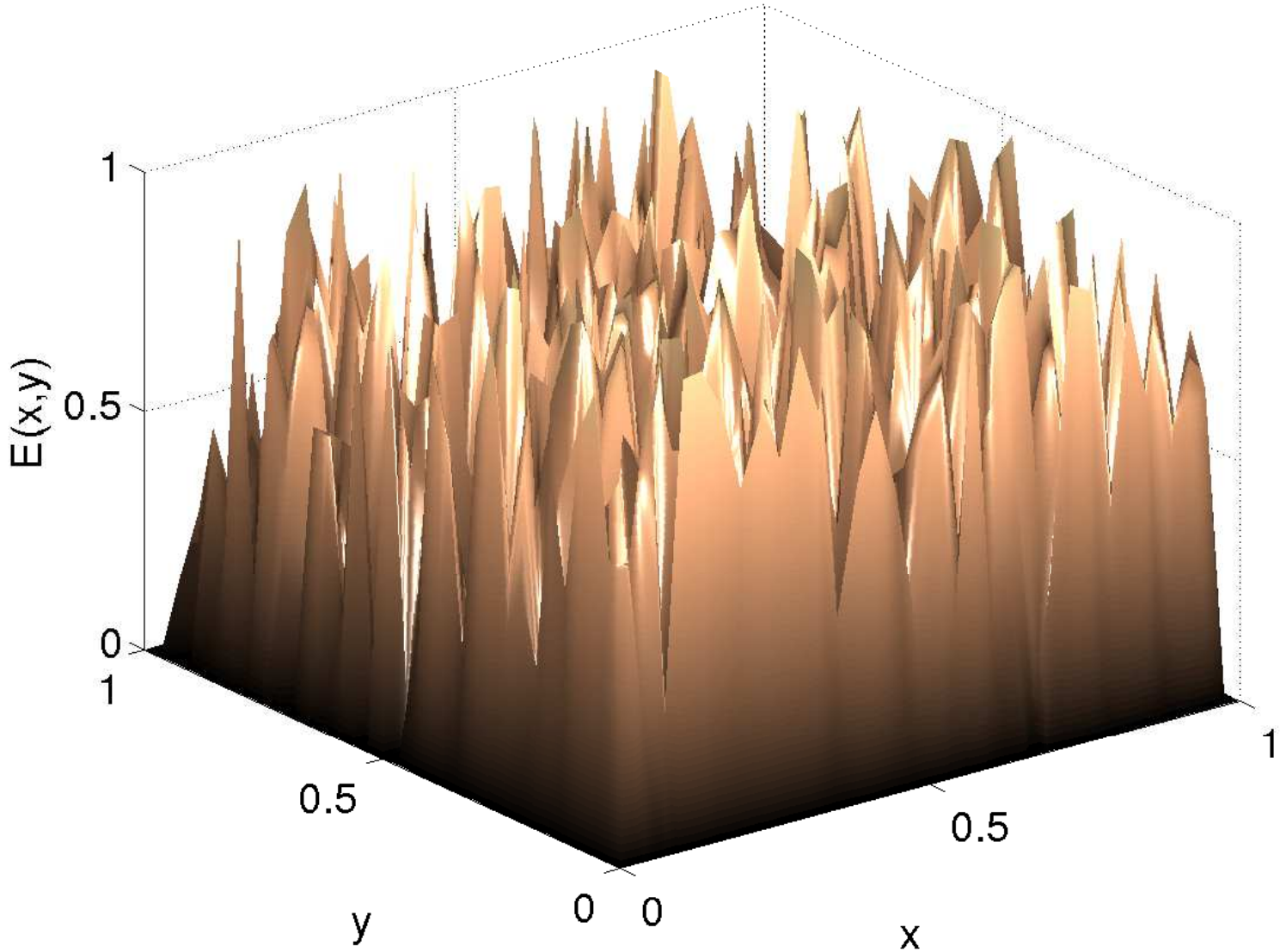
Convergence of Stationary Iterations

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



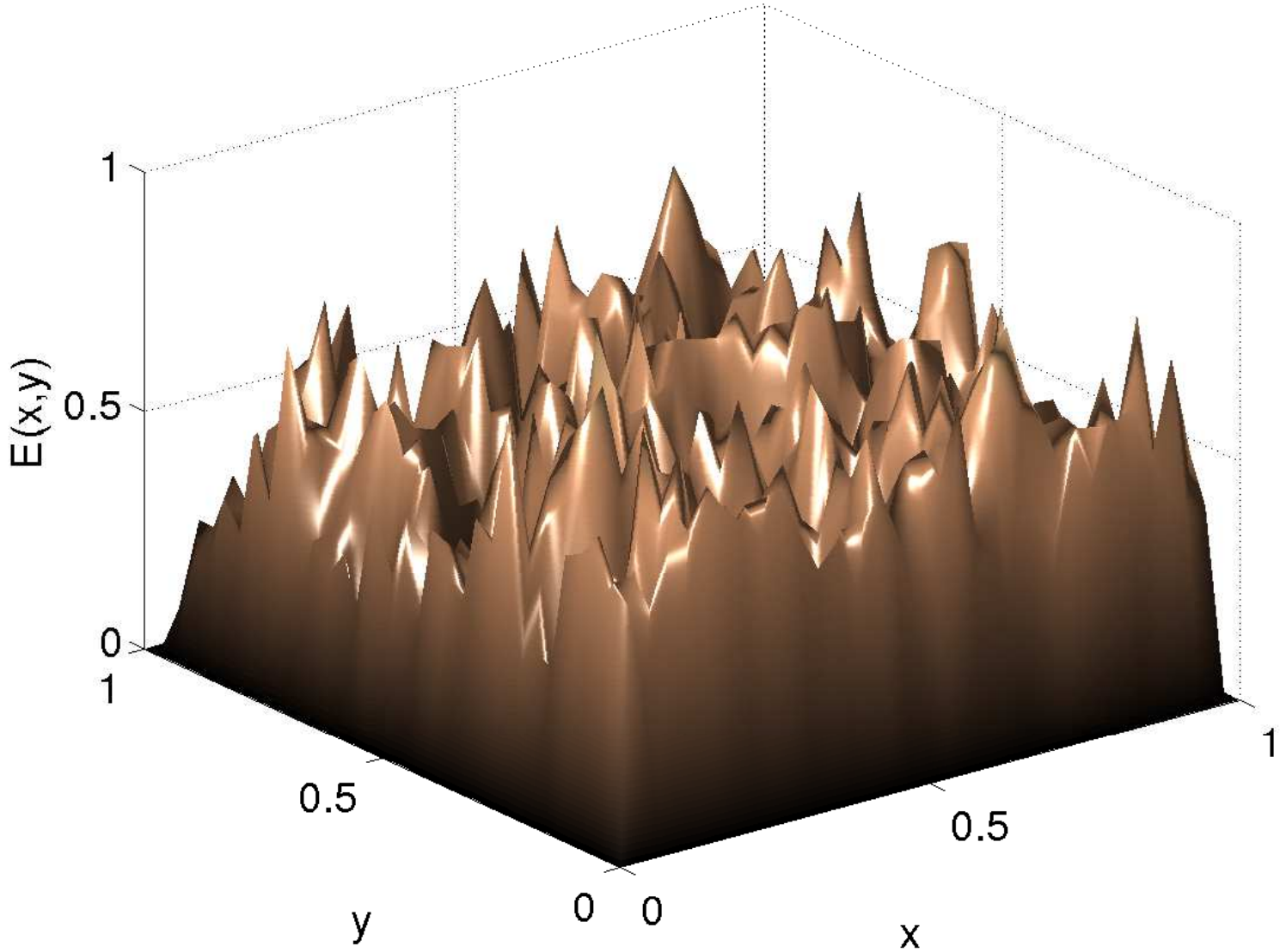
Gauss-Seidel Iteration: $e^{(n)} = (I - L^{-1}A)^n e^{(0)}$

Smoothing Property



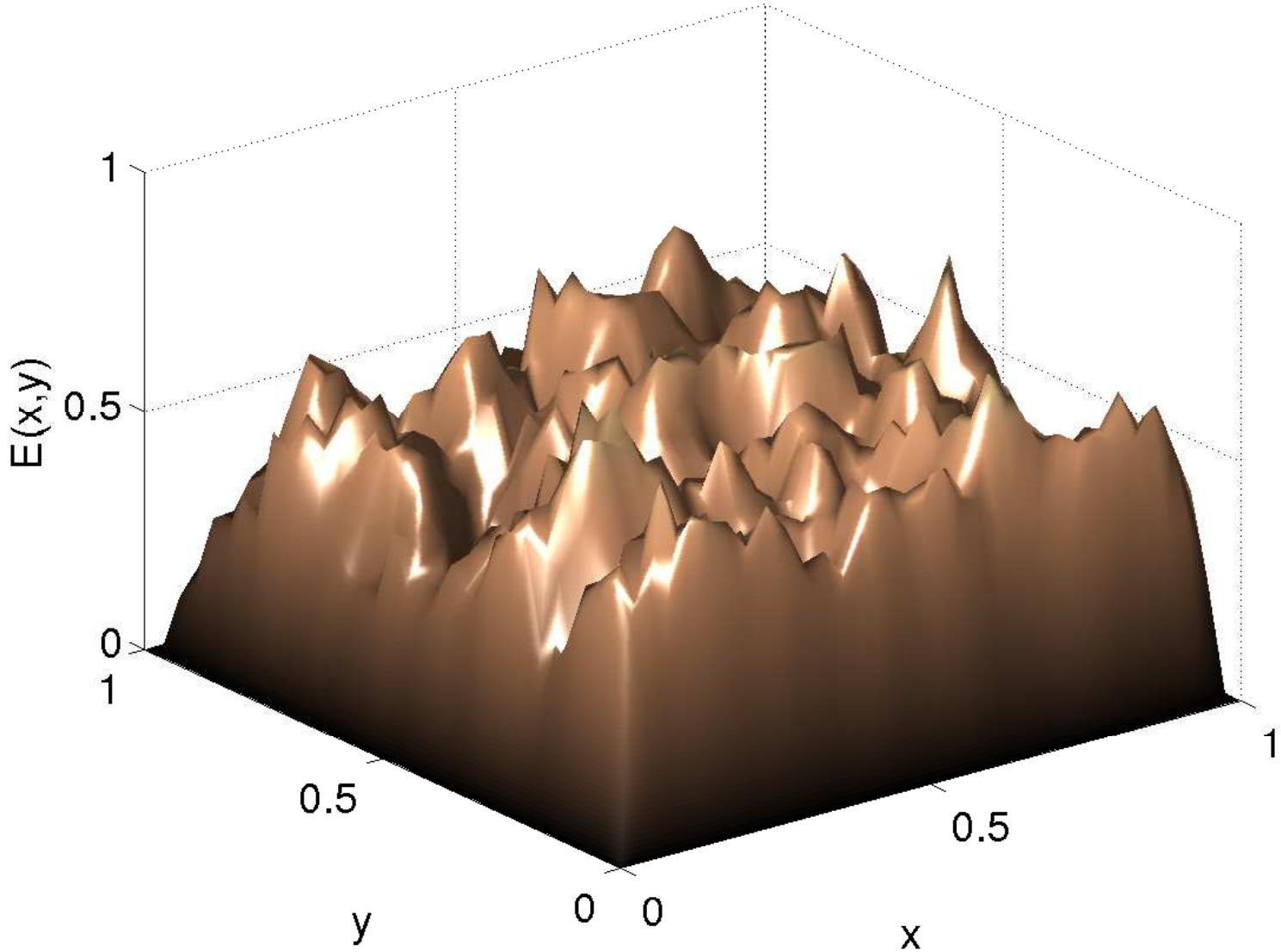
Random initial error

Smoothing Property



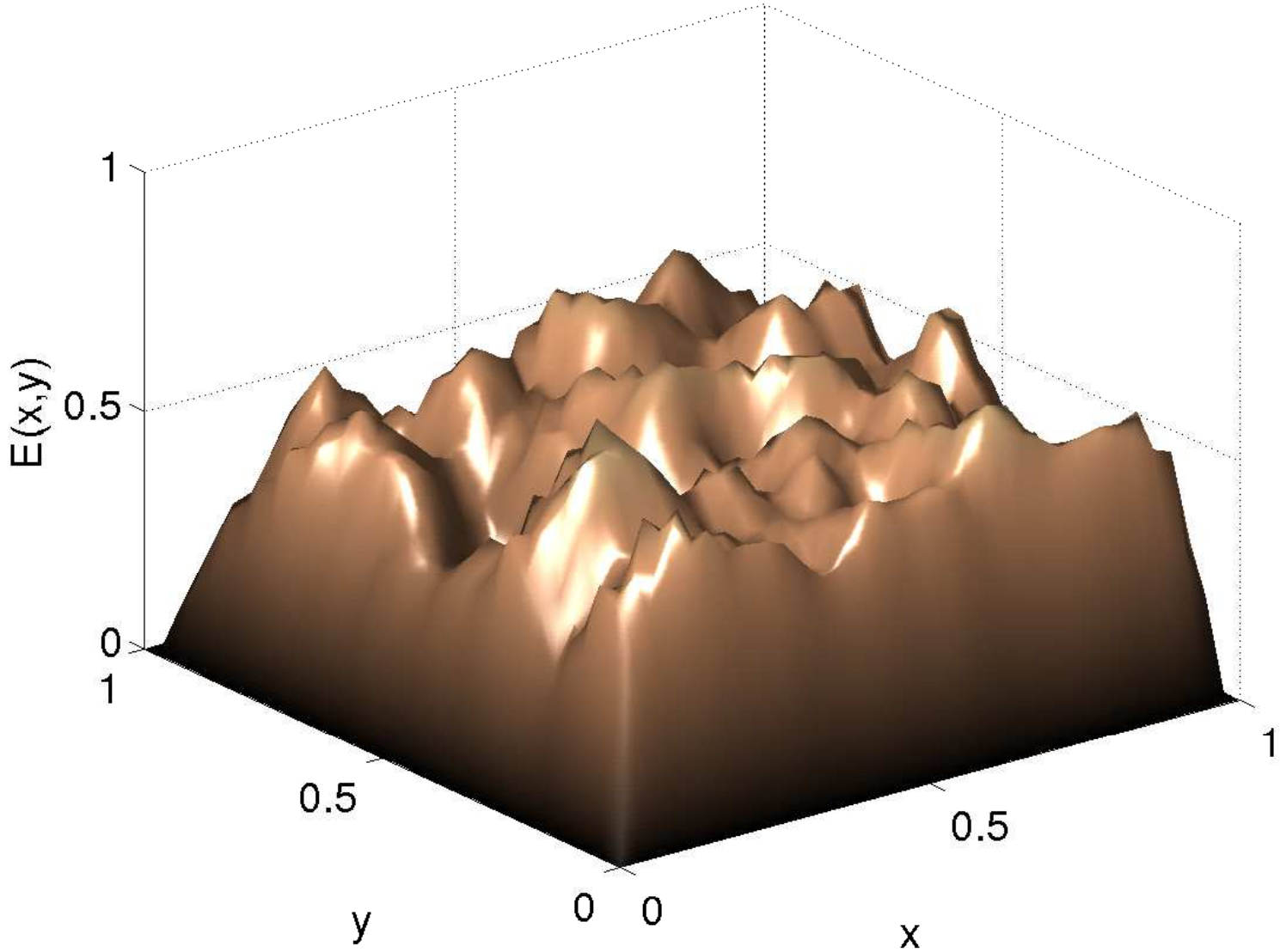
Error after 1 weighted Jacobi iteration

Smoothing Property



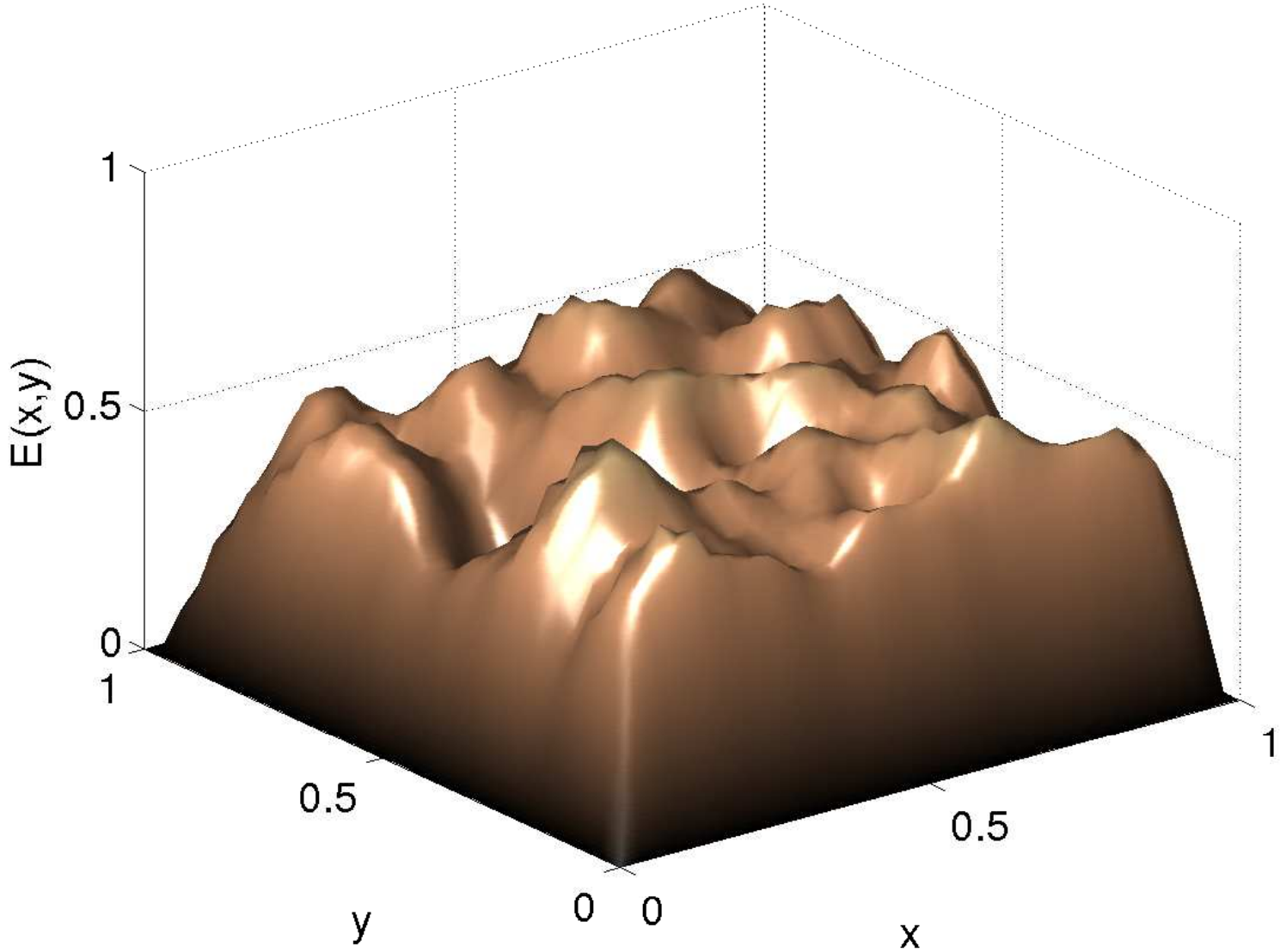
Error after 2 weighted Jacobi iterations

Smoothing Property



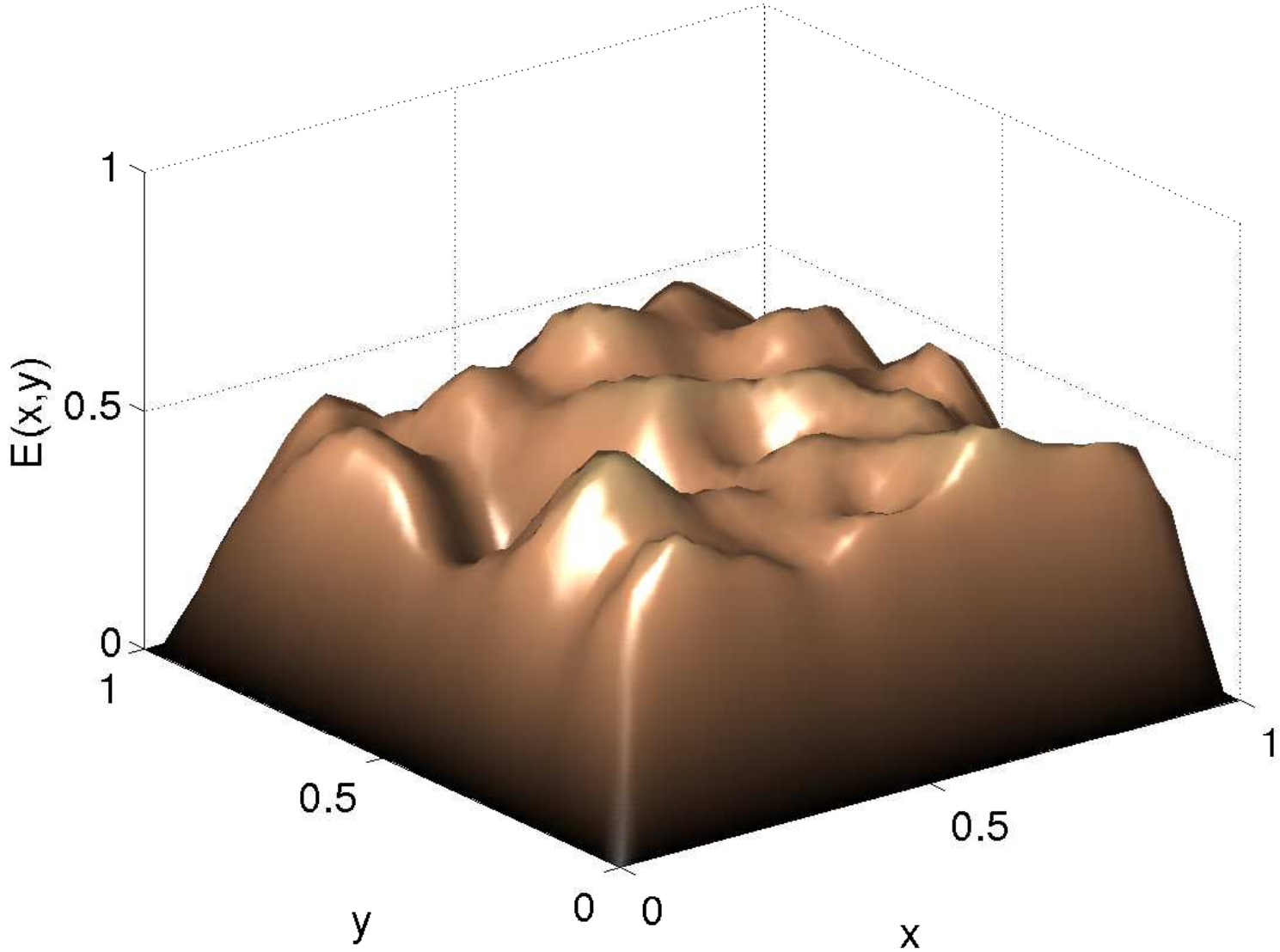
Error after 3 weighted Jacobi iterations

Smoothing Property



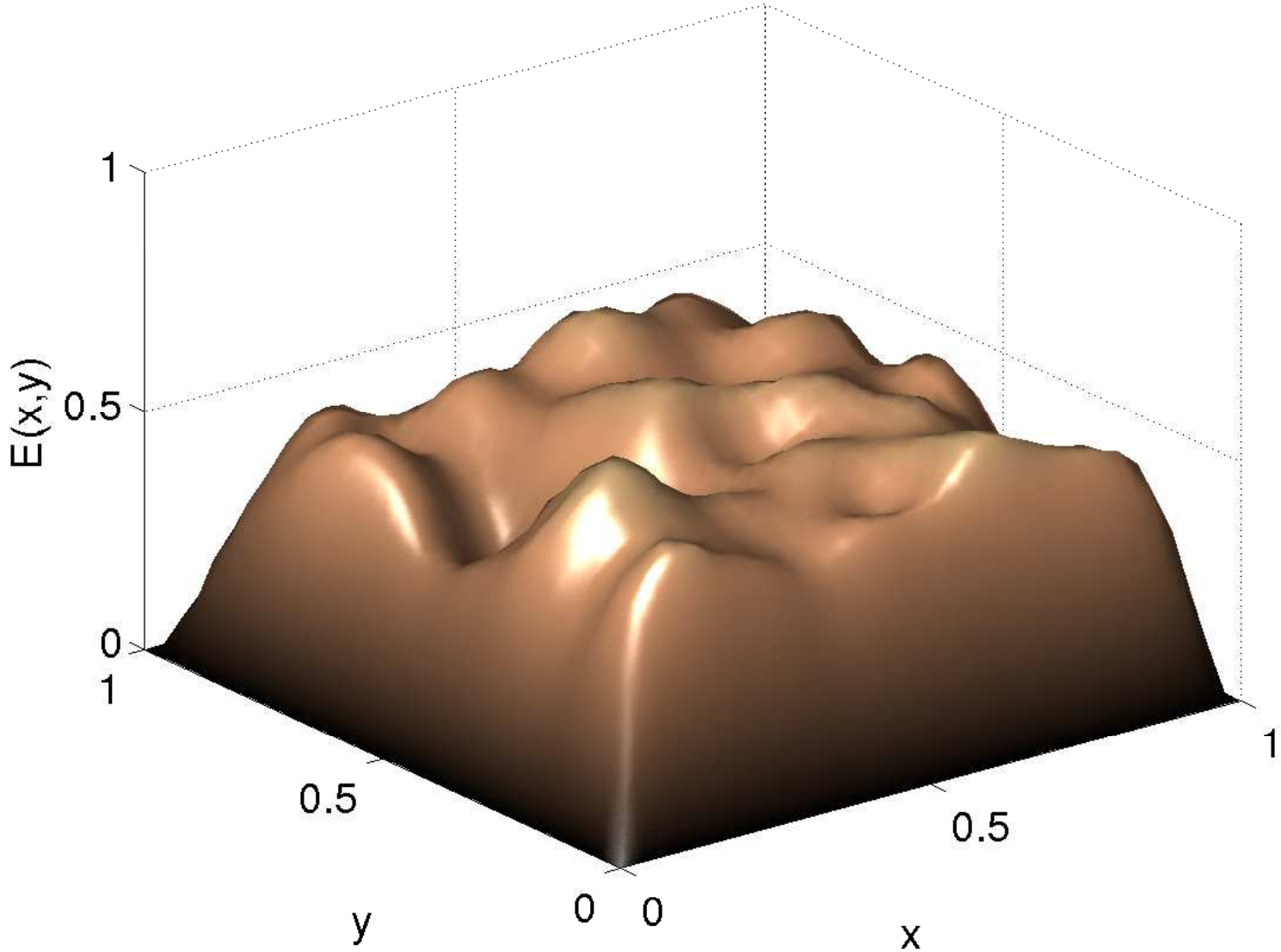
Error after 4 weighted Jacobi iterations

Smoothing Property



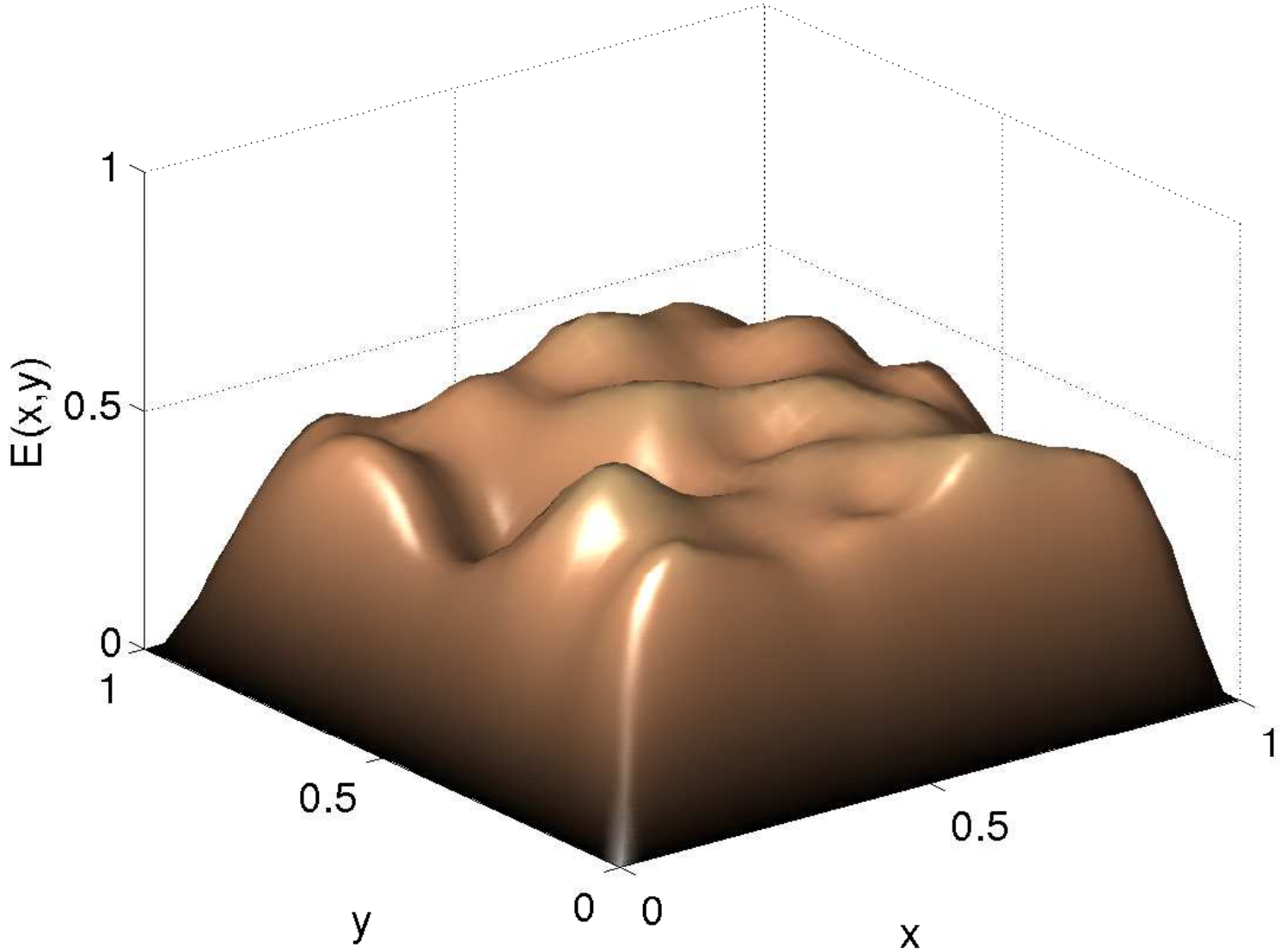
Error after 5 weighted Jacobi iterations

Smoothing Property



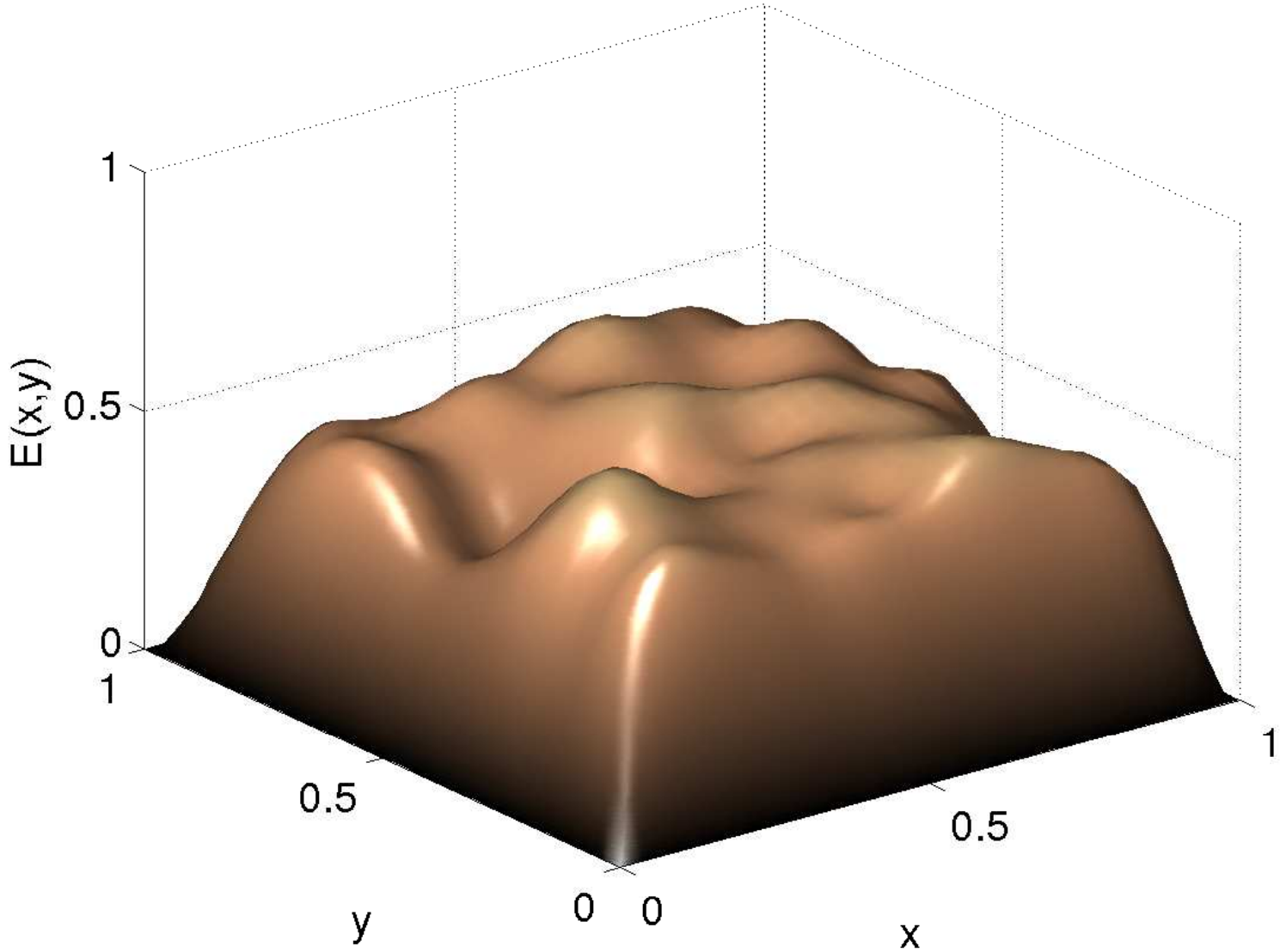
Error after 6 weighted Jacobi iterations

Smoothing Property



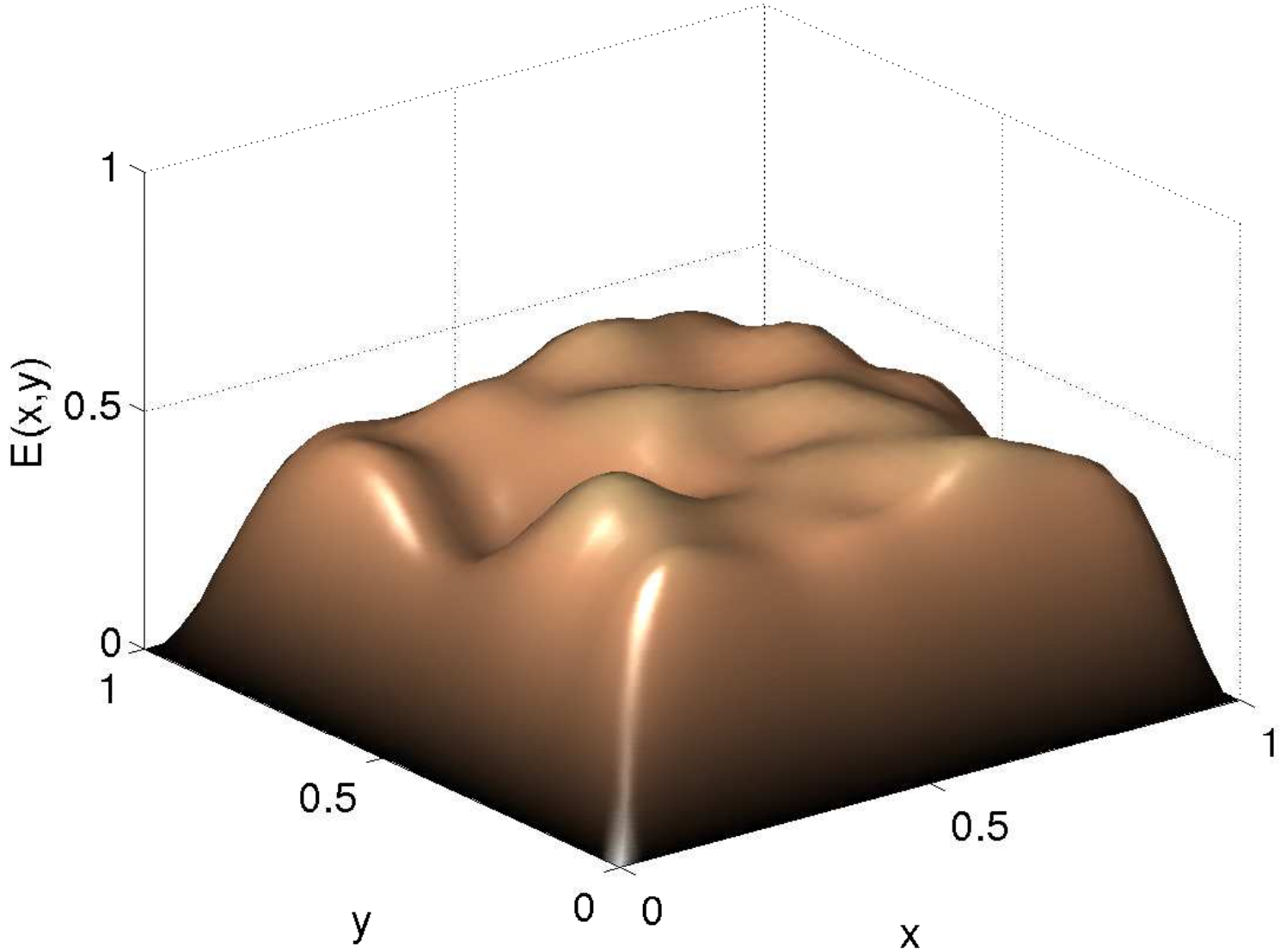
Error after 7 weighted Jacobi iterations

Smoothing Property



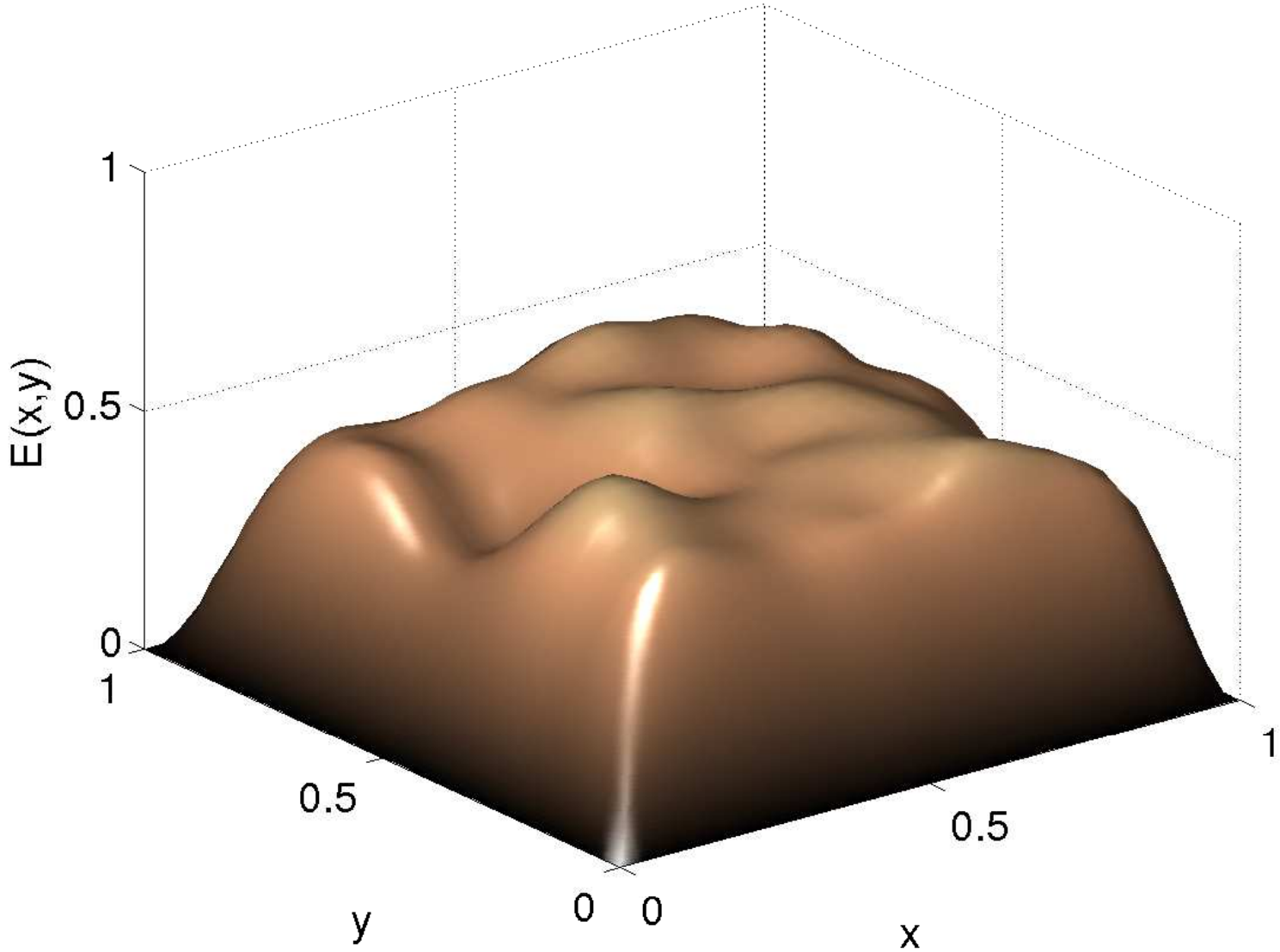
Error after 8 weighted Jacobi iterations

Smoothing Property



Error after 9 weighted Jacobi iterations

Smoothing Property



Error after 10 weighted Jacobi iterations

Complementarity

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

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

Complementarity

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

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

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

Coarse-Grid Correction

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

Variational Coarsening

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

What is the *best* x_c for correction?

A-norm and A-inner product

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

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

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

$$y^{\text{opt}} = \underset{y}{\text{argmin}} \, \|x - y\|_A$$

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 $(P^\star AP)x_c = P^\star A(x - x^{(1)}) = P^\star r^{(1)}$

Multigrid

Multigrid Components

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

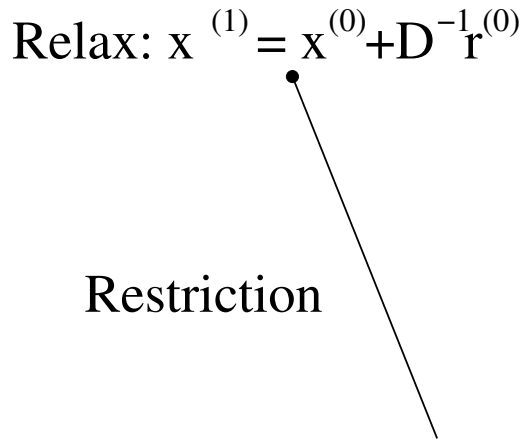
- Relaxation

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

Multigrid

Multigrid Components

- Relaxation
- Restriction

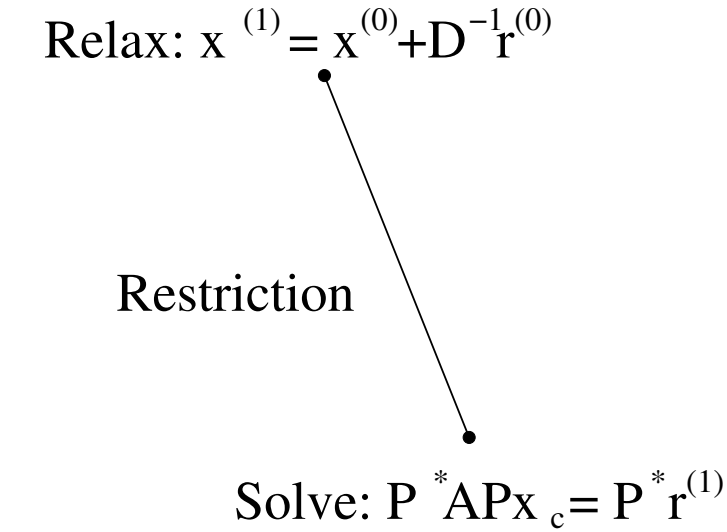


- Transfer residual to coarse grid
- Compute $P^{\star} r^{(1)}$

Multigrid

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction



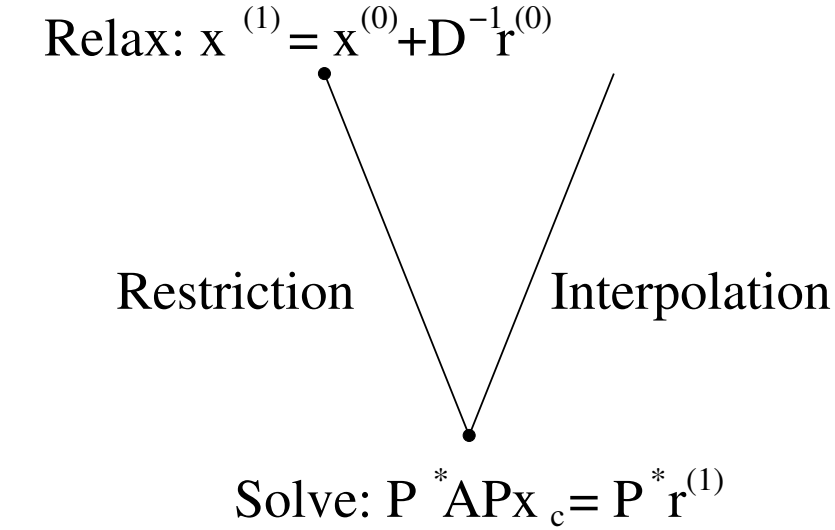
- Use coarse-grid correction to eliminate smooth errors
- Best correction, x_c, in terms of A-norm satisfies

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

Multigrid

Multigrid Components

- Relaxation
- Restriction
- Coarse-Grid Correction
- Interpolation

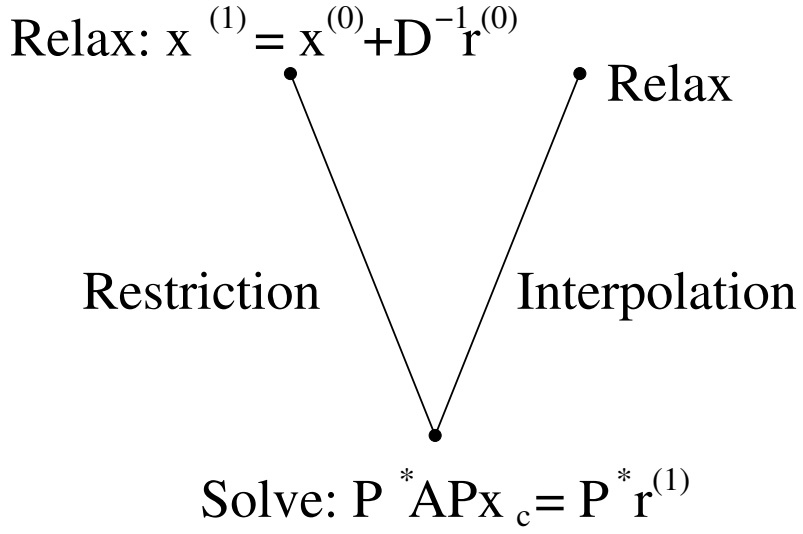


- Transfer correction to fine grid
- Compute $x^{(2)} = x^{(1)} + Px_c$

Multigrid

Multigrid Components

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

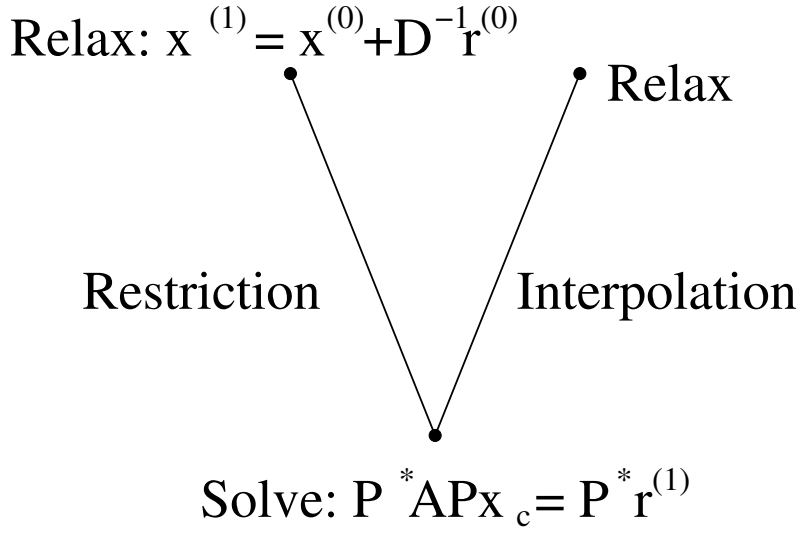


- Relax once again to remove oscillatory error introduced in coarse-grid correction

Multigrid

Multigrid Components

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



Direct solution of coarse-grid problem isn't practical

Recursion!

Apply same methodology to solve coarse-grid problem

Performance

- Uniform grid coarsening
- Bilinear Interpolation
- V(2,2) cycles, with under-relaxed Jacobi

grid	64^2	128^2	256^2	512^2
ρ_{MG}	0.164	0.165	0.165	0.165

Performance

- Uniform grid coarsening
- Bilinear Interpolation
- V(2,2) cycles, with under-relaxed Jacobi

grid	64^2	128^2	256^2	512^2
ρ_{MG}	0.164	0.165	0.165	0.165

Now relax simplifications:

- Allow $U(\mathbf{x}, \mu)$ to take physical (scalar) values
  - "Gauge Laplacian"

grid	64^2	128^2	256^2	512^2
ρ_{MG}	0.689	0.762	0.776	0.736

Accounting for Heterogeneity

Poor performance results from ignoring heterogeneity

Error after relaxation on Poisson's equation is smooth

- Low-order geometric interpolation is accurate

Accounting for Heterogeneity

Poor performance results from ignoring heterogeneity

Error after relaxation on Poisson's equation is smooth

- Low-order geometric interpolation is accurate

Geometric multigrid defines interpolation based on

- grid geometry
- assumptions on performance of relaxation

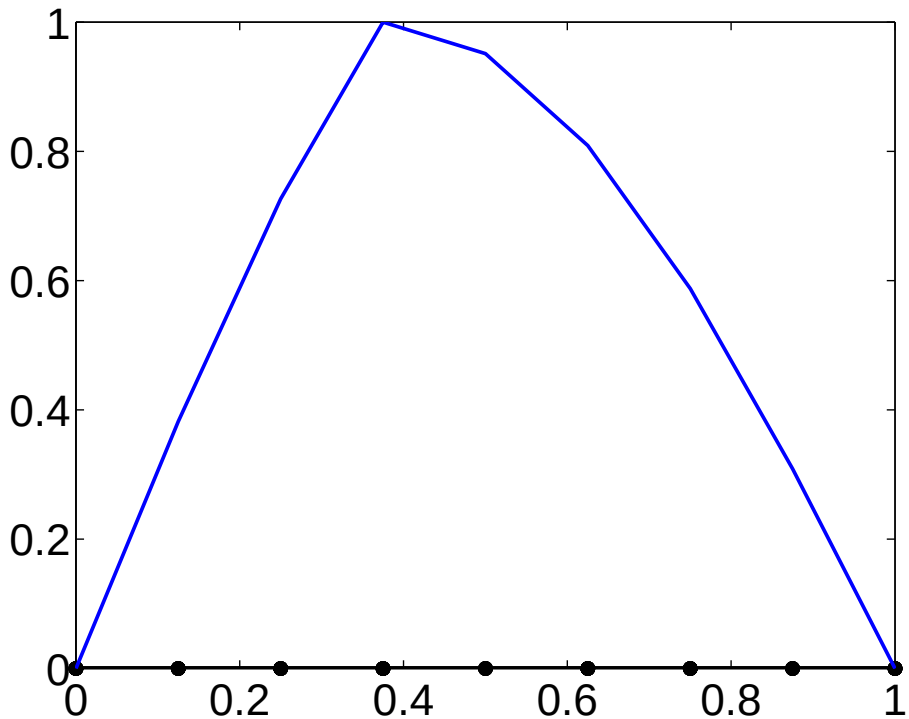
Heterogeneity strongly influences performance of relaxation

"Smooth" Errors

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

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

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



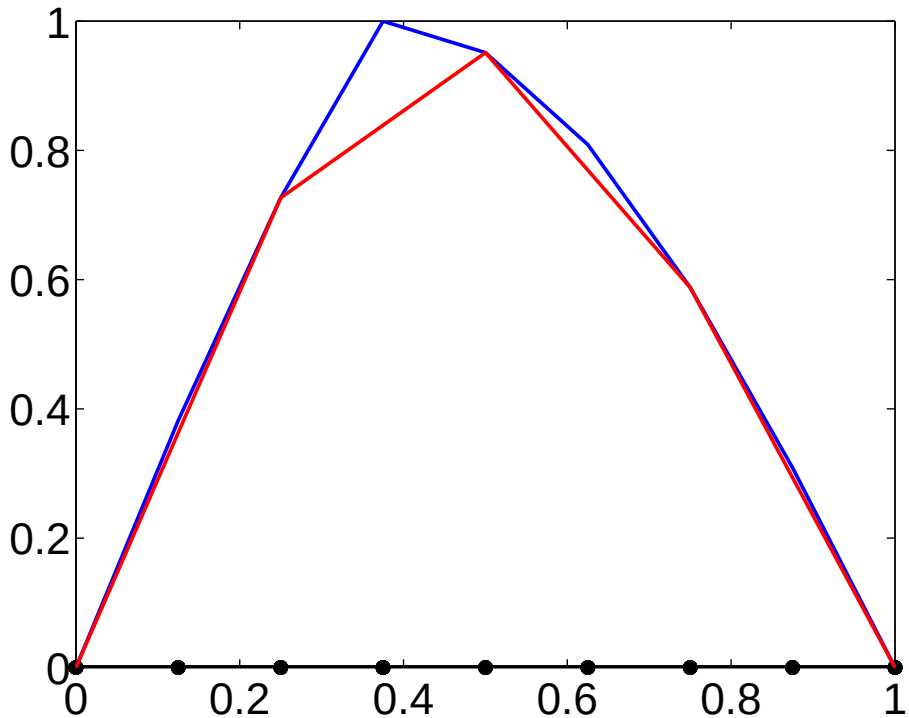
"Smooth" Errors

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

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

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

and linear interpolant from coarse grid



"Smooth" Errors

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

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

"Smooth" Errors

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

Algebraic Multigrid Interpolation

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

$$(Ae)_i \approx 0$$

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

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

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

Complex-Valued AMG

Gauge Laplacian is a Hermitian H-matrix

- A is an H-matrix if $\mathcal{M}(A)$ is an M-matrix,

$$(\mathcal{M}(A))_{ij} = \begin{cases} |a_{ii}| & \text{if } i = j \\ -|a_{ij}| & \text{if } i \neq j \end{cases},$$

- Jacobi/Gauss-Seidel converge for H-matrices
- Fourier analysis confirms (algebraic) smoothing properties
- Interpolation based on classical AMG
- Restriction as adjoint of interpolation
- Galerkin coarse-grid operators

S. MacLachlan and K. Oosterlee, *submitted*, 2007

Varga, Linear Algebra and Appl. 1976, **13**:1-9

Solving the Gauge Laplacian

Convergence Factors

grid	64^2	128^2	256^2	512^2
ρ_{MG}	0.689	0.762	0.776	0.736
ρ_{AMG}	0.277	0.378	0.404	0.390

Solving the Gauge Laplacian

Convergence Factors

grid	64^2	128^2	256^2	512^2
ρ_{MG}	0.689	0.762	0.776	0.736
ρ_{AMG}	0.277	0.378	0.404	0.390

Operator Complexities

grid	64^2	128^2	256^2	512^2
C_{MG}	1.63	1.62	1.61	1.60
C_{AMG}	2.68	2.59	2.55	2.52

Solving the Gauge Laplacian

Convergence Factors

grid	64^2	128^2	256^2	512^2
ρ_{MG}	0.689	0.762	0.776	0.736
ρ_{AMG}	0.277	0.378	0.404	0.390

Operator Complexities

grid	64^2	128^2	256^2	512^2
C_{MG}	1.63	1.62	1.61	1.60
C_{AMG}	2.68	2.59	2.55	2.52

Convergence factor per matvec equivalent

grid	64^2	128^2	256^2	512^2
ρ_{MG}^{eff}	0.944	0.959	0.961	0.953
ρ_{AMG}^{eff}	0.887	0.910	0.915	0.911

Shifted Gauge Laplacian

Now take realistic values of $U(\mathbf{x}, \mu)$ and m

$$\widetilde{M} \rightarrow \begin{bmatrix} & -U(\mathbf{x}, \binom{0}{1}) & \\ -U^{\star}(\mathbf{x} - \binom{h}{0}, \binom{1}{0}) & 4 - m & -U(\mathbf{x}, \binom{1}{0}) \\ & -U^{\star}(\mathbf{x} - \binom{0}{h}, \binom{0}{1}) & \end{bmatrix}$$

- $|U(\mathbf{x}, \mu)| = 1$ for all $\mathbf{x}$, μ
- For "physical" fields $U(\mathbf{x}, \mu)$ and $m = 0$, $\lambda_{\min}(\widetilde{M}) > 0$.
- Choose $m > 0$ so that $\lambda_{\min}(\widetilde{M}) \rightarrow 0$

Effect of Shifting

AMG Convergence Factors

$\lambda_{\min}(\widetilde{M})$	64^2	128^2	256^2	512^2
1.0	0.040	0.051	0.049	0.047
10^{-1}	0.475	0.598	0.579	0.563
10^{-2}	0.893	0.934	0.932	0.911
10^{-3}	0.988	0.993	0.993	0.990
10^{-4}	0.9988	0.9993	0.9993	0.9990

Algebraically Smooth Error

Slow-to-converge errors must be in range of interpolation

- AMG treats heterogeneity in coarse-grid correction
- Heterogeneity in $\widetilde{M}$ affects performance of relaxation
- As $\lambda_{\min}(\widetilde{M}) \longrightarrow 0$, performance of relaxation degrades
- As $\lambda_{\min}(\widetilde{M}) \longrightarrow 0$, accuracy of interpolation must increase

Algebraically Smooth Error

Slow-to-converge errors must be in range of interpolation

- AMG treats heterogeneity in coarse-grid correction
- Heterogeneity in $\widetilde{M}$ affects performance of relaxation
- As $\lambda_{\min}(\widetilde{M}) \longrightarrow 0$, performance of relaxation degrades
- As $\lambda_{\min}(\widetilde{M}) \longrightarrow 0$, accuracy of interpolation must increase

AMG process makes assumptions on relaxation for generality

- AMG assumptions are violated as $\lambda_{\min}(\widetilde{M}) \longrightarrow 0$

Krylov Acceleration

If only one bad eigenvalue, then CG acceleration should be effective

PCG Iterations so that $\|\mathbf{r}^{(k)}\|/\|\mathbf{r}^{(0)}\| < 10^{-10}$

$\lambda_{\min}(\widetilde{M})$	64^2	128^2	256^2	512^2
no shift	9	10	10	10

Krylov Acceleration

If only one bad eigenvalue, then CG acceleration should be effective

PCG Iterations so that $\|\mathbf{r}^{(k)}\|/\|\mathbf{r}^{(0)}\| < 10^{-10}$

$\lambda_{\min}(\widetilde{M})$	64^2	128^2	256^2	512^2
no shift	9	10	10	10
10^{-1}	12	14	13	13

Krylov Acceleration

If only one bad eigenvalue, then CG acceleration should be effective

PCG Iterations so that $\|\mathbf{r}^{(k)}\|/\|\mathbf{r}^{(0)}\| < 10^{-10}$

$\lambda_{\min}(\widetilde{M})$	64^2	128^2	256^2	512^2
no shift	9	10	10	10
10^{-1}	12	14	13	13
10^{-2}	24	34	26	28

Krylov Acceleration

If only one bad eigenvalue, then CG acceleration should be effective

PCG Iterations so that $\|\mathbf{r}^{(k)}\|/\|\mathbf{r}^{(0)}\| < 10^{-10}$

$\lambda_{\min}(\widetilde{M})$	64^2	128^2	256^2	512^2
no shift	9	10	10	10
10^{-1}	12	14	13	13
10^{-2}	24	34	26	28
10^{-3}	32	51	34	41
10^{-4}	35	57	38	47

Krylov Acceleration

If only one bad eigenvalue, then CG acceleration should be effective

PCG Iterations so that $\|\mathbf{r}^{(k)}\|/\|\mathbf{r}^{(0)}\| < 10^{-10}$

$\lambda_{\min}(\widetilde{M})$	64^2	128^2	256^2	512^2
no shift	9	10	10	10
10^{-1}	12	14	13	13
10^{-2}	24	34	26	28
10^{-3}	32	51	34	41
10^{-4}	35	57	38	47

May solve 1000+ RHS in each Monte-Carlo step
Can amortize expensive setup, if it pays off in solve phase

Calibrating Interpolation

AMG based on assumptions about slow-to-converge errors

What if we don't know what to assume?

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

AMG based on assumptions about slow-to-converge errors

What if we don't know what to assume?
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

Adapt AMG interpolation based on true performance of relaxation

---

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

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

Adaptive Algorithm

While $\|I - B_{\mathrm{MG}}^{-1}A\|_{\mathrm{est}}$ is large

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

Effect of Adaptivity

AMG Convergence Factors

$\lambda_{\min}(\widetilde{M})$	64^2	128^2	256^2	512^2
1.0	0.024	0.039	0.036	0.034
10^{-1}	0.197	0.311	0.328	0.294
10^{-2}	0.414	0.446	0.488	0.550
10^{-3}	0.587	0.527	0.542	0.630
10^{-4}	0.626	0.557	0.477	0.586

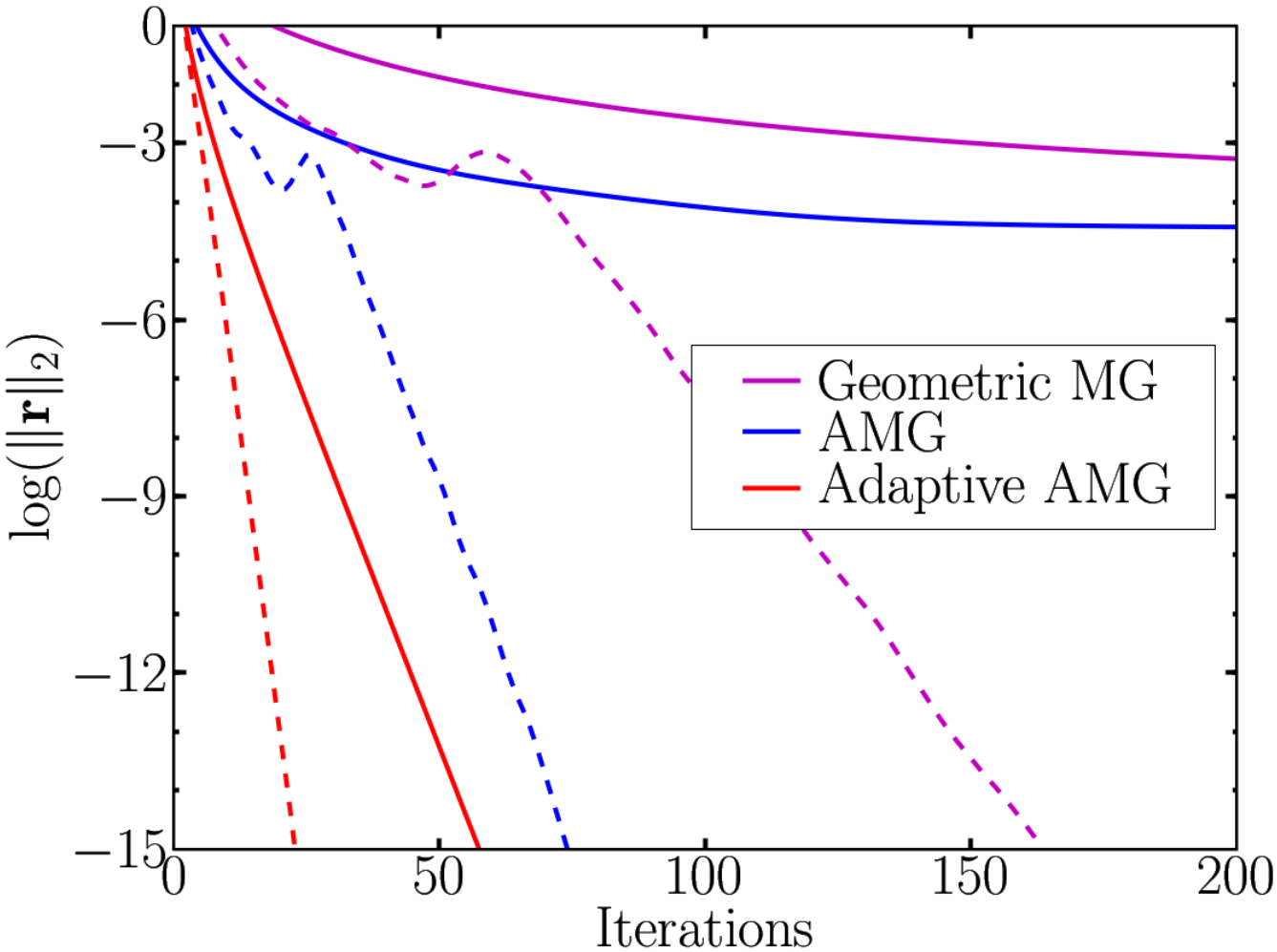
Effect of Adaptivity

AMG-PCG Iteration Counts

$\lambda_{\min}(\widetilde{M})$	64^2	128^2	256^2	512^2
no shift	6	7	7	7
10^{-1}	7	8	8	8
10^{-2}	10	11	11	12
10^{-3}	13	13	12	13
10^{-4}	14	14	11	13

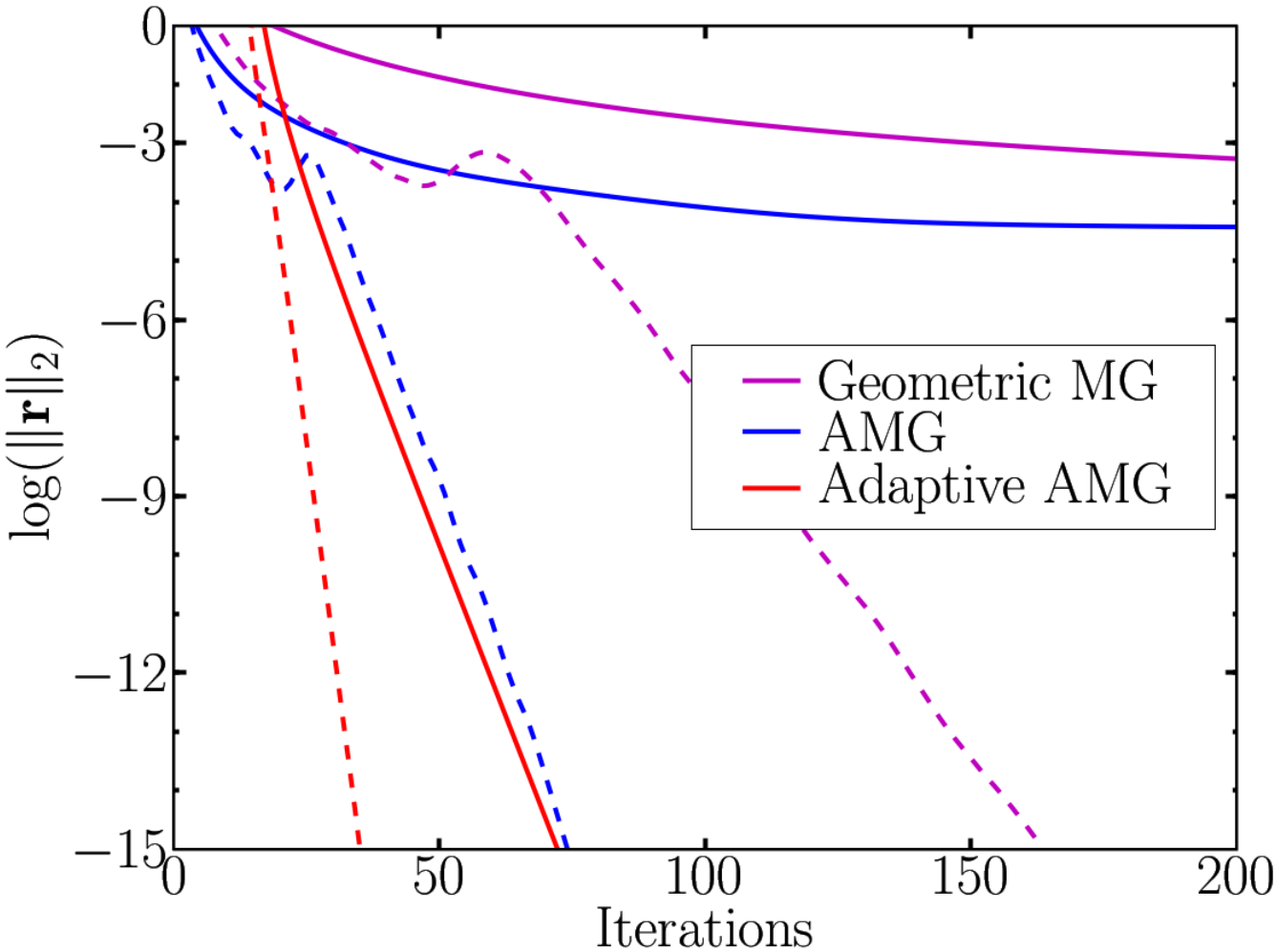
Cost of Adaptivity

Extra relaxations pay off with multiple RHS



Cost of Adaptivity

Extra relaxations pay off with multiple RHS



Optimising Adaptivity

Large parameter space to search

- Relaxation on $A\mathbf{x} = \mathbf{0}$ is cheap
- Constructing coarse grids is expense
- Ability of relaxation to improve prototype diminishes
- Coarse-grid correction significantly improves prototype

Relax until performance slows, then coarsen

Optimising Adaptivity

Large parameter space to search

- Relaxation on $A\mathbf{x} = \mathbf{0}$ is cheap
- Constructing coarse grids is expense
- Ability of relaxation to improve prototype diminishes
- Coarse-grid correction significantly improves prototype

Relax until performance slows, then coarsen

Difficult to optimise due to problem-dependent behaviour

Full Physics

Final challenge: attack full 4D system

$$\begin{pmatrix} -\frac{1}{2h}D^2 - mI_3 & 0 & \imath D_3 - D_4 & \imath D_1 - D_2 \\ 0 & -\frac{1}{2h}D^2 - mI_3 & \imath D_1 + D_2 & -\imath D_3 - D_4 \\ -\imath D_3 - D_4 & -\imath D_1 + D_2 & -\frac{1}{2h}D^2 - mI_3 & 0 \\ -\imath D_1 - D_2 & \imath D_3 - D_4 & 0 & -\frac{1}{2h}D^2 - mI_3 \end{pmatrix}$$

where $D_\mu = I_3\partial_\mu - iA_\mu$, $D^2 = \sum_\mu D_\mu^2$

Full Physics

Final challenge: attack full 4D system

$$\begin{pmatrix} -\frac{1}{2h}D^2 - mI_3 & 0 & \imath D_3 - D_4 & \imath D_1 - D_2 \\ 0 & -\frac{1}{2h}D^2 - mI_3 & \imath D_1 + D_2 & -\imath D_3 - D_4 \\ -\imath D_3 - D_4 & -\imath D_1 + D_2 & -\frac{1}{2h}D^2 - mI_3 & 0 \\ -\imath D_1 - D_2 & \imath D_3 - D_4 & 0 & -\frac{1}{2h}D^2 - mI_3 \end{pmatrix}$$

where $D_\mu = I_3\partial_\mu - iA_\mu$, $D^2 = \sum_\mu D_\mu^2$

Half step: 2D System:

$$\begin{pmatrix} -\frac{1}{2h}D^2 - mI_3 & \imath D_1 - D_2 \\ -\imath D_1 - D_2 & -\frac{1}{2h}D^2 - mI_3 \end{pmatrix}$$

Brute Force Approach

- Adaptive smoothed aggregation multigrid
- Solve equivalent real form of $\widetilde{M}^{\star}\widetilde{M}$
- 128×128 periodic lattice

Average residual reduction per iteration

	Diagonal-PCG				AdaptiveMG-PCG			
$\lambda_{\min}(\widetilde{M})$	0.3	0.1	0.05	0.01	0.3	0.1	0.05	0.01
$\beta \approx 0.2$	0.83	0.92	0.96	0.99	0.28	0.29	0.31	0.31
$\beta \approx 0.3$	0.86	0.93	0.97	0.98	0.31	0.40	0.42	0.42
$\beta \approx 0.5$	0.85	0.94	0.96	0.99	0.31	0.31	0.31	0.33

J. Brannick et al., to appear in *Proc. DD16*, 2007

Brute Force Approach

- Adaptive smoothed aggregation multigrid
- Solve equivalent real form of $\widetilde{M}^{\star}\widetilde{M}$
- 128×128 periodic lattice

Average residual reduction per iteration

	Diagonal-PCG				AdaptiveMG-PCG			
$\lambda_{\min}(\widetilde{M})$	0.3	0.1	0.05	0.01	0.3	0.1	0.05	0.01
$\beta \approx 0.2$	0.83	0.92	0.96	0.99	0.28	0.29	0.31	0.31
$\beta \approx 0.3$	0.86	0.93	0.97	0.98	0.31	0.40	0.42	0.42
$\beta \approx 0.5$	0.85	0.94	0.96	0.99	0.31	0.31	0.31	0.33

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

J. Brannick et al., to appear in *Proc. DD16*, 2007

Relaxation on $MM^\star$

Equivalent real form of $M^\star M$ expensive:

- Real form doubles needed storage
- Real form hides complex structure
- $M^\star M$ (or $MM^\star$) much denser than M

Advantage: Structure of $MM^\star$ more amenable to multigrid

$$(MM^\star)_{jj} = -D^2 + \left(-\frac{1}{2h}D^2 - mI_3\right)^2$$

$$(MM^\star)_{jk} \longrightarrow \text{lower-order terms for } j \neq k$$

Hybrid approach

Implictly relax on $MM^\star$, but coarsen based on M

- Kaczmarz relaxation on M
- No need to compute or store $MM^\star$
- Slow to converge components dominated by D^2
- Use adaptive AMG to design effective coarse-grid correction

Summary

- QCD attempts to explain strong force
- Numerical simulation requires many matrix solves
- Matrices are heterogeneous, but structured
- MG/AMG naturally handles some challenges
- Adaptive AMG key to addressing mass shift
- Key Question: most efficient setup approach
- Generalization to full physics underway