

A Comparison of Two-Level Preconditioning Strategies

Scott MacLachlan
scott.maclachlan@tufts.edu

Tufts University

June 3, 2008

With Thanks to ...

This research was performed in collaboration with

- Jok Tang, Kees Vuik, TU-Delft
- Reinhard Nabben, TU-Berlin

This research was supported by

- the BSIK/BRICKS project,
- the Deutsche Forschungsgemeinschaft (DFG), Project NA248/2-2, and
- the European Community's Sixth Framework Programme, through a Marie Curie Incoming International Fellowship, MIF1-CT-2006-021927.

The Big Question

How do we solve $Ax = b$?

The Big Question

How do we solve $Ax = b$?

Gaussian Elimination

- Recursive procedure to reduce A to a simpler form
- Use i^{th} equation to eliminate x_i from j^{th} equation for $j > i$

The Bigger Question

Why do we solve $Ax = b$?

Discrete PDEs

Given PDE $L(u) = f$ on domain Ω .

Finite Differences

- Represent u and f by pointwise values at nodes $\xi_i \in \Omega$
- Approximate $L(u(\xi_i))$ using Taylor series

Finite Elements

- Represent u and f by basis functions for domain and range of L
- Approximate $L(u)$ by projection onto subsets of these bases

The Real Question

How do we solve $Ax = b$?

When

- A is large (of dimension $n > 10^6$)?
- A is sparse (fixed number of nonzeros per row)?
- A inherits properties from continuum PDE?

The Real Question

How do we solve $Ax = b$?

When

- A is large (of dimension $n > 10^6$)?
- A is sparse (fixed number of nonzeros per row)?
- A inherits properties from continuum PDE?
- we only need an approximate solution?

The Real Question

How do we solve $Ax = b$?

When

- A is large (of dimension $n > 10^6$)?
- A is sparse (fixed number of nonzeros per row)?
- A inherits properties from continuum PDE?
- we only need an approximate solution?

Want **specialized techniques** that take advantage of matrix structure and are **faster than Gaussian elimination**

Outline

- Polynomial Methods
 - ▶ Preconditioning
 - ▶ Spectral equivalence
 - ▶ Classical preconditioners
- Two-level preconditioners
 - ▶ Projections
 - ▶ Deflation
 - ▶ Balancing Neumann-Neumann
 - ▶ Multigrid
- Some comparisons
 - ▶ Abstract framework
 - ▶ Comparison theorems
 - ▶ What does it mean?

Preconditioning

The solution of $A\mathbf{x} = \mathbf{b}$ solves many other linear systems

Consider solving $MA\mathbf{x} = M\mathbf{b}$

Idea: If MA has nicer properties than A , easier to solve

Preconditioning

The solution of $A\mathbf{x} = \mathbf{b}$ solves many other linear systems

Consider solving $MA\mathbf{x} = M\mathbf{b}$

Idea: If MA has nicer properties than A , easier to solve

Need: Preconditioner, M , close to A^{-1}

Need: Cheap computation of $M\mathbf{r}$

Iterative Methods

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

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

- Take $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \omega_0 \mathbf{r}^{(0)}$

Iterative Methods

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

$$M\mathbf{r}^{(0)} = M\mathbf{b} - MA\mathbf{x}^{(0)} = MA\mathbf{x} - MA\mathbf{x}^{(0)} = MA(\mathbf{x} - \mathbf{x}^{(0)})$$

- Take $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \omega_0 M\mathbf{r}^{(0)}$

Error propagation form: $\mathbf{e}^{(1)} = (I - \omega_0 MA)\mathbf{e}^{(0)}$

Iterative Methods

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

$$M\mathbf{r}^{(0)} = M\mathbf{b} - MA\mathbf{x}^{(0)} = MA\mathbf{x} - MA\mathbf{x}^{(0)} = MA(\mathbf{x} - \mathbf{x}^{(0)})$$

- Take $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \omega_0 M\mathbf{r}^{(0)}$

Error propagation form: $\mathbf{e}^{(1)} = (I - \omega_0 MA)\mathbf{e}^{(0)}$
 $\mathbf{e}^{(2)} = (I - \omega_1 MA)\mathbf{e}^{(1)}$

Iterative Methods

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

$$M\mathbf{r}^{(0)} = M\mathbf{b} - M\mathbf{A}\mathbf{x}^{(0)} = M\mathbf{A}\mathbf{x} - M\mathbf{A}\mathbf{x}^{(0)} = M\mathbf{A}(\mathbf{x} - \mathbf{x}^{(0)})$$

- Take $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \omega_0 M\mathbf{r}^{(0)}$

Error propagation form: $\mathbf{e}^{(1)} = (I - \omega_0 MA)\mathbf{e}^{(0)}$
 $\mathbf{e}^{(2)} = (I - \omega_1 MA)(I - \omega_0 MA)\mathbf{e}^{(0)}$

Iterative Methods

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

$$M\mathbf{r}^{(0)} = M\mathbf{b} - MA\mathbf{x}^{(0)} = MA\mathbf{x} - MA\mathbf{x}^{(0)} = MA(\mathbf{x} - \mathbf{x}^{(0)})$$

- Take $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \omega_0 M\mathbf{r}^{(0)}$

Error propagation form:

$$\begin{aligned}\mathbf{e}^{(1)} &= (I - \omega_0 MA)\mathbf{e}^{(0)} \\ \mathbf{e}^{(2)} &= (I - \omega_1 MA)(I - \omega_0 MA)\mathbf{e}^{(0)} \\ &\vdots \\ \mathbf{e}^{(k)} &= p_{k,0}(MA)\mathbf{e}^{(0)}\end{aligned}$$

Question: How do we pick coefficients: $\omega_0, \omega_1, \dots$

Polynomial Methods

Many ways to choose coefficients of $p_{k,0}$ (or weights, ω_i)

Uniform

- Fix $\omega_i = \omega$

Chebyshev

- Use estimate of field of values of MA to choose weights

Krylov

- Choose polynomials for optimality in some sense
- Conjugate gradient, GMRES, BiCGSTAB, . . .

Success depends on **closeness** of M and A

Spectral Equivalence

Measure closeness of M and A by bound

$$\lambda_{\min} \mathbf{x}^T A^{-1} \mathbf{x} \leq \mathbf{x}^T M \mathbf{x} \leq \lambda_{\max} \mathbf{x}^T A^{-1} \mathbf{x}, \quad \forall \mathbf{x}$$

λ_{\min} and λ_{\max} are called the **spectral equivalence bounds**

Convergence of polynomial methods typically depends on

$$\kappa(A^{\frac{1}{2}} M A^{\frac{1}{2}}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

For example,

$$\|\mathbf{e}^{(k)}\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|\mathbf{e}^{(0)}\|_A$$

Spectral Equivalence

Measure closeness of M and A by bound

$$\lambda_{\min} \mathbf{x}^T A^{-1} \mathbf{x} \leq \mathbf{x}^T M \mathbf{x} \leq \lambda_{\max} \mathbf{x}^T A^{-1} \mathbf{x}, \quad \forall \mathbf{x}$$

λ_{\min} and λ_{\max} are called the **spectral equivalence bounds**

Convergence of polynomial methods typically depends on

$$\kappa(A^{\frac{1}{2}} M A^{\frac{1}{2}}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

How do we pick an M so that $\kappa(A^{\frac{1}{2}} M A^{\frac{1}{2}})$ is small?

Spectral Equivalence

Measure closeness of M and A by bound

$$\lambda_{\min} \mathbf{x}^T A^{-1} \mathbf{x} \leq \mathbf{x}^T M \mathbf{x} \leq \lambda_{\max} \mathbf{x}^T A^{-1} \mathbf{x}, \quad \forall \mathbf{x}$$

λ_{\min} and λ_{\max} are called the **spectral equivalence bounds**

Convergence of polynomial methods typically depends on

$$\kappa(A^{\frac{1}{2}} M A^{\frac{1}{2}}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

How do we pick an M so that $\kappa(A^{\frac{1}{2}} M A^{\frac{1}{2}})$ is small?

Start from A

Classical Preconditioners

Classical choices for M include

- $M^{-1} = \alpha I$, for $\alpha \approx \|A\|$ (Richardson)
- $M^{-1} = \alpha \text{diag}(A)$ (Jacobi)
- $M^{-1} = \text{tril}(A)$ (Gauss-Seidel)
- $M^{-1} = \hat{L}\hat{U}$, for $A = LU$, $\hat{L} \approx L$, $\hat{U} \approx U$ (ILU)

Question: How well do these work?

Classical Preconditioners

Classical choices for M include

- $M^{-1} = \alpha I$, for $\alpha \approx \|A\|$ (Richardson)
- $M^{-1} = \alpha \text{diag}(A)$ (Jacobi)
- $M^{-1} = \text{tril}(A)$ (Gauss-Seidel)
- $M^{-1} = \hat{L}\hat{U}$, for $A = LU$, $\hat{L} \approx L$, $\hat{U} \approx U$ (ILU)

Question: How well do these work?

Define **well** by considering $\sigma \left(A^{\frac{1}{2}} M A^{\frac{1}{2}} \right)$

- Lower and upper bounds on $\sigma \left(A^{\frac{1}{2}} M A^{\frac{1}{2}} \right)$ define condition number: $\kappa \left(A^{\frac{1}{2}} M A^{\frac{1}{2}} \right) = \frac{\lambda_{\max}}{\lambda_{\min}}$

Simple Test Problem

Finite difference discretization of $-\Delta u = f$ in $[0, 1]^2$

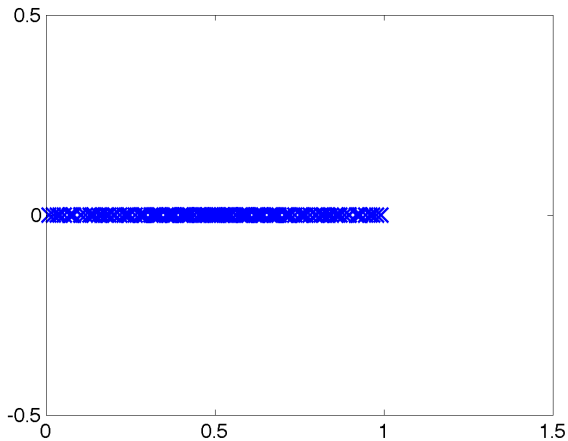
Approximate $-u_{xx} \approx \frac{1}{h^2} (-u_{i-1,j} + 2u_{i,j} - u_{i+1,j}) + O(h^2)$

$$\frac{1}{h^2} (-u_{i-1,j} + 2u_{i,j} - u_{i+1,j}) + \frac{1}{h^2} (-u_{i,j-1} + 2u_{i,j} - u_{i,j+1}) = f_{i,j}$$

for $1 \leq i \leq N$, $1 \leq j \leq N$, along with Dirichlet boundary conditions, $u_{i,j} = 0$, for $i = 0, N + 1$, $j = 0, N + 1$

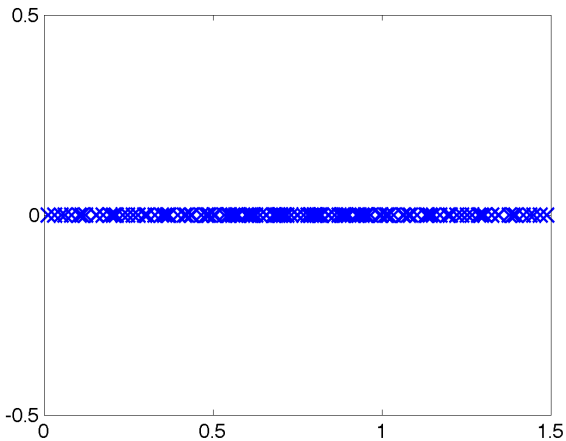
Matrix of dimension $n = N \times N$, with 5 nonzero entries per row

Spectral Pictures



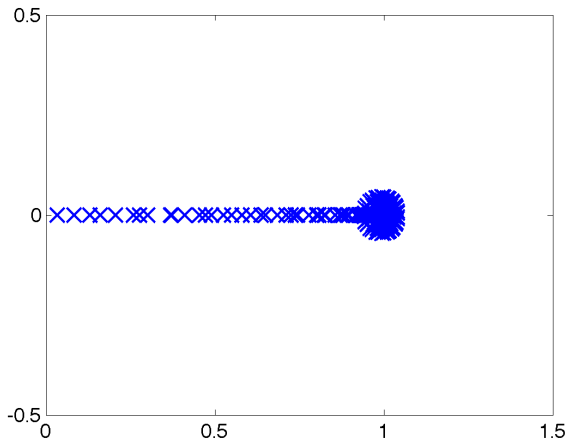
Richardson: $M^{-1} = \|A\|$

Spectral Pictures



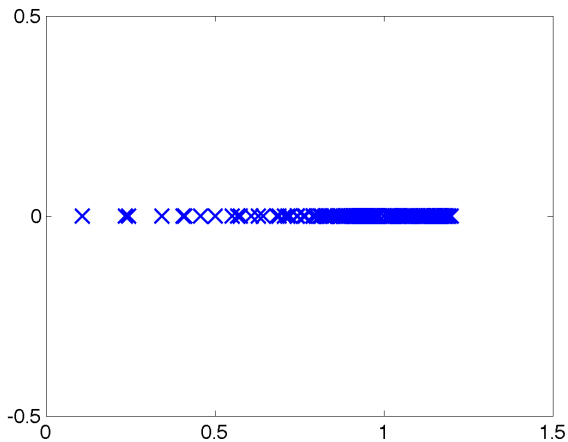
$$\text{Jacobi: } M^{-1} = \frac{3}{4} \text{diag}(A)$$

Spectral Pictures



Gauss-Seidel: $M^{-1} = \text{tril}(A)$

Spectral Pictures



$$\text{ILU: } M^{-1} = \hat{L}\hat{U}$$

The Problem

Convergence expected to slow as $\kappa(A^{\frac{1}{2}}MA^{\frac{1}{2}}) = \frac{\lambda_{\max}}{\lambda_{\min}}$ grows

N	κ_{JAC}	κ_{ILU}
16	116.5	9.5
32	440.7	37.3
64	1711.7	146.4
128	6743.7	603.5

We expect performance to degrade as N increases!

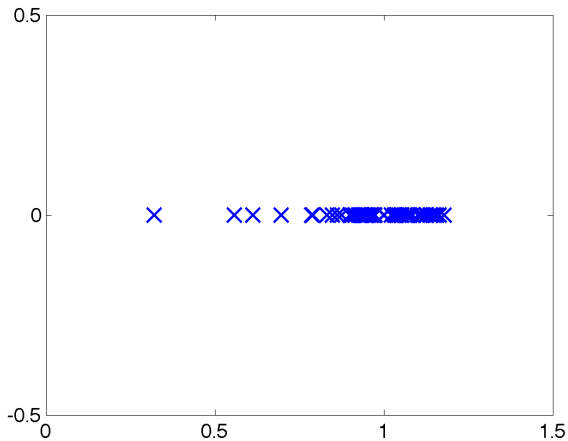
Optimal Performance

Because we're interested in solving large problems quickly, want

- $\kappa(A^{\frac{1}{2}}MA^{\frac{1}{2}})$ bounded, independently of N, n
- cost of computing MA to be proportional to $n = N^2$

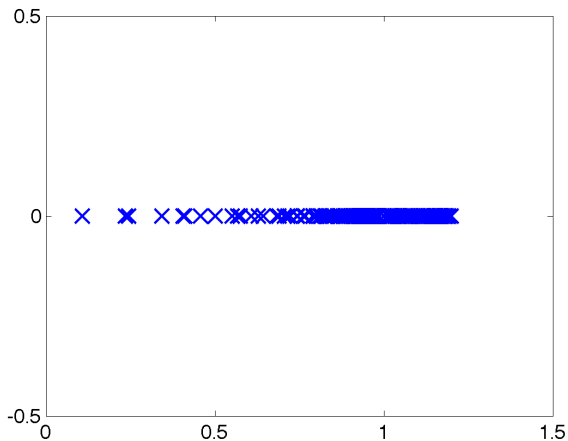
Question: Can we modify M to get optimal performance?

Spectral Picture Revisited



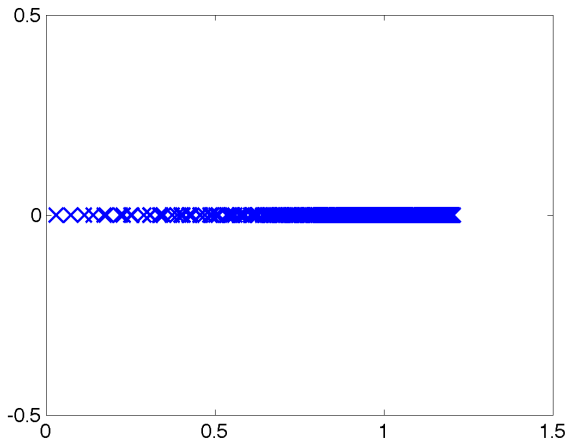
$$\text{ILU: } M^{-1} = \hat{L}\hat{U}, N = 8$$

Spectral Picture Revisited



$$\text{ILU: } M^{-1} = \hat{L}\hat{U}, N = 16$$

Spectral Picture Revisited



$$\text{ILU: } M^{-1} = \hat{L}\hat{U}, N = 32$$

Projection Methods

Only a few modes cause difficulty!

Try and remove these modes by another process. Let

- $Z \in \mathbb{R}^{n \times k}$ have full rank
- $W \in \mathbb{R}^{n \times (n-k)}$ have full rank
- $Z^T A W = 0$

Projection Methods

Only a few modes cause difficulty!

Try and remove these modes by another process. Let

- $Z \in \mathbb{R}^{n \times k}$ have full rank
- $W \in \mathbb{R}^{n \times (n-k)}$ have full rank
- $Z^T A W = 0$
- $\mathcal{R}(Z)$ contain troublesome modes

Write $\mathbf{e}^{(k)} = Z\mathbf{v} + W\mathbf{y}$

$$P\mathbf{e}^{(k)} = (I - Z(Z^T A Z)^{-1} Z^T A)\mathbf{e}^{(k)} = W\mathbf{y}$$

$$P^T \mathbf{r}^{(k)} = (I - A Z (Z^T A Z)^{-1} Z^T) A \mathbf{e}^{(k)} = A W \mathbf{y}$$

Two-Level Preconditioners

Preconditioners that combine M and P are called **two-level preconditioners**

Complementarity:

- M removes most errors
- P removes errors for which M is ineffective

Two-Level Preconditioners

Preconditioners that combine M and P are called **two-level preconditioners**

Complementarity:

- M removes most errors
- P removes errors for which M is ineffective

$$P = I - Z(Z^T A Z)^{-1} Z^T A$$

Applying P requires solving system with $Z^T A Z$

- $Z^T A Z$ is called the “coarse-grid” system

Two Approaches

1. Use P to remove error in $\mathbf{x}^{(k)}$ in $\text{span}(Z)$:

$$\begin{aligned}\mathbf{x}^{(k')} &= \mathbf{x}^{(k)} + Z(Z^T AZ)^{-1} Z^T (\mathbf{b} - A\mathbf{x}^{(k)}) \\ &= \mathbf{x}^{(k)} + Z(Z^T AZ)^{-1} Z^T \mathbf{r}^{(k)} \\ \mathbf{e}^{(k')} &= P\mathbf{e}^{(k)} = \mathbf{e}^{(k)} - Z(Z^T AZ)^{-1} Z^T A\mathbf{e}^{(k)}\end{aligned}$$

2. Use P to partition exact solution

$$\mathbf{x} = P\mathbf{x} + (I - P)\mathbf{x}$$

Since $(I - P)\mathbf{x} = Z(Z^T AZ)^{-1} Z^T A\mathbf{x} = Z(Z^T AZ)^{-1} Z^T \mathbf{b}$,
concentrate on computing $P\mathbf{x}$, but

$$AP\mathbf{x} = P^T A\mathbf{x} = P^T \mathbf{b}.$$

Deflation Techniques

Second approach: Apply P^T directly to residual

Write

$$\mathbf{r}^{(k)} = A\mathbf{e}^{(k)} = AZ\mathbf{v} + AW\mathbf{y},$$

so that

$$P^T\mathbf{r}^{(k)} = P^T A\mathbf{e}^{(k)} = AW\mathbf{y},$$

then precondition:

$$MP^T\mathbf{r}^{(k)} = MP^T A\mathbf{e}^{(k)} = MAW\mathbf{y}$$

Idea: exactly eliminate errors in $\mathcal{R}(Z)$, by computing $(I - P)\mathbf{x}$, then iterate for solution in $\mathcal{R}(W)$.

Deflation Preconditioner

Can write deflation algorithm as classical preconditioner

Solving for $\tilde{\mathbf{x}} = P\mathbf{x}$, in

$$MP^T A \tilde{\mathbf{x}} = MP^T \mathbf{b}$$

This is in the same form as classical preconditioning:

$$M\mathbf{A}\mathbf{x} = M\mathbf{b}$$

Identify deflation preconditioner as

$$\mathcal{P}_{\text{DEF}} = MP^T$$

Balancing Preconditioners

First Approach: Project out errors in $\mathcal{R}(Z)$ before and after applying M

1. $\mathbf{x}^{(k')} = \mathbf{x}^{(k)} + Z(Z^T A Z)^{-1} Z^T \mathbf{r}^{(k)}$
2. $\mathbf{x}^{(k'')} = \mathbf{x}^{(k')} + \omega_k M \mathbf{r}^{(k')}$
3. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k'')} + Z(Z^T A Z)^{-1} Z^T \mathbf{r}^{(k'')}$

Equivalent to solving preconditioned system

$$\mathcal{P}_{\text{BNN}} A \mathbf{x} = \mathcal{P}_{\text{BNN}} \mathbf{b},$$

for

$$\mathcal{P}_{\text{BNN}} = P M P^T + Z(Z^T A Z)^{-1} Z^T$$

Multigrid Preconditioners

First Approach: Apply M before and after projecting out errors in $\mathcal{R}(Z)$

1. $\mathbf{x}^{(k')} = \mathbf{x}^{(k)} + M\mathbf{r}^{(k)}$
2. $\mathbf{x}^{(k'')} = \mathbf{x}^{(k')} + Z(Z^T A Z)^{-1} Z^T \mathbf{r}^{(k')}$
3. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k'')} + M^T \mathbf{r}^{(k'')}$

Equivalent to solving preconditioned system

$$\mathcal{P}_{\text{MG}} A \mathbf{x} = \mathcal{P}_{\text{MG}} \mathbf{b},$$

for

$$\mathcal{P}_{\text{MG}} = M^T P^T + P M + M^T P^T A M + Z(Z^T A Z)^{-1} Z^T$$

Comparing Costs

Three very different implementations of same idea

Compare costs per iteration:

Deflation: $\mathcal{P}_{\text{DEF}} = MP^T$

- One multiplication with M
- One multiplication with P^T

Balancing: $\mathcal{P}_{\text{BNN}} = PMP^T + Z(Z^T AZ)^{-1}Z^T$

- One multiplication with M
- Two multiplications with P

Multigrid: $\mathcal{P}_{\text{MG}} = M^T P^T + PM + M^T P^T AM + \dots$

- One multiplication with M , one with M^T
- One multiplication with P

Comparing Performance

Comparing performance is more difficult than comparing costs

Actual Performance depends on

- Polynomial method chosen
- Right-hand side, \mathbf{b}
- Initial guess, $\mathbf{x}^{(0)}$
- Choice of M
- Choice of Z

Instead: Compare expected performance based on bounds

$$\|\mathbf{e}^{(k)}\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|\mathbf{e}^{(0)}\|_A$$

Comparing Bounds

Important factor in

$$\|\mathbf{e}^{(k)}\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|\mathbf{e}^{(0)}\|_A$$

is $\kappa(A^{\frac{1}{2}}MA^{\frac{1}{2}}) = \frac{\lambda_{\max}}{\lambda_{\min}}$.

Goal: Theoretical comparison of spectra of DEF, BNN, MG

Comparing DEF and BNN

For any given choice of M and Z ,

$$\sigma(\mathcal{P}_{\text{DEF}}A) = \{0, \dots, 0, \mu_{k+1}, \dots, \mu_n\}$$

Comparing DEF and BNN

For any given choice of M and Z ,

$$\sigma(\mathcal{P}_{\text{DEF}}A) = \{0, \dots, 0, \mu_{k+1}, \dots, \mu_n\}$$

Furthermore, if the same choices of M and Z are used in BNN,

$$\sigma(\mathcal{P}_{\text{BNN}}A) = \{1, \dots, 1, \mu_{k+1}, \dots, \mu_n\}$$

Comparing DEF and BNN

For any given choice of M and Z ,

$$\sigma(\mathcal{P}_{\text{DEF}}A) = \{0, \dots, 0, \mu_{k+1}, \dots, \mu_n\}$$

Furthermore, if the same choices of M and Z are used in BNN,

$$\sigma(\mathcal{P}_{\text{BNN}}A) = \{1, \dots, 1, \mu_{k+1}, \dots, \mu_n\}$$

So, if $\mu_{k+1} \leq 1 \leq \mu_n$, then $\kappa_{\text{DEF}} = \kappa_{\text{BNN}}$

Comparing DEF with MG

Direct comparison of DEF and MG is difficult

- MG requires multiplication by both M and M^T
- Preserving symmetry of operations is important
 - ▶ Difficult to just drop one of multiplications

Can show that multigrid with only M or M^T gives

$$\sigma(\mathcal{P}_{\text{MG1}}A) = \{1, \dots, 1, \mu_{k+1}, \dots, \mu_n\},$$

but numerical stability becomes a problem

Spectral Two-Level Preconditioners

Take

- M to be arbitrary
- $\{\mathbf{v}_j\}, \{\lambda_j\}$ to be the eigenvectors & eigenvalues of MA
- $Z = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$
- $0 < \lambda_{k+1} \leq \lambda_j \leq \lambda_n < 2$ for all $k + 1 < j < n$

Then, $\mathcal{P}_{\text{DEF}}A$ has eigenvalues

$$\begin{cases} 0 & \text{for } j = 1, \dots, k \\ \lambda_j & \text{for } j = k + 1, \dots, n \end{cases}$$

and $\mathcal{P}_{\text{MG}}A$ has eigenvalues

$$\begin{cases} 1 & \text{for } j = 1, \dots, k \\ \lambda_j(2 - \lambda_j) & \text{for } j = k + 1, \dots, n \end{cases}$$

C. Vuik, A. Segal, & J.A. Meijerink, J. Comput. Phys. 1999, **152**:385–403

B. Carpentieri, L. Giraud, & S. Gratton, SISC 2007, **29**:1593-1612

Spectral Two-Level Preconditioners

Take

- M to be arbitrary
- $\{\mathbf{v}_j\}, \{\lambda_j\}$ to be the eigenvectors & eigenvalues of MA
- $Z = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$
- $0 < \lambda_{k+1} \leq \lambda_j \leq \lambda_n < 2$ for all $k + 1 < j < n$

Then, $\mathcal{P}_{\text{DEF}}A$ has condition number

$$\kappa_{\text{DEF}} = \frac{\lambda_n}{\lambda_{k+1}}$$

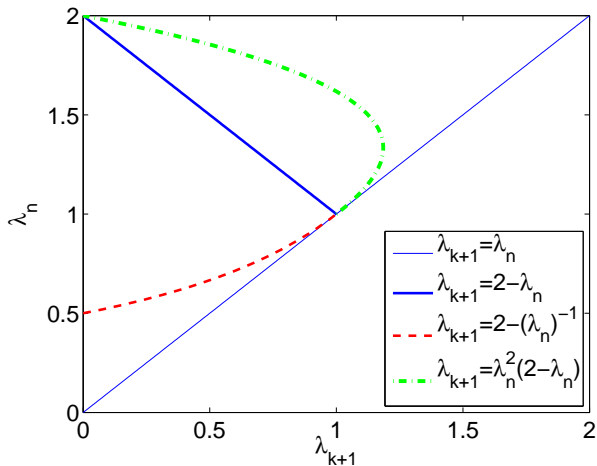
and $\mathcal{P}_{\text{MG}}A$ has condition number

$$\kappa_{\text{MG}} = \frac{1}{\min\{\lambda_{k+1}(2 - \lambda_{k+1}), \lambda_n(2 - \lambda_n)\}}$$

C. Vuik, A. Segal, & J.A. Meijerink, J. Comput. Phys. 1999, **152**:385–403

B. Carpentieri, L. Giraud, & S. Gratton, SISC 2007, **29**:1593-1612

Comparing Spectral Preconditioners



Relaxation Parameters

Restriction that $0 < \lambda_{k+1} \leq \lambda_j \leq \lambda_n < 2$ may not be natural.

Imposed by enforcing that \mathcal{P}_{MG} be symmetric and definite

- No such restriction for \mathcal{P}_{DEF}
- Satisfied for many typical choices of M
- Easily enforced by adding parameter, $M \leftarrow \alpha M$

Addition of **optimal relaxation parameter**, α , has significant effect on MG performance

For $M = I$, $\alpha = \frac{2}{\lambda_{k+1} + \lambda_n}$ is optimal choice. With this α ,

$$\kappa_{\text{MG}} = \frac{(\lambda_{k+1} + \lambda_n)^2}{4\lambda_{k+1}\lambda_n} \leq \frac{\lambda_n}{\lambda_{k+1}} = \kappa_{\text{DEF}}$$

Symmetrizing M

Main difficulty: MG has two classical preconditioning steps, DEF and BNN use just one

Idea: Use different M in MG and DEF/BNN

Recall: MG given by

1. $\mathbf{x}^{(k')} = \mathbf{x}^{(k)} + M\mathbf{r}^{(k)}$
2. $\mathbf{x}^{(k'')} = \mathbf{x}^{(k')} + Z(Z^T AZ)^{-1}Z^T \mathbf{r}^{(k')}$
3. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k'')} + M^T \mathbf{r}^{(k'')}$

BNN given by

1. $\mathbf{x}^{(k')} = \mathbf{x}^{(k)} + Z(Z^T AZ)^{-1}Z^T \mathbf{r}^{(k)}$
2. $\mathbf{x}^{(k'')} = \mathbf{x}^{(k')} + M\mathbf{r}^{(k')}$
3. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k'')} + Z(Z^T AZ)^{-1}Z^T \mathbf{r}^{(k'')}$

Symmetrizing M

Main difficulty: MG has two classical preconditioning steps, DEF and BNN use just one

Idea: Use different M in MG and DEF/BNN

Use \tilde{M} in DEF/BNN, where

$$(I - \tilde{M}A) = (I - M^T A)(I - MA)$$

Compute: $\tilde{M} = M^T + M - M^T A M$

Comparing MG, BNN, and DEF

Compare

- MG using M and arbitrary Z
- BNN using \tilde{M} and same Z
- DEF using \tilde{M} and same Z

Then,

$$\begin{aligned}\sigma(\mathcal{P}_{\text{MG}}A) &= \{1, \dots, 1, \lambda_{k+1}, \dots, \lambda_n\}, \\ \sigma(\mathcal{P}_{\text{BNN}}A) &= \{1, \dots, 1, \lambda_{k+1}, \dots, \lambda_n\}, \\ \sigma(\mathcal{P}_{\text{DEF}}A) &= \{0, \dots, 0, \lambda_{k+1}, \dots, \lambda_n\},\end{aligned}$$

where λ_{k+1}, λ_n depend on choices of M, Z .

Consequences

Convergence analysis says the choice of two-level approach doesn't matter

Instead, we should concentrate on:

- Choices of M , Z
- Efficient implementations
- Numerical stability

Stability

Analysis is based on exactly computing

$$\mathbf{x}^{(k')} = \mathbf{x}^{(k)} + Z(Z^T A Z)^{-1} Z^T \mathbf{r}^{(k)}$$

This requires solution of

$$(Z^T A Z) \mathbf{w} = Z^T \mathbf{r}^{(k)}$$

In practice, solve this system by recursion, perturbing eigenvalues of $\mathcal{P}A$

- For MG, BNN, perturbs unit eigenvalues
- For DEF, perturbs zero eigenvalues

Small non-zero eigenvalues can cause convergence problems

Model Problems: Bubbly Flows

Consider two-phase bubbly fluid flow

- Navier-Stokes equations, discontinuous ρ
- Operator-splitting approach
- Pressure-correction solves

Solving $\nabla \cdot \frac{1}{\rho} \nabla p = f$ on cell-centred mesh.

Choices of M and Z

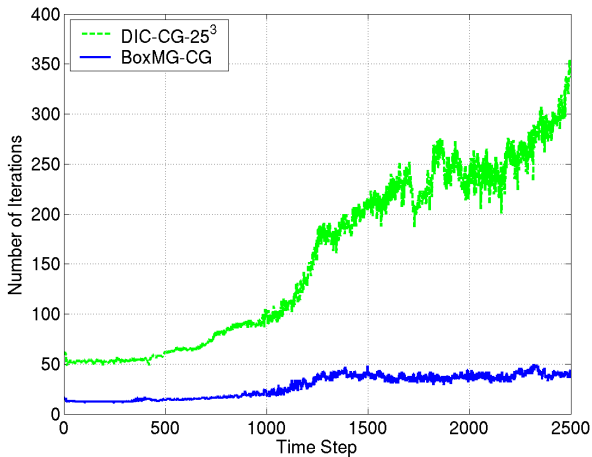
Compare

- Incomplete Cholesky with subdomain deflation
- Gauss-Seidel with trilinear multigrid interpolation
- Gauss-Seidel with operator-induced multigrid interpolation

Iteration counts for $n = 64^3$ lattice with 8 bubbles

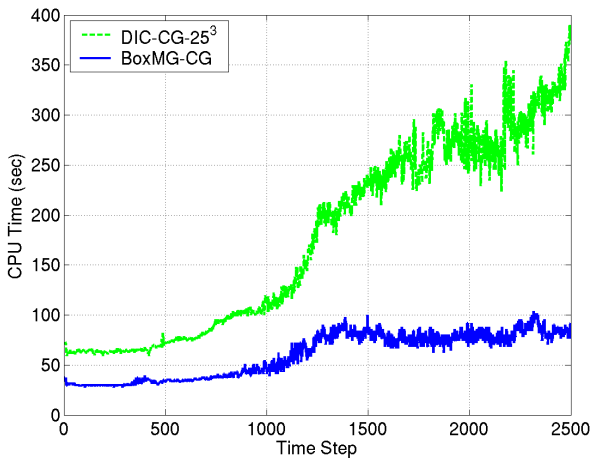
Contrast	10^{-1}	10^{-3}	10^{-5}
IC-subdomain	35	54	59
GS-trilinear	10	18	37
GS-operator	12	12	12

Time-Dependent Simulation



200³ spatial grid; two rising air bubbles in water

Time-Dependent Simulation



200³ spatial grid; two rising air bubbles in water

Conclusions

- Preconditioned polynomial methods best option for solving many linear systems
- Convergence depends on $\kappa \left(A^{\frac{1}{2}} M A^{\frac{1}{2}} \right)$
- Improve performance by incorporating projection, P
- Three variants: Deflation, Balancing, Multigrid
- Parameters important
- For “right” choice of M , performance bounds are identical
- Deflation may have stability issues
- **Important choices** are M and Z