Greedy strategies for multilevel partitioning

Scott MacLachlan   Yousef Saad

Department of Computer Science and Engineering,
University of Minnesota
{maclach,saad}@cs.umn.edu

Sparse Days at CERFACS
June 16, 2006

Preconditioning

Goal: solve $A\mathbf{x} = \mathbf{b}$ as efficiently as possible

Difficulty grows with discrete problem size, n

- Condition Number
- Cost of LU factorization
- Total cost of Jacobi iteration
- Total cost of Krylov subspace iteration

Preconditioning

Goal: solve $A\mathbf{x} = \mathbf{b}$ as efficiently as possible

Difficulty grows with discrete problem size, n

- Condition Number
- Cost of LU factorization
- Total cost of Jacobi iteration
- Total cost of Krylov subspace iteration

Idea: Transform A to control total costs

$$A\mathbf{x} = \mathbf{b} \qquad \longrightarrow \qquad B^{-1}A\mathbf{x} = B^{-1}\mathbf{b}$$

- Number of Krylov iterations independent of n
- Cost of computing $B^{-1}\mathbf{r}$ grows like $nnz(A)$

Algebraic Preconditioners

If we know where A came from, we have a good chance to define an effective B

What if we don't?

- continuum problem or application
- discretization procedure
- problem already transformed (unsuccessfully)
- variability in coefficients
- new application

Algebraic preconditioners define B based only on A

Multilevel Preconditioners

Some types of error may be easily eliminated

- A may have small independent sets of variables
- May know some part of solution
- Richardson iteration, $I - \frac{\sigma}{\|A\|}A$, effectively eliminates eigenvectors with large eigenvalues

Multilevel Preconditioners

Some types of error may be easily eliminated

- A may have small independent sets of variables
- May know some part of solution
- Richardson iteration, $I - \frac{\sigma}{\|A\|}A$, effectively eliminates eigenvectors with large eigenvalues

Idea: Treat errors that are not easily eliminated separately

- easy to resolve $\longrightarrow$ fine subspace
  - treat with A as given
- hard to resolve $\longrightarrow$ coarse subspace
  - restrict A to this subspace and resolve there

Errors and Residuals

(Preconditioned) Residual is only indicator of error

Arnoldi recombines iterates with their residuals:

- $\mathbf{v}_{j+1} = \alpha \left(A\mathbf{v}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i \right)$

Krylov space grows in direction of residual:

- $\text{span}\{\mathbf{v}_j\} = \text{span}\{A^j\mathbf{v}_0\}$

At early stages, error dominated by components with small relative residuals

Hard to reduce $\mathbf{e}$ when $\mathbf{r} = A\mathbf{e}$ is small compared to $\mathbf{e}$

Harmonic Extensions

$A\mathbf{e}$ can be smaller than e if $(A\mathbf{e})_i = 0$ when $e_i \neq 0$ for many i.

Partition A and $\mathbf{e}$:

$$A\mathbf{e} = \begin{bmatrix} A_{ff} & -A_{fc} \\ -A_{cf} & A_{cc} \end{bmatrix} \begin{pmatrix} \mathbf{e}_f \\ \mathbf{e}_c \end{pmatrix}$$

Harmonic Extensions

$A\mathbf{e}$ can be smaller than e if $(A\mathbf{e})_i = 0$ when $e_i \neq 0$ for many i.

Partition A and $\mathbf{e}$:

$$A\mathbf{e} = \begin{bmatrix} A_{ff} & -A_{fc} \\ -A_{cf} & A_{cc} \end{bmatrix} \begin{pmatrix} \mathbf{e}_f \\ \mathbf{e}_c \end{pmatrix}$$

If

$$\mathbf{e} = \begin{bmatrix} A_{ff}^{-1} A_{fc} \\ I \end{bmatrix} \mathbf{e}_c,$$

then $\mathbf{e}$ is called the harmonic extension of $\mathbf{e}_c$

Block Factorization

Partition

$$A\mathbf{x} = \begin{bmatrix} A_{ff} & -A_{fc} \\ -A_{cf} & A_{cc} \end{bmatrix} \begin{pmatrix} \mathbf{x}_f \\ \mathbf{x}_c \end{pmatrix} = \begin{pmatrix} \mathbf{b}_f \\ \mathbf{b}_c \end{pmatrix} = \mathbf{b},$$

then block factor,

$$A = \begin{bmatrix} I & 0 \\ -A_{cf}A_{ff}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{ff} & 0 \\ 0 & \hat{A}_{cc} \end{bmatrix} \begin{bmatrix} I & -A_{ff}^{-1}A_{fc} \\ 0 & I \end{bmatrix},$$

where

$$\hat{A}_{cc} = \begin{bmatrix} A_{ff}^{-1}A_{fc} \\ I \end{bmatrix}^T A \begin{bmatrix} A_{ff}^{-1}A_{fc} \\ I \end{bmatrix} = A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}.$$

Approximate Block Factorizations

Block factorization,

$$A = \begin{bmatrix} I & 0 \\ -A_{cf}A_{ff}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{ff} & 0 \\ 0 & \hat{A}_{cc} \end{bmatrix} \begin{bmatrix} I & -A_{ff}^{-1}A_{fc} \\ 0 & I \end{bmatrix},$$

suggests a preconditioning strategy. If

- $A_{ff}^{-1}\mathbf{y}_f$ is easily approximated
- $\hat{A}_{cc}\mathbf{x}_c = \mathbf{y}_c$ is easily (approximately) solved

then A^{-1} is easily approximated.

Many preconditioners are based on this principle

AMLI, additive multigrid, approximate cyclic reduction, ILUM, ARMS, ...

Algebraic Recursive Multilevel Solver

Approximate A_{ff} by its ILUT factors, $A_{ff} \approx LU$.
Preconditioner is

$$B = \begin{bmatrix} I & 0 \\ -A_{cf} U^{-1} L^{-1} & I \end{bmatrix} \begin{bmatrix} LU & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & -U^{-1} L^{-1} A_{fc} \\ 0 & I \end{bmatrix},$$

where $S \approx A_{cc} - A_{cf} U^{-1} L^{-1} A_{fc}$.
Coarse-grid problems

- computed using techniques akin to ILUT
- solved recursively

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

ARMS Analysis

Let

- A be symmetric and positive definite
- $B = \begin{bmatrix} I & 0 \\ -A_{cf}D_{ff}^{-1} & I \end{bmatrix} \begin{bmatrix} D_{ff} & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & -D_{ff}^{-1}A_{fc} \\ 0 & I \end{bmatrix}$
- $\begin{bmatrix} D_{ff} & -A_{fc} \\ -A_{cf} & Acc \end{bmatrix}$ be positive semi-definite
- $\mathbf{x}_f^T D_{ff} \mathbf{x}_f \leq \lambda_{\text{min}} \mathbf{x}_f^T D_{ff} \mathbf{x}_f \leq \mathbf{x}_f^T A_{ff} \mathbf{x}_f \leq \lambda_{\text{max}} \mathbf{x}_f^T D_{ff} \mathbf{x}_f$
- $\nu_{\text{min}} \mathbf{x}_c^T S \mathbf{x}_c \leq \mathbf{x}_c^T \hat{A}_{cc} \mathbf{x}_c \leq \nu_{\text{max}} \mathbf{x}_c^T S \mathbf{x}_c$

Then,

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

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

Role of Partitioning

Bound depends on

- equivalence of D_{ff} and A_{ff}
- equivalence of S and $\hat{A}_{cc}$

Goals of partition are

- effective reduction, $|C| \ll |F|$
- efficient computation of $D_{ff}^{-1}\mathbf{y}_f$
- good equivalence, λ_{max} small

Diagonal Dominance

Jacobi on A_{ff} converges if it is diagonally dominant
Stronger dominance $\longrightarrow$ faster convergence

A_{ff} is θ-dominant if, for each $i \in F$,

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

Partitioning Goal: Find largest set F such that A_{ff} is θ-dominant.

Complexity

The problem,

$$\max\{|F| : A_{ff} \text{ is } \theta\text{-dominant}\},$$

is NP-complete.

Instead, use simple greedy strategy:

- define measure of suitability for F
- Add all acceptable points to F
- Remove some unsuitable points into C
- Update measures of undecided points

The Symmetric Case

Measure is given by diagonal dominance

- Initialize $U = \{1, \ldots, n\}$, $F = C = \emptyset$
- For each point in U, compute $\hat{\theta}_i = \dfrac{|a_{ii}|}{\displaystyle\sum_{j \in F \cup U} |a_{ij}|}$
- Whenever $\hat{\theta}_i \geq \theta$, $i \rightarrow F$
- While $U \neq \emptyset$, pick $j = \text{argmin}_{i \in U}\{\hat{\theta}_i\}$
  - $j \rightarrow C$
  - Update $\hat{\theta}_i$ for all $i \in U$ with $a_{ji} \neq 0$

The Non-Symmetric Case

Separate measures for rows and columns

- Accept/reject rows based on row diagonal dominance
- Accept/reject columns based on interaction with rows

Same strategy as symmetric case, but now

- look for dominance of row by any eligible column
- accept row/column pairs that give θ-dominance
- reject rows whenever no domination is possible
- reject single column when no row can be sorted

Resulting partition comes from nonsymmetric permutation

Theory and Practice

For θ-dominant A_{ff}, want $\sigma(D_{ff}^{-1} A_{ff})$ bounded

- True if $D_{ff} = \text{diag}(A_{ff})$
  - sparsest possible ILU of A_{ff}
- More fill within incomplete factorization should give better equivalence

Theory and Practice

For θ-dominant A_{ff}, want $\sigma(D_{ff}^{-1}A_{ff})$ bounded

- True if $D_{ff} = \text{diag}(A_{ff})$
  - sparsest possible ILU of A_{ff}
- More fill within incomplete factorization should give better equivalence

Combine dominance-based partitioning with classical algebraic coarsening

- Diagonal-dominance partitioning
- ILUT, fixed drop and fill thresholds
- Compute $S \approx \hat{A}_{cc}$ using thresholding
- Recursively solve coarse-scale system

PDE Test Problems

Two-dimensional bilinear finite element discretizations of

$$-\nabla \cdot K(x,y)\nabla p(x,y) = 0.$$

Problem 1: $K(x,y) = 1$

Problem 2: $K(x,y) = 10^{-8} + 10(x^2 + y^2)$

Problem 3: $K(x,y) = 10^{-8}$ on 20% of the cells, chosen randomly; $K(x,y) = 1$ otherwise

Problem 4: $K(x,y) = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$

ARMS Results

Prob.	Grid	c_B	t_{setup}	t_{solve}	# iters.
1	128×128	2.59	0.3	0.3	28
	256×256	2.65	1.5	2.5	44
	512×512	2.68	12.7	24.5	82
2	128×128	2.60	0.3	0.4	31
	256×256	2.65	1.5	3.4	56
	512×512	2.68	12.7	31.7	97
3	128×128	1.40	0.2	0.4	32
	256×256	1.41	0.7	2.5	45
	512×512	1.42	3.1	25.1	83
4	128×128	1.61	0.2	0.3	26
	256×256	1.62	0.8	2.3	42
	512×512	1.63	3.3	17.3	65

General ARMS Tests

- Test set from Rutherford-Appleton Labs
- 22 Selected problems, from 120K to 3.6M non-zeros
- Compared to ILUTP, fill factors adjusted to match ARMS preconditioner complexities

N. Gould and J. Scott, ACM Trans. Math. Softw. 2004, **30**:300-325

General ARMS Tests

- Test set from Rutherford-Appleton Labs
- 22 Selected problems, from 120K to 3.6M non-zeros
- Compared to ILUTP, fill factors adjusted to match ARMS preconditioner complexities

Results:

- ARMS converged in available memory (2GB + 1 GB swap) on 21 problems
- ILUTP converged for 12 problems, limited to memory or $2\times$ ARMS iteration count
- ILUTP needed fewer iterations for 8 problems
- Equal iterations for 1
- ARMS needed fewer iterations for 12

N. Gould and J. Scott, ACM Trans. Math. Softw. 2004, **30**:300-325

General ARMS Tests

- Test set from Rutherford-Appleton Labs
- 22 Selected problems, from 120K to 3.6M non-zeros
- Compared to ILUTP, fill factors adjusted to match ARMS preconditioner complexities

Results:

- ARMS converged in available memory (2GB + 1 GB swap) on 21 problems
- ILUTP converged for 12 problems, limited to memory or $2\times$ ARMS iteration count
- ILUTP needed least time for 6 problems
- Equal time for 1
- ARMS faster for 14 problems

N. Gould and J. Scott, ACM Trans. Math. Softw. 2004, **30**:300-325

Nonsymmetric Tests

- Test problems from earlier paper (58 matrices)
- Test problems from circuit simulation (41 matrices)

Compare using performance profiles

- S = set of solvers
- P = set of problems
- s_{ij} = performance of solver $i \in S$ on problem $j \in P$

Define $\hat{s}_j = \min_{i \in S} \{s_{ij}\}$, then take

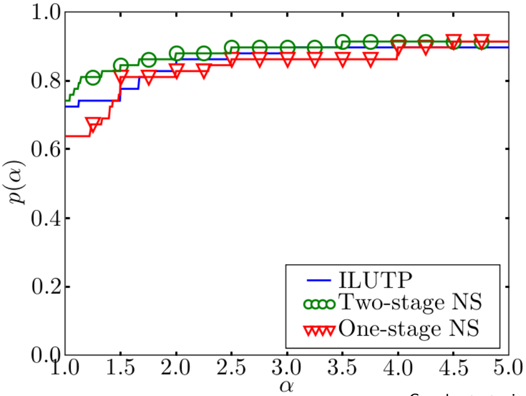
$$p_i(\alpha) = \frac{|\{j : s_{ij} \leq \alpha \hat{s}_j\}|}{|P|}$$

Y. Saad, SIAM J. Sci. Comp. 2006, **27**:1032-1057

E. Dolan and J. Moré, Math. Program., Ser. A 2002, **91**:201-213

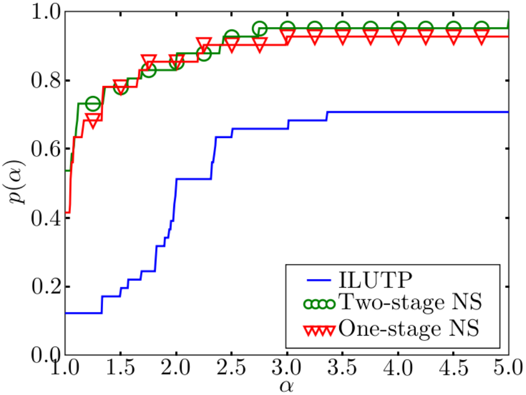
General Nonsymmetric Tests

- ILUTP, ARMS
  - use new (single-stage) partitioning and old (two-stage) approach
- 58 problems from Harwell-Boeing collection
  - All RUA matrices with dimension > 500



Circuit Simulation Tests

- ILUTP, ARMS
  - use new (single-stage) partitioning and old (two-stage) approach
- 41 problems from UF collection
  - Bomhof, Hamm, Schenk, and Wang collections



Further Reorderings

Can ARMS partitions be improved by further reordering?

- A_{ff} block ordered as F-rows are selected
- Consider RCM, dissection, MMD, QMD, and AMF

Reordering often improves iteration times

- Improvement usually slight
- Added setup cost not usually recovered
- RCM or One-way Dissection work best
- Consistent with earlier studies of incomplete factorizations

---

I. Duff and G. Meurant, BIT 1989, **29**:635-657

Summary

- Theoretical motivation: fine-scale spectral equivalence
- Choose partition to guarantee good equivalence
- Diagonal dominance is simple, but effective
- Multilevel results show robustness and efficiency
- Returns diminishing for improved partitions

---

http://www.cs.umn.edu/~maclach/research/selection.pdf

http://www.cs.umn.edu/~maclach/research/nonsymm.pdf

Summary

- Theoretical motivation: fine-scale spectral equivalence
- Choose partition to guarantee good equivalence
- Diagonal dominance is simple, but effective
- Multilevel results show robustness and efficiency
- Returns diminishing for improved partitions

Future Directions

- More complicated measures
- Better tuning of rest of ARMS solver
- Use spectral equivalence ideas to improve performance

---

http://www.cs.umn.edu/~maclach/research/selection.pdf

http://www.cs.umn.edu/~maclach/research/nonsymm.pdf

Current Work

Can we better use diagonal dominance of A_{ff} in choice of D_{ff}?

- Consider ILU vs. MILU
  - For M-Matrices, MILU gives better equivalence than ILU
- A_{ff} is θ-diagonally dominant

Idea: Use compensation within ILU to improve/guarantee spectral equivalence

---

I. Gustafsson, BIT 1978, **18**:142-156

ARMS vs. AMG

ARMS is additive, AMG is multiplicative

- Multigrid equivalent of ARMS is AMGr
  - Relaxation based only on A_{ff}
  - Interpolation based on approximation to A_{ff}^{-1}
  - Variational coarse-grid operator
- Additive preconditioner setting can be more forgiving
- Multiplicative solver setting can be more efficient

ARMS “works” more often than AMG
When AMG “works”, it is often more efficient than ARMS

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