5.3 Preconditioning

Direct solvers:
Sequential, loosing sparsity

Iterative solvers:
easy parallel and sparse,
but possibly slowly convergent

Combination of both methods:

Include preconditioner $M \approx A$ in the form $M^{-1} A x = M^{-1} b$, such that

- M is easy to deal with in parallel (reduced approximate direct solver)
- spectrum of $M^{-1} A$ is much better clustered

Or include preconditioner $M \approx A^{-1}$ in the form $M A x = M b$, such that

- M is easy to deal with in parallel (reduced approximate inverse)
- spectrum of $M A$ is much better clustered

Stationary Preconditioners

General both sided preconditioning: $Ax = b \leftarrow\rightarrow M\,A\,(K\,y) = M\,b$ and $K\,y = x$

Stationary iteration to splitting $A = M - N$:

Convergence depends on $\| I - M^{-1} A \| < 1$

That is exactly a condition for a good preconditioner:

spectrum clustered around 1, $M^{-1} A \approx I$

If splitting leads to fast convergence, than M is also a good preconditioner.

In this sense, pcg with stationary preconditioner M can be seen as an accelaration

of the stationary method with splitting M-N.

Stationary Preconditioners II

Jacobi splitting with D = diag(A) gives Jacobi preconditioner   M:= D

Gauss-Seidel splitting M = D – L leads to Gauss-Seidel preconditioner

Relaxation: $x_{k,new} := (1 - \omega)\, x_{k-1} + \omega\, x_k$

convex combination of old and new iterate (Jacobi or GS)

Symmetrization: first iteration with preconditioner M
second iteration with M^T

$$M_{new} := M + M^T - M^T A M$$

Special case damped GS → SSOR: $\frac{1}{2-\omega}\left(\frac{1}{\omega}D - L\right)\left(\frac{1}{\omega}D\right)^{-1}\left(\frac{1}{\omega}D - L\right)^T$

ILU Preconditioner

Idea: Apply Gauss-Elimination algorithm, but only on allowed pattern!

Incomplete LU factorization → ILU

Reduce in the FOR-loops the indices to the indices with

- allowed pattern, e.g. ILU(0) for pattern of A
- values that are not to small, ILUT for ILU with treshold

Leads to approximate LU factorization

$$A = L\,U + R, \qquad \text{preconditioner} \quad M = L\,U$$

with all ignored fill-in entries collected in R.

MILU: collect all ignored fill-in entries on the related main diagonal elements → maintains the row sum or the action on $(1,1,\ldots,1)^T$

Overview explicit preconditioners

ILU and stationary methods use approximations on A itself.

The resulting preconditioners are given by triangular matrices L,
that have to be solved in each iteration step: $L^{-1}x_k$! Strongly sequential!

Jacobi easy to parallelize, but slow convergence.

Question: How to derive preconditioners that lead to fast convergence and are easy to parallelize?

Idea: Find approximations on $M \approx A^{-1}$.
Then the solution of the linear system given by the preconditioner, is only Mx_k, a matrix vector product!

Parallel Preconditioning

Find preconditioner M , that satisfies three conditions:

(i) The computation of M is fast in parallel

(ii) The application M x in each iteration step is easy in parallel

(iii) The spectrum AM or MA is clustered → fast convergence

Examples: For GS is (i) and (iii) OK, but not (ii)

For Jacobi is (i) and (ii) OK, but not (iii)

For ILU is (iii) OK, but not (i) and (ii)

Polynomial Preconditioners

Characteristic polynomial for A:

$$0 = q(A) = \gamma_n A^n + \gamma_{n-1} A^{n-1} + \ldots + \gamma_1 A + \gamma_0 I$$

Gives polynomial representation for A^{-1} ($\gamma_0 \neq 0$) :

$$A^{-1} = \frac{1}{\gamma_0}\left(-\gamma_n A^{n-1} - \gamma_{n-1} A^{n-2} - \ldots - \gamma_1 I\right) = p(A)$$

Therefore, it makes sense to approximate A^{-1} by a polynomial in A.

Better approximation by finding region **S** in **R** or **C** that contains nearly all eigenvalues, and then find polynomial p that is near the inverse in **S**

$$\min_{p_n} \left\| P(A)A - I \right\|, \quad \min_{p_{n(x)}} \left(\max_{\lambda \in S} \left| p(\lambda)\lambda - 1 \right| \right)$$

Solution: Normalized Chebyshev polynomials

Advantage of polynomial preconditioner: better in parallel

Disadvantage: Not-optimal approximation in the same Krylov subspace

Sparse approximate Inverses

Other approach for approximating A^{-1} by norm minimization:

$$\min_{M \in \wp} \| AM - I \|$$

over some sparsity pattern **P**.

Choice of the norm?
- analytic (to allow the explicit solution of this problem)
- easy to compute (in parallel)

Optimal norm: Frobenius norm

$$\|A\|_F^2 := \sum_{i,j=1}^{n} a_{i,j}^2 = \sum_{j=1}^{n} \|A_{\bullet,j}\|^2 = trace\left(A^T A\right) = \langle A, A \rangle$$

SPAI in parallel

First, we choose the pattern **P** in a static way a priori, e.g. as the pattern of A

$$\min_{M \in \wp} \left\| AM - I \right\|_F^2 = \min_{M \in \wp} \sum_{j=1}^{n} \left\| (AM - I) e_j \right\|_2^2 =$$

$$= \sum_{j=1}^{n} \min_{M_j \in \wp_j} \left\| AM_j - e_j \right\|^2$$

Hence, to minimize the Frobenius norm, we have to solve n Least Squares

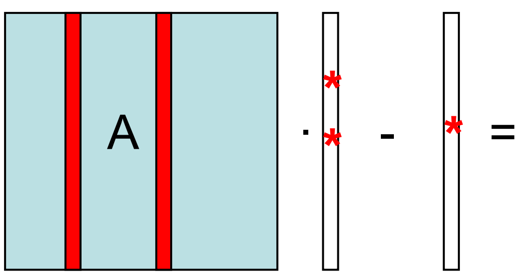
problems in the sparse columns of M.

This can be done fully in parallel!   But costs for LS problems?

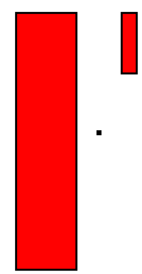
SPAI and LS

$$\min_{M_j \in \wp_j} \left\| AM_j - e_j \right\| = \min \left\| A \begin{pmatrix} 0 \\ * \\ 0 \\ * \\ 0 \\ 0 \end{pmatrix} - e_j \right\| = \min \left\| AM_j(J_j) - e_j \right\|$$

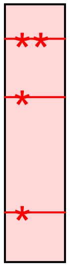
Denote by J_j the set of allowed indices in the j-th column of M.



$$A(:, J_j) \cdot M(J_j) =$$

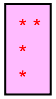


SPAI and Sparse LS



$A(:,J_j)$ is a sparse rectangular matrix.

I_j index set, shadow of J_j.

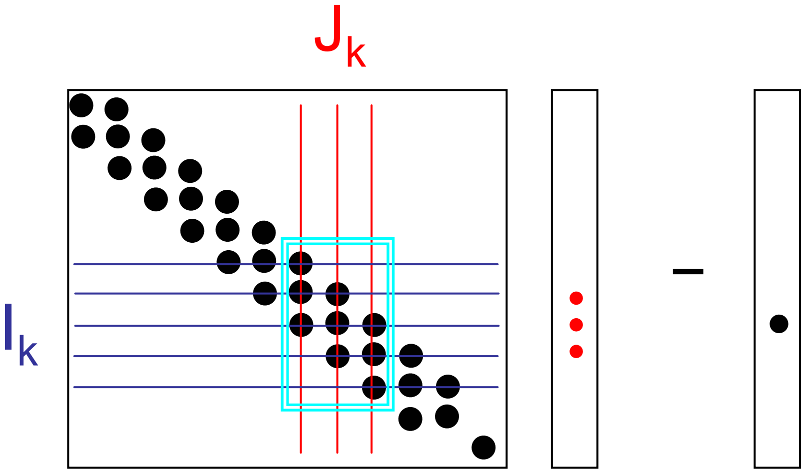


We can reduce the sparse LS to $A(I_j,J_j)$:

$$\min_{M_j} \left\| A(I_j, J_j) M(J_j) - e(I_j) \right\| = \min_{M_j} \left\| \tilde{A}\tilde{M}_j - \tilde{e}_j \right\|$$

Solve small LS problem by Householder QR for $A(I_j,J_j)$ → $\tilde{M}_j$ → M_j

Computing M_k



$$= \quad A M_k - e_k$$

Delete superfluous zeros in Least Squares Problem:

For index set J_k in M_k keep only $A(: , J_k)$

In $A(: , J_k)$ keep only nonzero rows $A(I_k , J_k)$

Solve small Least Squares problem in $A(I_k , J_k)$, e.g. by QR-decomposition, Householder method.

Sparsity Pattern?

A^{-1} will be no more sparse!

As a priori choice of a good approximate sparsity pattern for M we can choose

the pattern of

- A^k
- $(A^T A)^k A^T$ for some k=1,2
- A_ε with sparsified A
- a combination of above

A_ε by sparsification of A: delete all entries with

$$|A_{i,j}| < \varepsilon$$

Dynamic Pattern Finding (SPAI)

- Start with thin approximate pattern J_k for M_k
- Compute optimal column $M_{k,opt}(J_k)$ by LS
- Find new entry j for M_k such that $M_{k,opt}(J_k) + \lambda e_j$ has smaller residual in the Frobenius norm.

$$\min \left\| A(M_k + \lambda e_j) - e_k \right\|^2 = \min \left\| (AM_k - e_k) + \lambda A e_j \right\|^2 =$$

$$\min \left(\left\| r_k \right\|^2 + 2\lambda (r_k^T A_j) + \lambda^2 \left\| A_j \right\|^2 \right)$$

Choose index j with $r_k^T A_j \neq 0$ and $\lambda_{opt} = -\frac{r_k^T A_j}{\left\| A_j \right\|^2}$

with $\min = \left\| r_k \right\|^2 - \frac{(r_k^T A_j)^2}{\left\| A_j \right\|^2}$

Block SPAI

Partition the given matrix in small blocks
(2 x 2 or 3 x 3) and apply the Frobenius norm minimization with blockwise pattern.

Advantage: Underlying block structure will also appear in the pattern of A^{-1} $\rightarrow$ *improved pattern*

Iterative SPAI

Start with pattern of A $\rightarrow$ M_1

Construct M_2 relative to new matrix AM_1

Construct M_3 relative to new matrix AM_1M_2

…..

Advantage: Cheaper, but inferior approximation

Factorized SPAI, FSPAI

Approximate the inverse Cholesky factor of $A = L_A^T \cdot L_A$

$$A^{-1} = (L_A^T L_A)^{-1} = L_A^{-1} L_A^{-T} \approx L \cdot L^T$$

$$L_A^{-1} \approx L \quad \Longrightarrow \quad \min \left\| L_A L - I \right\|_F$$

Normal equations give $A(J_k, J_k) \cdot L(J_k) = \alpha \cdot e_k$

The same L results from $\min \dfrac{trace(L^T AL)}{\det(L^T AL)^{(1/n)}}$

SPAI and Target Matrices

$$\min \left\| AM - P \right\|_F$$

Assume, P is a good sparse preconditioner for A. Improve P by computing M and solving the above Frobenius norm minimization.

If A is given by two parts, e.g. an advection part and an diffusion part we can choose P as Laplacian relative to the diffusion part - easy to solve - and then we add M for improving P relative to the advection part

MSPAI Probing

Generalize Frobenius norm minimization to

$$\min_M \|CM - B\|_F = \min_M \left\| \begin{pmatrix} C_0 \\ \rho \cdot u^T \end{pmatrix} M - \begin{pmatrix} B_0 \\ \rho \cdot v^T \end{pmatrix} \right\|_F$$

For example

$$\min_M \|CM - B\|_F = \min_M \left\| \begin{pmatrix} A \\ \rho \cdot e^T A \end{pmatrix} M - \begin{pmatrix} I \\ \rho \cdot e^T \end{pmatrix} \right\|_F$$

Original SPAI extended by an additional norm minimization to deliver especially good results on vector e.

Similarly
$$\min_M \left\| \begin{pmatrix} I \\ \rho \cdot e^T \end{pmatrix} (C_0 M - B_0) \right\|_F = \min_M \| W (C_0 M - B_0) \|_F$$

$$\min_M \left\| \begin{pmatrix} I \\ \rho \cdot e^T \end{pmatrix} (AM - I) \right\|_F = \min_M \| W (AM - I) \|_F$$

Probing

Find preconditioner of special form (tridiag, band) for preconditioning a matrix that is not given explicitly, but only by its action on certain vectors, e.g. $e^T S = f^T$.

Example: Schur complement S.

Choose e.g. $e=(1,1,\ldots,1)^T$, preconditioner as diagonal matrix $D=\mathrm{diag}(d_1,\ldots,d_n)$.

Then we have to satisfy $e^T D = e^T S = f^T$.

After the computation of f the solution is given by $d_j = f_j$.

Disadvantage: Can use only very special pattern for M and probing vectors e.

MSPAI regularization for the general probing method with any pattern and any e:

$$\min_{\wp} \left\| M - \tilde{S} \right\|_F^2 + \rho^2 \left\| e^T M - e^T S \right\|^2$$

with any sparse approximation $\tilde{S}$ *of* S