Chapter 1. On the resolution of linear systems

Laurence HALPERN

LAGA - Université Paris 13

HCMV, november 2016

Purpose

Solve $AX = b$.

- A is a squared matrix,
- b is a given righthand side,

Purpose

Solve $AX = b$.

- A is a squared matrix,
- b is a given righthand side, or a family of given righthand sides

Outline

Description

$$\underbrace{\begin{pmatrix} 1 & 3 & 1 \\ 1 & 1 & -1 \\ 3 & 11 & 6 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \\ \\ \\ \end{pmatrix}}_{X} = \underbrace{\begin{pmatrix} 9 \\ 1 \\ 36 \end{pmatrix}}_{b}$$

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 11 & 6 & 36 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 2 & 3 & 9 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 0 & 1 & 1 \end{array}\right)$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 1 & 1 \end{pmatrix}}_{M} \underbrace{\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 11 & 6 & 36 \end{array}\right)}_{(A|b)} = \underbrace{\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 0 & 1 & 1 \end{array}\right)}_{(U|Mb)}$$

$$Ax = b \iff Ux : MAx = Mb$$

M is a preconditioner

$$M = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 1 & 1 \end{pmatrix} \longrightarrow L := M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & -1 & 1 \end{pmatrix}$$

$$U = MA \iff A = LU, Ax = b \iff LUx = b$$

1. LU decomposition $\mathcal{O}(\frac{2n^3}{3})$ elementary operations.

$$Ax = b \iff Ux : MAx = Mb$$

M is a preconditioner

$$M = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 1 & 1 \end{pmatrix} \longrightarrow L := M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & -1 & 1 \end{pmatrix}$$

$$U = MA \iff A = LU, Ax = b \iff LUx = b$$

1. LU decomposition $\mathcal{O}(\frac{2n^3}{3})$ elementary operations.
2. Solve $Ly = b$ $\mathcal{O}(n^2)$ elementary operations.

$$Ax = b \iff Ux : MAx = Mb$$

M is a preconditioner

$$M = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 1 & 1 \end{pmatrix} \longrightarrow L := M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & -1 & 1 \end{pmatrix}$$

$$U = MA \iff A = LU, Ax = b \iff LUx = b$$

1. LU decomposition $\mathcal{O}(\frac{2n^3}{3})$ elementary operations.
2. Solve $Ly = b$ $\mathcal{O}(n^2)$ elementary operations.
3. Solve $Ux = y$ $\mathcal{O}(n^2)$ elementary operations.

$$Ax = b \iff Ux : MAx = Mb$$

M is a preconditioner

$$M = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 1 & 1 \end{pmatrix} \longrightarrow L := M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & -1 & 1 \end{pmatrix}$$

$$U = MA \iff A = LU, Ax = b \iff LUx = b$$

1. LU decomposition $\mathcal{O}(\frac{2n^3}{3})$ elementary operations.
2. Solve $Ly = b$ $\mathcal{O}(n^2)$ elementary operations.
3. Solve $Ux = y$ $\mathcal{O}(n^2)$ elementary operations.

For P values of the righthand side, $N_{op} \sim \frac{2n^3}{3} + P \times 2n^2$.

Theoretical results

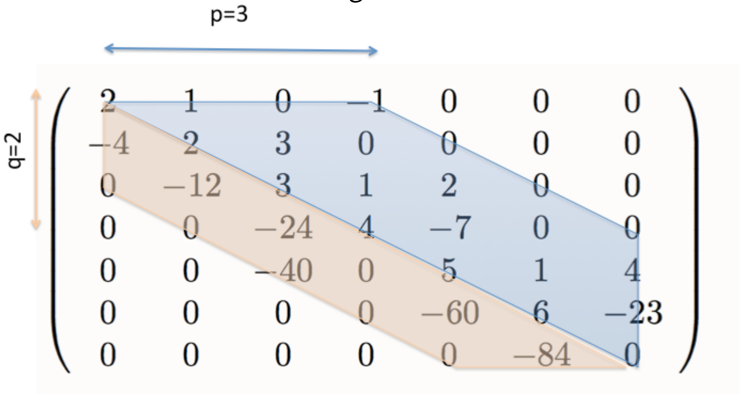
Theorem 1 Let A be an invertible matrix, with principal minors $\neq 0$. Then there exists a unique matrix L lower triangular with $l_{ii} = 1$ for all i, and a unique matrix U upper triangular, such that $A = LU$. Furthermore $det\,(A) = \prod_{i=1}^{n} u_{ii}$.

Theorem 2 Let A be an invertible matrix. There exist a permutation matrix P, a matrix L lower triangular with $l_{ii} = 1$ for all i, and a matrix U upper triangular, such that

$$PA = LU$$

Sparse and banded matrices

Wilkinson 69' : *any matrix with enough zeros that it pays to take advantage of them.*



A banded matrix, upper bandwidth $p = 3$ and lower bandwidth $q = 2$, in total $p + q + 1$ nonzero diagonals.

Sparse and banded matrices

$$U = \begin{pmatrix} 2 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 4 & 3 & -2 & 0 & 0 & 0 \\ 0 & 0 & 12 & -5 & 2 & 0 & 0 \\ 0 & 0 & 0 & -6 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 & 9 & -11 \\ 0 & 0 & 0 & 0 & 0 & 0 & -102.7 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & -3.3 & 2.81 & 0 & 0 & \\ 0 & 0 & 0 & 0 & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -9.3 & 1 \end{pmatrix}$$

L lowerbanded $q = 2$, and U upperbanded $p = 3$.

Manipulating sparse matrices in matlab

```
>>S=sparse([2 3 1 2],[1 1 2 3],[2 4 1 3])
 S =
   (2,1)        2
   (3,1)        4
   (1,2)        1
   (2,3)        3
>>S=speye(2,3)
S =
   (1,1)        1
   (2,2)        1
>>n=4;
>>e=ones(n,1)
e =

     1
     1
     1
     1
```

```
>>A=spdiags([e -2*e e],-1:1,n,n)
A =

   (1,1)       -2
   (2,1)        1
   (1,2)        1
   (2,2)       -2
   (3,2)        1
   (2,3)        1
   (3,3)       -2
   (4,3)        1
   (3,4)        1
   (4,4)       -2

>>full(A)
ans =

    -2     1     0     0
     1    -2     1     0
     0     1    -2     1
```

Manipulating sparse matrices in matlab

```
>>B = repmat((1:n)',1,3)
B =
     1     1     1
     2     2     2
     3     3     3
     4     4     4
>>A=spdiags(B,[-2 0 1],n,n)
A =
   (1,1)        1
   (3,1)        1
   (1,2)        2
   (2,2)        2
   (4,2)        2
   (2,3)        3
   (3,3)        3
   (3,4)        4
   (4,4)        4
```

```
>>full(A)

ans =

     1     2     0     0
     0     2     3     0
     1     0     3     4
     0     2     0     4
```

Sparse and banded matrices with pivoting

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -0.5 & -0.17 & -0.05 & -0.21 & 0.025 & 0.0027 & 1 \end{pmatrix}$$

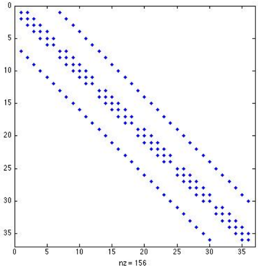
$$U = \begin{pmatrix} -4 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & -12 & 3 & 1 & 2 & 0 & 0 \\ 0 & 0 & -40 & 0 & 5 & 1 & 4 \\ 0 & 0 & 0 & 4 & -10 & -0.6 & -2.4 \\ 0 & 0 & 0 & 0 & -60 & 6 & -23 \\ 0 & 0 & 0 & 0 & 0 & -84 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.275 \end{pmatrix}$$

The permutation matrix

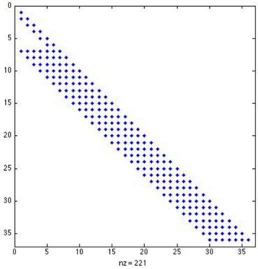
$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Cholewski

36×36 sparse matrix of $2 - D$ finite differences in a square.
With the command spy de matlab



A bandmatrix sparse matrix



Corresponding Cholewski

Summary

Direct methods for small full systems

Iterative methods $\rightarrow$ matrix vector product $\rightarrow$ sparse systems.

Outline

Stationary iterative methods

$$AX = b \; ; \quad A = M - N \; ; \qquad MX = NX + b,$$

$$MX^{m+1} = NX^m + b.$$

Use A = D - E - F.

1. Jacobi : $M = D$ diagonal part of A.

Stationary iterative methods

$$AX = b \ ; \quad A = M - N \ ; \qquad MX = NX + b,$$
$$MX^{m+1} = NX^{m} + b.$$

Use A = D - E - F.

1. Jacobi : $M = D$ diagonal part of A.
2. Gauss-Seidel : $M = D - E$ lower part of A.

Stationary iterative methods

$$AX = b\ ; \quad A = M - N\ ; \qquad MX = NX + b,$$
$$MX^{m+1} = NX^m + b.$$

Use A = D - E - F.

1. Jacobi : $M = D$ diagonal part of A.
2. Gauss-Seidel : $M = D - E$ lower part of A.
3. Relaxation :

$$M = \frac{1}{\omega}D - E, \;\; N = F + \frac{1-\omega}{\omega}D$$

Stationary iterative methods

$$AX = b\ ; \quad A = M - N\ ; \qquad MX = NX + b,$$

$$MX^{m+1} = NX^m + b.$$

Use A = D - E - F.

1. Jacobi : $M = D$ diagonal part of A.
2. Gauss-Seidel : $M = D - E$ lower part of A.
3. Relaxation :

$$M = \frac{1}{\omega}D - E,\ \ N = F + \frac{1-\omega}{\omega}D$$

4. Richardson algorithm

$$X^{m+1} = X^m - \rho r^m = X^m - \rho(AX^m - b)$$

Stationary iterative methods

$$AX = b \; ; \quad A = M - N \; ; \qquad MX = NX + b,$$

$$MX^{m+1} = NX^m + b.$$

Use A = D - E - F.

1. Jacobi : $M = D$ diagonal part of A.
2. Gauss-Seidel : $M = D - E$ lower part of A.
3. Relaxation : $\hat{U}^{m+1}$ obtained by Gauss-Seidel,

$$X^{m+1} = \omega \hat{U}^{m+1} + (1-\omega) X^m.$$

$$M = \frac{1}{\omega} D - E, \;\; N = F + \frac{1-\omega}{\omega} D$$

4. Richardson algorithm

$$X^{m+1} = X^m - \rho r^m = X^m - \rho(AX^m - b)$$

Stationary iterative methods

$$AX = b \; ; \quad A = M - N \; ; \qquad MX = NX + b,$$

$$MX^{m+1} = NX^m + b.$$

Use A = D - E - F.

1. Jacobi : $M = D$ diagonal part of A.
2. Gauss-Seidel : $M = D - E$ lower part of A.
3. Relaxation :

$$M = \frac{1}{\omega}D - E, \;\; N = F + \frac{1-\omega}{\omega}D$$

4. Richardson algorithm

$$X^{m+1} = X^m - \rho r^m = X^m - \rho(AX^m - b)$$

$$M = \tfrac{1}{\rho}I$$

Stationary iterative methods

$$AX = b \; ; \quad A = M - N \; ; \qquad MX = NX + b,$$
$$MX^{m+1} = NX^m + b.$$

Use A = D - E - F.

1. Jacobi : $M = D$ diagonal part of A.
2. Gauss-Seidel : $M = D - E$ lower part of A.
3. Relaxation :

$$M = \frac{1}{\omega} D - E, \;\; N = F + \frac{1 - \omega}{\omega} D$$

4. Richardson algorithm

$$X^{m+1} = X^m - \rho r^m = X^m - \rho (AX^m - b)$$

$$M = \tfrac{1}{\rho} I \quad \rho_{opt} = \tfrac{2}{\lambda_1 + \lambda_n}$$

Stationary methods, continue

$$
\begin{aligned}
MX^{m+1} = NX^m + b \quad &\iff MX^{m+1} = (M - A)X^m + b \\
&\iff X^{m+1} = (I - M^{-1}A)X^m + M^{-1}b \\
&\iff \text{fixed point algorithm to solve } M^{-1}AX = M^{-1}b
\end{aligned}
$$

Stationary methods, continue

$$
\begin{aligned}
MX^{m+1} = NX^m + b &\iff MX^{m+1} = (M - A)X^m + b \\
&\iff X^{m+1} = (I - M^{-1}A)X^m + M^{-1}b \\
&\iff \text{fixed point algorithm to solve } M^{-1}AX = M^{-1}b
\end{aligned}
$$

Preconditioning

$$
\begin{aligned}
AX = b &\iff M^{-1}AX = M^{-1}b \\
&\iff X = (I - M^{-1}A)X + M^{-1}b
\end{aligned}
$$

Stationary methods, continue

$$\text{Error } e^m := X - X^m,$$
$$\text{Residual } \; r^m := b - AX^m = AX - AX^m = Ae^m.$$

$$MX^{m+1} = NX^m + b$$

$$MX = NX + b$$

$$Me^{m+1} = Ne^m$$

$$e^{m+1} = M^{-1}Ne^m$$

$R = M^{-1}N$ is the iteration matrix

Useful alternative formula $R = I - M^{-1}A$.

Fundamentals tools

$$X^{m+1} = RX^m + \tilde{b}, \quad e^{m+1} = Re^m, \ R = M^{-1}N.$$

Theorem The sequence is convergent for any initial guess X^0 if and only if $\rho(R) < 1$.
$\rho(R) = \max\{|\lambda|, \lambda \text{ eigenvalue of } A\}$: convergence factor.

$$\frac{\|e^{m+1}\|}{\|e^m\|} \lesssim \rho(R)$$

Convergence rate $C = -\ln_{10}\rho(R)$. $\|e^{m+1}\| \sim 10^{-C}\|e^m\|$.
C digits per iteration.
To reduce the initial error by a factor ϵ, we need

$$\frac{\|e^m\|}{\|e^0\|} \lesssim (\rho(R))^m \sim \epsilon$$

So we have $M \sim \dfrac{\ln \epsilon}{\ln \rho(R)}$.

M-matrices

Definition : $A \in \mathbb{R}^{n \times n}$ is a M-matrix if

1. $a_{ii} > 0$ for $i = 1, \ldots, n$,
2. $a_{ij} \leq 0$ for $i \neq j$, $i, j = 1, \ldots, n$,
3. A is invertible,
4. $A^{-1} \geq 0$.

Theorem If A is a M-matrix and $A = M - N$ is a regular splitting (M is invertible and both M^{-1} and N are nonnegative), then the stationary method converges.

Symmetric positive definite matrices

Householder-John theorem : Suppose A is positive. If $M + M^T - A$ is positive definite, then $\rho(R) < 1$.

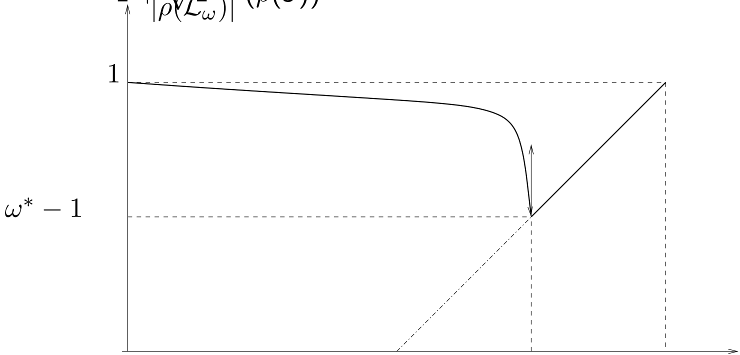
Corollary

1. If $D + E + F$ is positive definite, then Jacobi converges.
2. If $\omega \in (0, 2)$, then SOR converges.

Tridiagonale matrices

1. $\rho(\mathcal{L}_1) = (\rho(J))^2$: Jacobi Gauss-Seidel converge or diverge simultaneously. If convergent, Gauss-Seidel is twice as fast.
2. Suppose the eigenvalues of J are real. Then Jacobi and SOR converge or diverge simultaneously for $\omega \in]0, 2[$.
3. Same assumptions, SOR has an optimal parameter

$$\omega^* = \frac{2}{1 + \sqrt{1 - (\rho(J))^2}}, \quad \rho(\mathcal{L}_{\omega^*}) = \omega^* - 1.$$



Outline

Descent methods. A sdp

The descent directions p_m are given. Define

$$X^{m+1} = X^m + \alpha_m p^m, \quad e^{m+1} = e^m - \alpha_m p^m, \quad r^{m+1} = r^m - \alpha_m A p^m.$$

Theorem X is the solution of $AX = b \iff$ it minimizes over $\mathbb{R}^N$ the functional $J(y) = \frac{1}{2}(Ay, y) - (b, y)$.
Equivalent to minimizing
$G(y) = \frac{1}{2}(A(y - X), y - X) = \frac{1}{2}\|y - X\|_A^2$.
At step m, minimize J in the direction of p_m

$$\boxed{\alpha_m = \frac{(p^m, r^m)}{(Ap^m, p^m)}, \quad (p^m, r^{m+1}) = 0}$$

$$G(x^{m+1}) = G(x^m)(1 - \mu_m), \quad \mu_m = \frac{(r^m, p^m)^2}{(Ap^m, p^m)(A^{-1}r^m, r^m)}$$

Steepest descent (gradient à pas optimal)

$$p^m = r^m.$$

$$X^{m+1} = X^m + \alpha_m r^m, \quad e^{m+1} = e^m - \alpha_m r^m, \quad r^{m+1} = (I - \alpha_m A) p^m.$$

$$\boxed{\alpha_m = \frac{\|r^m\|^2}{(Ar^m, r^m)}, \quad (r^m, r^{m+1}) = 0}$$

$$G(x^{m+1}) = G(x^m)\left(1 - \frac{\|r^m\|^4}{(Ar^m, r^m)(A^{-1}r^m, r^m)}\right) \leq \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^2 G(x^m)$$

Conjugate gradient

$$X^{m+1} = X^m + \alpha_m p^m, \quad \alpha_m = \frac{(p^m, r^m)}{(Ap^m, p^m)}, \quad (r^m, p^{m-1}) = 0.$$

Search p^m as $p^m = r^m + \beta_m p^{m-1}$

$$G(x^{m+1}) = G(x^m)(1 - \mu_m)$$

$$\mu_m = \frac{(r^m, p^m)^2}{(Ap^m, p^m)(A^{-1}r^m, r^m)} = \frac{\|r^m\|^4}{(Ap^m, p^m)(A^{-1}r^m, r^m)}$$

Maximize μ_m, or minimize

$$(Ap^m, p^m) = \beta_m^2 (Ap^{m-1}, p^{m-1}) + 2\beta_m (Ap^{m-1}, r^m) + (Ar^m, r^m)$$

$$\beta_m = -\frac{(Ap^{m-1}, r^m)}{(Ap^{m-1}, p^{m-1})} \quad \Rightarrow (Ap^{m-1}, p^m) = 0$$

$$(r^m, r^{m+1}) = 0, \quad \beta_m = \frac{\|r^m\|^2}{\|r^{m-1}\|^2}.$$

Other properties

Choose $p^0 = r^0$. Then $\forall m \geq 1$, if $r^i \neq 0$ for $i < m$.

1. $(r^m, p^i) = 0$ for $i \leq m - 1$.

Definition Krylov space $\mathcal{K}_m = \text{vec}(r^0, Ar^0, \ldots, A^{m-1}r^0)$.

Other properties

Choose $p^0 = r^0$. Then $\forall m \geq 1$, if $r^i \neq 0$ for $i < m$.

1. $(r^m, p^i) = 0$ for $i \leq m-1$.
2. $\text{vec}(r^0, \ldots, r^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.

Definition Krylov space $\mathcal{K}_m = \text{vec}(r^0, Ar^0, \ldots, A^{m-1} r^0)$.

Other properties

Choose $p^0 = r^0$. Then $\forall m \geq 1$, if $r^i \neq 0$ for $i < m$.

1. $(r^m, p^i) = 0$ for $i \leq m - 1$.
2. $\text{vec}(r^0, \ldots, r^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.
3. $\text{vec}(p^0, \ldots, p^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.

Definition Krylov space $\mathcal{K}_m = \text{vec}(r^0, Ar^0, \ldots, A^{m-1} r^0)$.

Other properties

Choose $p^0 = r^0$. Then $\forall m \geq 1$, if $r^i \neq 0$ for $i < m$.

1. $(r^m, p^i) = 0$ for $i \leq m-1$.
2. $\text{vec}(r^0, \ldots, r^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.
3. $\text{vec}(p^0, \ldots, p^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.
4. $(p^m, Ap^i) = 0$ for $i \leq m-1$.

Definition Krylov space $\mathcal{K}_m = \text{vec}(r^0, Ar^0, \ldots, A^{m-1} r^0)$.

Other properties

Choose $p^0 = r^0$. Then $\forall m \geq 1$, if $r^i \neq 0$ for $i < m$.

1. $(r^m, p^i) = 0$ for $i \leq m-1$.
2. $\text{vec}(r^0, \ldots, r^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.
3. $\text{vec}(p^0, \ldots, p^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.
4. $(p^m, Ap^i) = 0$ for $i \leq m-1$.
5. $(r^m, r^i) = 0$ for $i \leq m-1$.

Definition Krylov space $\mathcal{K}_m = \text{vec}(r^0, Ar^0, \ldots, A^{m-1} r^0)$.

Other properties

Choose $p^0 = r^0$. Then $\forall m \geq 1$, if $r^i \neq 0$ for $i < m$.

1. $(r^m, p^i) = 0$ for $i \leq m - 1$.
2. $\text{vec}(r^0, \ldots, r^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.
3. $\text{vec}(p^0, \ldots, p^m) = \text{vec}(r^0, Ar^0, \ldots, A^m r^0)$.
4. $(p^m, Ap^i) = 0$ for $i \leq m - 1$.
5. $(r^m, r^i) = 0$ for $i \leq m - 1$.

Definition Krylov space $\mathcal{K}_m = \text{vec}(r^0, Ar^0, \ldots, A^{m-1} r^0)$.

Other properties

Choose $p^0 = r^0$. Then $\forall m \geq 1$, if $r^i \neq 0$ for $i < m$.

1. $(r^m, p^i) = 0$ for $i \leq m-1$.
2. $\mathrm{vec}(r^0, \dots, r^m) = \mathrm{vec}(r^0, Ar^0, \dots, A^m r^0)$.
3. $\mathrm{vec}(p^0, \dots, p^m) = \mathrm{vec}(r^0, Ar^0, \dots, A^m r^0)$.
4. $(p^m, Ap^i) = 0$ for $i \leq m-1$.
5. $(r^m, r^i) = 0$ for $i \leq m-1$.

Definition Krylov space $\mathcal{K}_m = \mathrm{vec}(r^0, Ar^0, \dots, A^{m-1} r^0)$.

Theorem (optimality of CG) A symétrique définie positive,

$$\|x^m - x\|_A = \inf_{y \in x^0 + \mathcal{K}_m} \|y - x\|_A, \quad \|x\|_A = \sqrt{x^T A x}.$$

Final properties

Theorem Convergence in at most N steps (size of the matrix)

Theorem $\|x^m\|_A \leq 2 \dfrac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1} \|x^{m-1}\|_A$

Method	Steepest descent	Conjugate gradient
Convergence factor ρ	$\dfrac{\kappa(A)-1}{\kappa(A)+1}$	$\dfrac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}$
$\rho(h=0.1)$	0.98	0.82
$N_{it}(h=0.1)$ to 10^{-2} error	230	23

Comparison

The algorithm

$$X^0 \text{chosen}, \quad p^0 = r^0 = b - AX^0.$$

While $m < Niter$ or $\|r^m\| \geq tol$, do

$$\begin{aligned}
\alpha_m &= \frac{\|r^m\|^2}{(Ap^m, p^m)}, \\
X^{m+1} &= X^m + \alpha_m p^m, \\
r^{m+1} &= r^m - \alpha_m A p^m, \\
\beta_{m+1} &= \frac{\|r^{m+1}\|^2}{\|r^m\|^2}, \\
p^{m+1} &= r^{m+1} - \beta_{m+1} p^m.
\end{aligned}$$

1-D Poisson problem

Poisson equation $-u'' = f$ on $(0,1)$,
Dirichlet boundary conditions $u(0) = g_g, \; u(1) = g_d$.
Second order finite difference stencil.

$$(0,1) = \cup(x_j, x_{j+1}), \quad x_{j+1} - x_j = h = \frac{1}{n+1}, \quad j = 0, \ldots, n.$$

$$-\frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} \sim f(x_i), \quad i = 1, \ldots n$$

$$u_0 = g_g, \quad u_{n+1} = g_d.$$

$$|u_i - u(x_i)| \leq h^2 \; \frac{\sup_{x\in[a,b]} |u^{(4)}(x)|}{12}$$

.

1-D Poisson problem

Discrete unknowns $U =^t (u_1, \ldots, u_n)$.

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & 0 & \\ & \ddots & \ddots & \ddots & \\ 0 & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \quad b = \begin{pmatrix} f_1 - \frac{g_g}{h^2} \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n - \frac{g_d}{h^2} \end{pmatrix}$$

The matrix A is symmetric definite positive.

Discrete problem to be solved is

$$AX = b$$

Condition number and error

$$AX = b, \quad A\hat{X} = \hat{b}$$

Define $\kappa(A) = \|A\|_2 \|A^{-1}\|_2$. If A is symmetric > 0, $\kappa(A) = \frac{\max \lambda_i}{\min \lambda_i}$.

Theorem

$$\frac{\|\hat{X} - X\|_2}{\|X\|_2} \leq \kappa(A) \frac{\|\hat{b} - b\|_2}{\|b\|_2}$$

and there is a b such that it is equal.

Eigenvalues of A $(h \times (n+1) = 1)$.

$$\lambda_k = \frac{2}{h^2}(1 - \cos \frac{k\pi}{n+1}) = \frac{4}{h^2} \sin^2 \frac{k\pi h}{2}, \quad V_k = (\sin \frac{jk\pi}{n+1})_{1 \leq i \leq n},$$

$$\kappa(A) = \frac{\sin^2 \frac{n\pi h}{2}}{\sin^2 \frac{\pi h}{2}} = \frac{\cos^2 \frac{\pi h}{2}}{\sin^2 \frac{\pi h}{2}} \sim \frac{4}{\pi^2 h^2}$$

Comparison of the iterative methods

Algorithm	**spectral radius** $\rho(R)$	$n = 5$	$n = 30$
Jacobi	$\cos \pi h$	0.866	0.995
Gauss-Seidel	$(\rho(J))^2 = \cos^2 \pi h$	0.750	0.990
SOR	$\dfrac{1 - \sin \pi h}{1 + \sin \pi h}$	0.333	0.816
steepest descent	$\dfrac{K(A) - 1}{K(A) + 1}$	0.866	0.995
conjugate gradient	$\dfrac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1}$	0.577	0.903

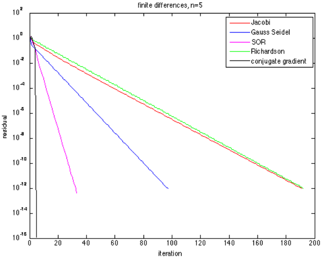
Reduction factor for one digit $M \sim -\dfrac{1}{\mathrm{Log}_{10}\rho(R)}$:

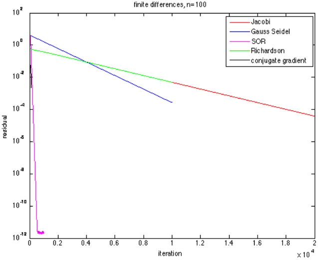
n	Jacobi	Gauss-Seidel	SOR	St Des	CG
5	16	8	2	16	4
30	448	224	11	448	23

Asymptotic behavior

Algorithm	spectral radius
Jacobi	$1 - \frac{\pi^2}{2} h^2,$
Gauss-Seidel	$1 - \pi^2 h^2,$
SOR	$1 - 2\pi h$
gradient	$1 - \pi h,$
conjugate gradient	$1 - \frac{\pi h}{2}.$

Convergence history





Number of elementary operations

Gauss elimination	n^2
optimal overrelaxation	$n^{3/2}$
FFT	$n \ln_2(n)$
conjugate gradient	$n^{5/4}$
multigrid	n

Asymptotic order of the number of elementary operations needed to solve the $1-D$ problem as a function of the number of grid points

Outline

Preconditioning : purpose

Take the system $AX = b$, with A symmetric definite positive, and the conjugate gradient algorithm. The speed of convergence of the algorithm deteriorates when $\kappa(A)$ increases. The purpose is to replace the problem by another system, better conditioned. Let M be a symmetric regular matrix. Multiply the system on the left by M^{-1}.

$$AX = b \iff M^{-1}AX = M^{-1}b \iff (M^{-1}AM^{-1})MX = M^{-1}b$$

Define

$$\tilde{A} = M^{-1}AM^{-1}, \quad \tilde{X} = MX, \quad \tilde{b} = M^{-1}b,$$

and the new problem to solve $\tilde{A}\tilde{X} = \tilde{b}$. Since M is symmetric, $\tilde{A}$ is symmetric definite positive. Write the conjugate gradient algorithm for this "tilde" problem.

The algorithm for $\tilde{A}$

$$\tilde{X}^0 \text{ given}, \quad \tilde{p}^0 = \tilde{r}^0 = \tilde{b} - \tilde{A}\tilde{X}^0.$$

While $m < Niter$ or $\|\tilde{r}^m\| \geq tol$, do

$$\begin{aligned}
\alpha_m &= \frac{\|\tilde{r}^m\|^2}{(\tilde{A}\tilde{p}^m, \tilde{p}^m)}, \\
\tilde{X}^{m+1} &= \tilde{X}^m + \alpha_m \tilde{p}^m, \\
\tilde{r}^{m+1} &= \tilde{r}^m - \alpha_m \tilde{A}\tilde{p}^m, \\
\beta_{m+1} &= \frac{\|\tilde{r}^{m+1}\|^2}{\|\tilde{r}^m\|^2}, \\
\tilde{p}^{m+1} &= \tilde{r}^{m+1} - \beta_{m+1}\tilde{p}^m.
\end{aligned}$$

Now define

$$p^m = M^{-1}\tilde{p}^m, \quad X^m = M^{-1}\tilde{X}^m, \quad r^m = M\tilde{r}^m,$$

and replace in the algorithme above.

The algorithm for A

$$Mp^0 = M^{-1}r^0 = M^{-1}b - M^{-1}AM^{-1}MX^0 \iff \begin{cases} p^0 = M^{-2}r^0, \\ r^0 = b - AX^0. \end{cases}$$

$$\|\tilde{r}^m\|^2 = (M^{-1}r^m, M^{-1}r^m) = (M^{-2}r^m, r^m)$$

Define $\boxed{z^m = M^{-2}r^m}$. Then $\boxed{\beta_{m+1} = \dfrac{(z^{m+1}, r^{m+1})}{(z^m, r^m)}}$.

$$(\tilde{A}\tilde{p}^m, \tilde{p}^m) = (M^{-1}AM^{-1}Mp^m, Mp^m) = (Ap^m, p^m)$$

$$\Rightarrow \boxed{\alpha_m = \frac{(z^m, r^m)}{(Ap^m, p^m)}}.$$

$$MX^{m+1} = MX^m + \alpha_m Mp^m \iff \boxed{X^{m+1} = X^m + \alpha_m p^m}.$$

$$M^{-1}r^{m+1} = M^{-1}r^m - \alpha_m M^{-1}AM^{-1}Mp^m \iff \boxed{r^{m+1} = r^m - \alpha_m Ap^m}.$$

$$Mp^{m+1} = M^{-1}r^{m+1} - \beta_{m+1}Mp^m \iff \boxed{p^{m+1} = z^{m+1} - \beta_{m+1}p^m}.$$

The algorithm for A

Define $C = M^2$.
Initialization. given X^0 , $r^0 = b - AX^0$, solve $Cz^0 = r^0$, $p^0 = z^0$.

While $m < Niter$ or $\|r^m\| \geq tol$, do

$$
\begin{aligned}
\alpha_m &= \frac{(z^m, r^m)}{(Ap^m, p^m)},\\
X^{m+1} &= X^m + \alpha_m p^m,\\
r^{m+1} &= r^m - \alpha_m A p^m,\\
\text{solve } Cz^{m+1} &= r^{m+1},\\
\beta_{m+1} &= \frac{(z^{m+1}, r^{m+1})}{(z^m, r^m)},\\
p^{m+1} &= z^{m+1} - \beta_{m+1} p^m.
\end{aligned}
$$

Now forget about M.

How to choose C

C must be chosen such that

1. $\tilde{A}$ is better conditioned than A to lessen the convergence factor,
2. C is easy to invert to solve $Cz^{m+1} = r^{m+1}$.

Use an iterative method such that $A = C - N$ with symmetric C. For instance it can be a symmetrized version of SOR, named SSOR, defined for $\omega \in (0, 2)$ by

$$C = \frac{1}{\omega(2-\omega)}(D - \omega E)D^{-1}(D - \omega F).$$

If A is symmetric definite positive, so is D and its coefficients are positive, then its square root $\sqrt{D}$ is defined naturally as the diagonal matrix of the square roots of the coefficients. Then

$$C = SS^T, \quad \text{with } S = \frac{1}{\sqrt{\omega(2-\omega)}}(D - \omega E)D^{-1/2},$$

yielding a natural Cholewski decomposition of C.

How to choose C

C must be chosen such that

1. $\tilde{A}$ is better conditioned than A to lessen the convergence factor,
2. C is easy to invert to solve $Cz^{m+1} = r^{m+1}$.

Use an iterative method such that $A = C - N$ with symmetric C.

$$C = SS^T, \quad \text{with } S = \frac{1}{\sqrt{\omega(2-\omega)}}(D - \omega E)D^{-1/2},$$

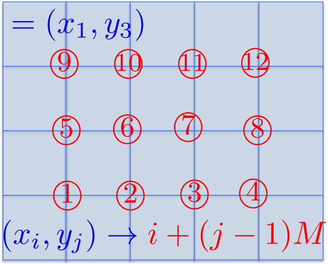
Another possibility is to use an incomplete Cholewski decomposition of A. Even if A is sparse, that is has many zeros, the process of LU or Cholewski decomposition is very expensive, since it creates non zero values.

Example : Matrix of finite differences in a rectangle

Poisson equation

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

$$1 \leq i \leq M, 1 \leq j \leq N$$



Numbering by line, $M = 4$, $N = 3$.

$(x_i, y_j) \rightarrow i + (j-1)M$. A vector of all unknowns X is created :

$$X = ((u_{1,1}, u_{1,2}, u_{1,N}), (u_{2,1}, u_{2,2}, u_{2,N}), \cdots (u_{M,1}, u_{M,2}, u_{M,N}))$$

with $X_{i+(j-1)*M} = u_{i,j}$.

Example : Matrix of finite differences in a square

If the equations are numbered the same way (equation $\#k$ is the equation at point k), the matrix is block-diagonal :

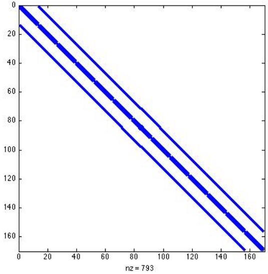
$$A = \frac{1}{h^2}\begin{pmatrix} B & -C & & 0_M & \\ -C & B & -C & & \\ & \ddots & \ddots & \ddots & \\ & & -C & B & -C \\ & 0_M & & -C & B \end{pmatrix} \quad (1)$$

$$C = I_M, \quad B = \begin{pmatrix} 4 & -1 & & 0 & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & 0 & & -1 & 4 \end{pmatrix}$$

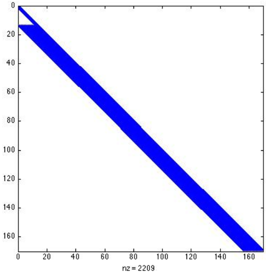
The righthand side is $b_{i+(j-1)*M} = f_{i,j}$, and the system takes the form $AZ = b$.

Cholewski decomposition of A

The block-Cholewski decomposition of A, $A = RR^T$, is block-bidiagonal, but the blocks are not tridiagonal as before, as the spy command of matlab can show, in the case where $M = 15$.

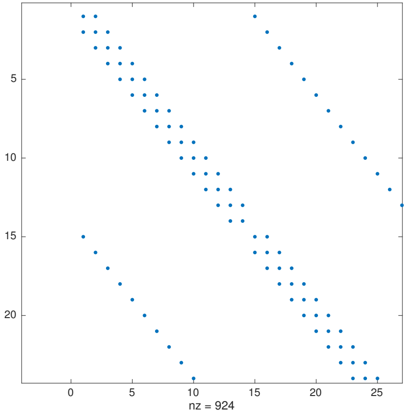


spy(A)

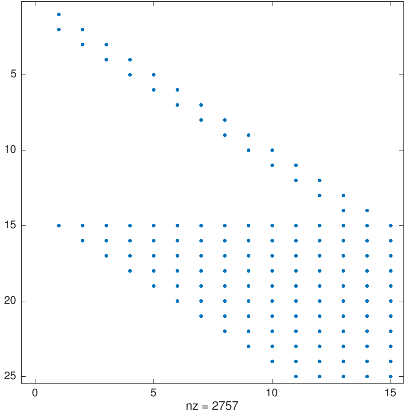


spy(R)

Cholewski decomposition of A, zoom



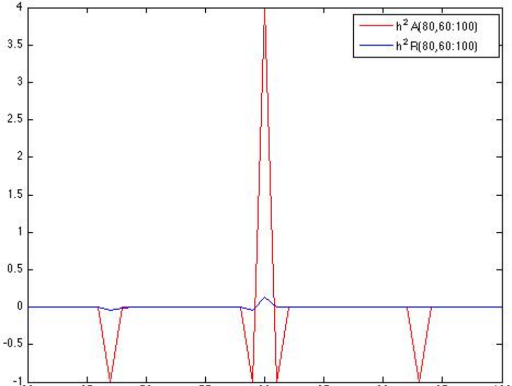
spy(A) zoom



spy(R) zoom

Cholewski decomposition of A, continue

However, if we look closely to the values of R between the main diagonales where A was non zero, we see that where the coefficients of A are zero, the coefficients of R are small. Therefore the incomplete Cholewski preconditioning computes only the values of R where the coefficient of A is not zero, and gain a lot of time.



Cholewski

$$A = RR^T;$$

$$\begin{pmatrix} A_{11} & \times & \times & \cdots & \times \\ A_{21} & A_{22} & \times & & \times \\ A_{k1} & \vdots & A_{kk} & & \times \\ \vdots & \vdots & A_{k+1,k} & \vdots & \\ A_{n1} & & A_{n,k} & & A_{nn} \end{pmatrix}$$

$$\begin{pmatrix} R_{11} & 0 & \cdots & \cdots & 0 \\ R_{21} & R_{22} & 0 & \cdots & 0 \\ R_{k1} & R_{k2} & R_{kk} & 0 & \cdots \\ \vdots & \vdots & \vdots & & \vdots \\ R_{n1} & R_{n2} & \vdots & R_{nn-1} & R_{nn} \end{pmatrix} \begin{pmatrix} R_{11} & R_{12}^T & \cdots & \cdots & R_{1n}^T \\ 0 & R_{22} & & & \\ 0 & 0 & R_{kk} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & R_{nn} \end{pmatrix}$$

Cholewski

Column 1

$$
\begin{aligned}
A_{11} &= R_{11}^2 \\
A_{21} &= R_{21} R_{11} \\
&\vdots \\
A_{n1} &= R_{n1} R_{11}
\end{aligned}
$$

Cholewski

Column 1

$$
\begin{aligned}
A_{11} &= R_{11}{}^{2} \\
A_{21} &= R_{21} R_{11} \\
&\vdots \\
A_{n1} &= R_{n1} R_{11}
\end{aligned}
$$

Cholewski

Column 1

Column 2

$$
\begin{aligned}
A_{22} &= R_{21}^2 + R_{22}^2 \\
A_{32} &= R_{31} R_{21} + R_{32} R_{22} \\
&\vdots \\
A_{n2} &= R_{n1} R_{21} + R_{n2} R_{22}
\end{aligned}
$$

Cholewski

Column 1

Column 2

$$
\begin{aligned}
A_{22} &= R_{21}^2 + {R_{22}}^2 \\
A_{32} &= R_{31} R_{21} + R_{32} R_{22} \\
&\vdots \\
A_{n2} &= R_{n1} R_{21} + R_{n2} R_{22}
\end{aligned}
$$

Cholewski

Column 1

Column 2

$$
\begin{aligned}
A_{22} &= R_{21}^2 + {R_{22}}^2 \\
A_{32} &= R_{31} R_{21} + R_{32} R_{22} \\
&\vdots \\
A_{n2} &= R_{n1} R_{21} + R_{n2} R_{22}
\end{aligned}
$$

Cholewski

Column 1

Column 2

Column 3

$$
\begin{aligned}
A_{kk} &= R_{k1}^2 + R_{k2}^2 + \cdots + R_{k-1k-1}^2 & +R_{kk}^2 \\
A_{k+1k} &= R_{k+11}R_{k1} + R_{k+12}R_{k2} + \cdots + R_{k+1k-1}R_{kk-1} & +R_{k+1k}R_{kk} \\
&\vdots \\
A_{nk} &= R_{n1}R_{k1} + R_{n2}R_{k2} + \cdots + R_{nk-1}R_{kk-1} & +R_{nk}R_{kk}
\end{aligned}
$$

Cholewski

Column 1

Column 2

Column 3

$$
\begin{aligned}
A_{kk} &= R_{k1}^2 + R_{k2}^2 + \cdots + R_{k-1k-1}^2 && + R_{kk}^2 \\
A_{k+1k} &= R_{k+11}R_{k1} + R_{k+12}R_{k2} + \cdots + R_{k+1k-1}R_{kk-1} && + R_{k+1k}R_{kk} \\
&\vdots \\
A_{nk} &= R_{n1}R_{k1} + R_{n2}R_{k2} + \cdots + R_{nk-1}R_{kk-1} && + R_{nk}R_{kk}
\end{aligned}
$$

Cholewski

Column 1

Column 2

Column 3

$$
\begin{aligned}
A_{kk} &= R_{k1}^2 + R_{k2}^2 + \cdots + R_{k-1k-1}^2 && + R_{kk}^2 \\
A_{k+1k} &= R_{k+11}R_{k1} + R_{k+12}R_{k2} + \cdots + R_{k+1k-1}R_{kk-1} && + R_{k+1k}R_{kk} \\
&\vdots \\
A_{nk} &= R_{n1}R_{k1} + R_{n2}R_{k2} + \cdots + R_{nk-1}R_{kk-1} && + R_{nk}R_{kk}
\end{aligned}
$$

Cholewski , matlab script

$$RR^T = A;$$

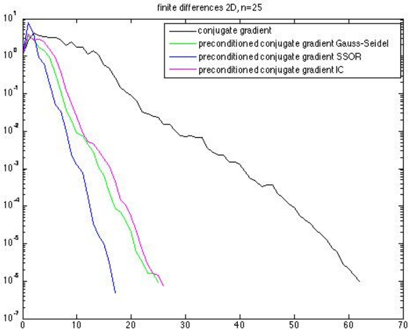
```
R=tril(A);
for k=1:nn
    R(k,k)=sqrt(R(k,k));
    R(k+1:nn,k)=R(k+1:nn,k)/R(k,k);
    for j=k+1:nn
        R(j:nn,j)=R(j:nn,j)-R(j:nn,k)*R(j,k);
    end
end
```

Incomplete Cholewski

```
RI=tril(A);
for k=1:nn
    RI(k,k)=sqrt(RI(k,k));
    for j=k+1:nn
        if RI(j,k) ~= 0
            RI(j,k)=RI(j,k)/RI(k,k);
        end
    end
    for j=k+1:nn
        for i=j:n
            if RI(i,j) ~= 0
                RI(i,j)=RI(i,j)-RI(i,k)*RI(j,k);
            end
        end
    end
end
```

Comparison

For the 2-D finite differences matrix and $n = 25$ internal points in each direction, we compare the convergence of the conjugate gradient and various preconditioning : Gauss-Seidel, SSOR with optimal parameter, and incomplete Cholewski. The gain even with $\omega = 1$ is striking. Furthermore Gauss-Seidel is comparable with Incomplete Cholewski.



Outline

The return of CG

$$X^0 \text{chosen}, \quad p^0 = r^0 = b - AX^0.$$

While $m < Niter$ or $\|r^m\| \geq tol$, do

$$\begin{aligned}
\alpha_m &= \frac{\|r^m\|^2}{(Ap^m, p^m)}, \\
X^{m+1} &= X^m + \alpha_m p^m, \\
r^{m+1} &= r^m - \alpha_m Ap^m, \\
\beta_{m+1} &= \frac{\|r^{m+1}\|^2}{\|r^m\|^2}, \\
p^{m+1} &= r^{m+1} - \beta_{m+1} p^m.
\end{aligned}$$

The return of CG

$$A \text{ sdp} \quad Ax = b \iff x = \text{Argmin } \frac{1}{2}\|Ay - b\|_2^2$$

Definition Krylov space $\mathcal{K}_m(A, r_0) = \text{vec}(r^0, Ar^0, \ldots, A^{m-1}r^0)$.

$$\|x^m - x\|_A = \inf_{y \in x^0 + \mathcal{K}_m} \|y - x\|_A,$$

$$\|x\|_A = \sqrt{x^T A x} = \sqrt{(Ax, x)}.$$

$$(r_i, r_j) = 0 \quad \text{and} \quad (Ap_i, p_j) = 0 \text{ for} i \neq j$$

Extension to non symmetric matrices

$$A \text{ sdp} \quad Ax = b \iff x = \text{Argmin } \frac{1}{2}\|Ay - b\|_2^2$$

rq : $x_0 = 0 \Rightarrow r_0 = -b$.

$$A \text{ non sdp} \quad x \approx x_m$$

$\mathcal{K}_m(A, r_0) = \text{vec}(r^0, Ar^0, \ldots, A^{m-1}r^0)$.

$$r^m = Ax^m - b, \quad \|r^m\| = \inf_{r \in \mathcal{K}_m} \|r\|.$$

We start with the determination of an orthogonal basis for $\mathcal{K}_m$.

Arnoldi algorithm

Let v_1 with $\|v_1\| = 1$.

```
for j=1:m do
h(i,j)=(A*v(j,:),v(i,:)) , i=1:j
w(j,:)=A*v(j,:)-sum(h(i,j)v(i,:)
h(j+1,j)=norm(w(j,:),2)
If h(j+1,j) == 0 stop
v(j+1,:)= w(j,:)/h(j+1,j)
```

Theorem If the algorithm goes through m, then $(w_1, \dots, w_m)$ is an orthonormal basis of $\mathcal{K}_m = \mathcal{L}(v_1, \dots, v_m)$.
The proof goes by recursion.

Arnoldi algorithm, continue

Define $V_m = [v_1, \dots, v_m]$ (matrix with column j equal to v_j),

$$\widetilde{H_m} = \begin{pmatrix} h_{11} & & \cdots & & h_{1m} \\ h_{21} & h_{22} & & \cdots & h_{2m} \\ 0 & h_{32} & \ddots & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & h_{mm-1} & h_{mm} \\ 0 & 0 & 0 & 0 & h_{m+1m} \end{pmatrix}$$

H_m is the $m \times m$ matrix obtained from the $(m+1) \times m$ matrix $\widetilde{H_m}$ by deleting the last row.

Proposition

$$AV_m = V_{m+1}\widetilde{H_m} = V_m H_m + w_m e_m^T, \quad V_m^T A V_m = H_M$$

Solving $Ax = b$, full orthogonalization method or FOM

Search for an approximate solution in $x_0 + \mathcal{K}_m(A, r_0)$ in the form $x_m = x_0 + V_m y$, and impose $r_m \perp \mathcal{K}_m(A, r_0)$. This is equivalent to $V_m^T r_m = 0$, which is written as

$$V_m^T A V_m y = V_m^T r_0 \text{ or } H_m y = \|r_0\| e_1.$$

The small system can be solved at each step using a direct method

FOM algorithm

```
function [X,R,H,Q]=FOM(A,b,x0);
% FOM full orthogonalization method
% [X,R,H,Q]=FOM(A,b,x0) computes the decomposition A=QHQ?, Q orthogonal
% and H upper Hessenberg, Q(:,1)=r/norm(r), using Arnoldi in order to
% solve the system Ax=b with the full orthogonalization method. X contains
% the iterates and R the residuals
n=length(A); X=x0;
r=b-A*x0; R=r; r0norm=norm(r);
Q(:,1)=r/r0norm;
for k=1:n
    v =A*Q(:,k);
    for j=1:k
        H(j,k)=Q(:,j)'*v; v=v-H(j,k)*Q(:,j);
    end
    e0=zeros(k,1); e0(1)=r0norm; % solve system
    y=H\e0; x= x0+Q*y;
    X=[X x];
    R=[R b-A*x];
    if k<n
        H(k+1,k)=norm(v); Q(:,k+1)=v/H(k+1,k);
    end
end
```

GMRES algorithm

Here we don't expect to find r_m orthogonal to $\mathcal{K}_m(A, r_0)$, but we minimize the residual in $\mathcal{K}_m(A, r_0)$, which is equivalent to the minimization of $J(y) = \|b - A(x_0 + V_m y)\|_2$ for y in $\mathbb{R}^m$, with $v_1 = r_0/\|r_0\|$. Use the Proposition to write

$$b - A(x_0 + V_m y) = r_0 - AV_m y = \|r_0\| v_1 - V_{m+1}\widetilde{H}_m y = V_{m+1}(\|r_0\| e_1 - \widetilde{H}_m y)$$

Since V_{m+1} is orthogonal, then

$$\|b - A(x_0 + V_m y)\| = \|\|r_0\| e_1 - \widetilde{H}_m y\|.$$

This small minimization problem can be solved by the Givens reflection method.

Theorem Let $A \in \mathbb{R}^n \times \mathbb{R}^n$ be invertible, $b \in \mathbb{R}^n$ and m be the degree of the minimal polynomial of A. Then GMRES applied to the linear system $Ax = b$ converges to the exact solution in at most m iterations.

Restarted GMRES

For reasons of storage cost, the GMRES algorithm is mostly used by restarting every M steps :
Compute $x_1, \cdots, x_M$.
If r_M is small enough, stop,
else restart with $x_0 = x_M$.