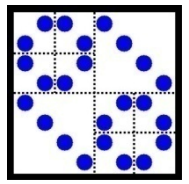


5.2.3 GMRES for general A

Consider small subspace U_m and determine optimal approximate solution for $A x = b$ in U_m. Hence x is of the form $x := U_m y$

$$\min_{x \in U_m} \|Ax - b\|_2 = \min_{y} \|A(U_m y) - b\|_2$$

Can be solved by normal equations: $(U_m^T A^T A U_m)\, y = U_m^T A^T b$

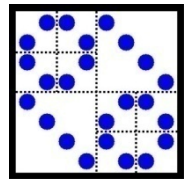
Try to find sequence of "good" subspaces $U_1 \rightarrow U_2 \rightarrow U_3 \rightarrow \ldots$

such that iteratively we can update the optimal solutions

$$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \ldots \; A^{-1}b$$

using mainly matrix-vector-products.

GMRES: Subspace



What subspace for fast convergence and easy computations?

$U_m := K_m(A,b) = \text{span}(b, Ab, \ldots, A^{m-1}b)$ Krylov space!

Problem: b, Ab, A^2b , … is a bad basis for this subspace!

So we need a first step to compute a good basis for U_m :

$$u_1 := b / \|b\|_2 ;$$

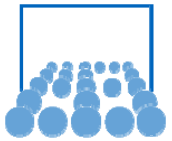
$$for \quad j = 2 : m$$

$$\tilde{u}_j := Au_{j-1} - \sum_{k=1}^{j-1} \left(u_k^T Au_{j-1}\right) u_k = Au_{j-1} - \sum_{k=1}^{j-1} h_{k,j-1} u_k ;$$

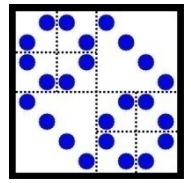
$$u_j := \tilde{u}_j / \|\tilde{u}_j\|_2 = \tilde{u}_j / h_{j,j-1} ;$$

Standard Ortho-gonalization Method:

Arnoldi method



Matrix form of Arnoldi ONB



U_m = span(b, Ab, …, A^{m-1}b) = span(u_1, u_2 , … , u_m) ONB

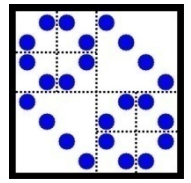
Write this Orthogonalization method in matrix form:

$$Au_{j-1} = \sum_{k=1}^{j-1} h_{k,j-1} u_k + \tilde{u}_j = \sum_{k=1}^{j} h_{k,j-1} u_k$$

$$A \cdot U_m = A\begin{pmatrix} u_1 & \dots & u_m \end{pmatrix} = \begin{pmatrix} u_1 & \dots & u_{m+1} \end{pmatrix} \cdot \tilde{H}_{m+1,m} = U_{m+1} \tilde{H}_{m+1,m}$$

$$\tilde{H}_{m+1,m} = \begin{pmatrix} h_{11} & \cdots & \cdots & h_{1m} \\ h_{21} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ & & \ddots & h_{mm} \\ 0 & & 0 & h_{m+1,m} \end{pmatrix}$$

GMRES: Minimization



This leads to the minimization problem

$$\min_{x \in U_m} \|Ax - b\| = \min_{y} \|AU_m y - b\| =$$

$$= \min_{y} \left\| U_{m+1} \tilde{H}_{m+1,m} y - \|b\| u_1 \right\| =$$

$$= \min_{y} \left\| U_{m+1} \left(\tilde{H}_{m+1,m} y - \|b\| e_1 \right) \right\| =$$

$$= \min_{y} \left\| \tilde{H}_{m+1,m} y - \|b\| e_1 \right\|$$

Because U_{m+1} is part of an orthogonal matrix.

GMRES: QR

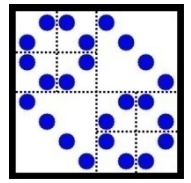
Use Givens matrices to compute a QR-factorization of the upper Hessenbergmatrix $H_{m+1,m}$

$$G_1 \cdot \begin{pmatrix} * & \cdots & * \\ * & \ddots & \vdots \\ & \ddots & * \\ & & * \end{pmatrix} = \begin{pmatrix} * & \cdots & * \\ 0 & \ddots & \vdots \\ & \ddots & * \\ & & * \end{pmatrix}$$

$$G_m \ldots G_2 G_1 \cdot \tilde{H}_{m+1,m} = Q \cdot \tilde{H}_{m+1,m} = \begin{pmatrix} R_m \\ 0 \end{pmatrix}$$

QR via Givens matrices is a simplified version of QR via Householder matrices

GMRES



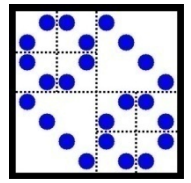
$$\min_{x \in K_m} \left\| Ax - b \right\| = \min_{y} \left\| \tilde{H}_{m+1,m} y - \| b \| e_1 \right\| =$$

$$= \min_{y} \left\| Q^T R y - \| b \| e_1 \right\| = \min_{y} \left\| R y - \| b \| Q \; e_1 \right\| =$$

$$= \min_{y} \left\| \begin{pmatrix} R_m \\ 0 \end{pmatrix} y - \tilde{b} \right\| = \min_{y} \left\| \begin{pmatrix} R_m y - \tilde{b}_m \\ -\tilde{\beta}_m \end{pmatrix} \right\|$$

Solution: $R_m y = \tilde{b}_m , \quad x_m = U_m y$

GMRES Algorithm

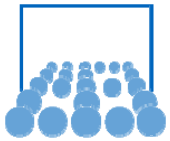


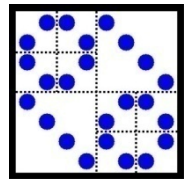
GMRES is a clever implementation of these sequence of Least Squares problems.

Computing step by step for increasing m
- next Au_m
- enlarged $H_{m+1,m}$ by Arnoldi orthogonalization (gives new u_{m+1})
- next Givens matrix G_m
- update triangular solves to get next y_m and x_m.

Disadvantage: large Hessenberg matrices!

Therefore, use restarted GMRES, e.g. GMRES(20)





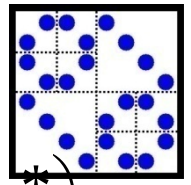
Start: $b, \quad u_1 := b/\|b\|$

$$h_{2,1}u_2 = Au_1 - h_{1,1}u_1, \qquad H_{2,1} = \begin{pmatrix} h_{1,1} \\ h_{2,1} \end{pmatrix}, \qquad G_1 H_{2,1} = \begin{pmatrix} v_{1,1} \\ 0 \end{pmatrix},$$

$$\|Ax - b\| = \|AU_1 r_1 - b\| = \|U_2 H_{2,1} r_1 - U_2 \|b\| e_1\| =$$

$$= \left\| G_1{}^T \begin{pmatrix} v_{1,1} \\ 0 \end{pmatrix} r_1 - \|b\| e_1 \right\| = \left\| \begin{pmatrix} v_{1,1} \\ 0 \end{pmatrix} r_1 - \|b\| G_1 e_1 \right\|$$

$$r_1 = \left(G_1 e_1 \|b\|\right)_1 / v_{1,1}, \qquad x_1 = u_1 r_1 = U_1 r_1$$



Second step:

$$h_{3,2}u_3 = Au_2 - h_{2,2}u_2 - h_{1,2}u_1, \qquad H_{3,2} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ 0 & h_{32} \end{pmatrix}, \qquad G_1 H_{3,2} = \begin{pmatrix} v_1 & * \\ 0 & * \\ 0 & * \end{pmatrix},$$

$$G_2 G_1 H_{3,2} = \begin{pmatrix} v_{1,1} & v_{1,2} \\ 0 & v_{2,2} \\ 0 & 0 \end{pmatrix}$$

$$\|Ax - b\| = \|AU_2 r_2 - b\| = \|U_3 H_{3,2} r_2 - U_3 \|b\| e_1\| =$$

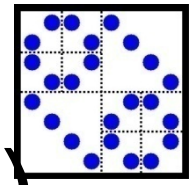
$$= \left\| G_1^T G_2^T \begin{pmatrix} v_{1,1} & v_{1,2} \\ 0 & v_{2,2} \\ 0 & 0 \end{pmatrix} r_2 - \|b\| e_1 \right\| = \left\| \begin{pmatrix} v_{1,1} & v_{1,2} \\ 0 & v_{2,2} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} r_{2,1} \\ r_{2,2} \end{pmatrix} - \|b\| G_2 G_1 e_1 \right\|$$

$$r_{2,2} = \left(G_2 G_1 e_1 \|b\|\right)_2 / v_{2,2}, \qquad r_{2,1} = \left(\left(G_2 G_1 e_1 \|b\|\right)_1 - v_{1,2} r_{2,2}\right) / v_{1,1}$$

$$x_2 = U_2 r_2 = u_1 r_{2,1} + u_2 r_{2,2}$$

and so on …

GMRES Error (A diagonalisable)



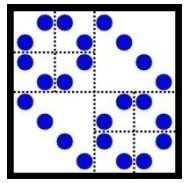
$$\|r_m\|_2 = \|Ax_m - b\| = \min_{x \in U_m} \|Ax - b\| =$$

$$= \min_{\alpha_1,\ldots,\alpha_m} \left\| A\left(\sum_{j=0}^{m-1} \alpha_j A^j b \right) - b \right\| = \min_{p_{m-1}} \|Ap_{m-1}(A)b - b\|$$

$$= \min_{q_m(0)=1} \|q_m(A) \cdot b\| = \min_{q_m(0)=1} \|V q_m(\Lambda) V^{-1} \cdot b\| =$$

$$= \min_{q_m(0)=1} \|V \cdot diag(q_m(\lambda_j)) \cdot V^{-1} \cdot b\| \leq$$

$$\leq cond(V) \cdot \min_{q_m(0)=1} \left[\max_{j=1,\ldots,n} |q_m(\lambda_j)| \right] \cdot \|r_0\|$$



Iterative methods: Survey

Basic iterative methods:

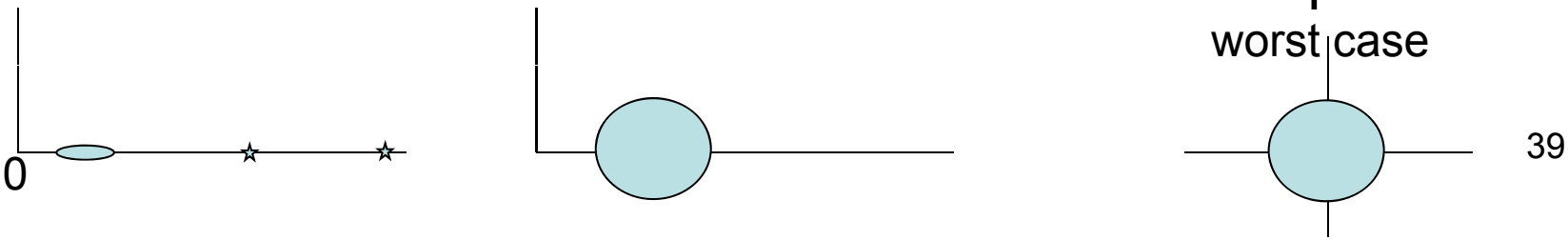
spd: pcg = preconditioned conjugate gradient method

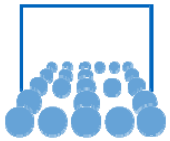
Symmetric indefinite: MINRES, SYMMLQ, (pcg on normal equations)

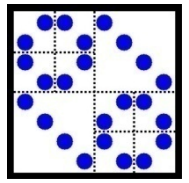
General: GMRES (optimal like cg, but expensive)
BiCG, BiCGSTAB, CGS (not optimal, but cheap)
QMR (quasi optimal, cheap)

Essential for convergence: position of eigenvalues (singular values)

Fast convergence for: well conditioned problems or matrices with clustered spectrum



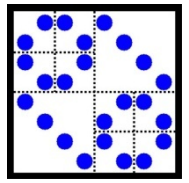




MATLAB example: comparison
stationary.m
pcg, GMRES

→ Improve convergence → preconditioning

5.3 Preconditioning



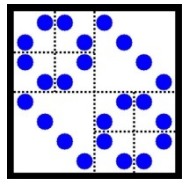
Direct solvers:
Sequential, loosing sparsity

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:

$$Ax = b \quad \leftrightarrow \quad M A (K y) = M b \text{ and } K y = x$$

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

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

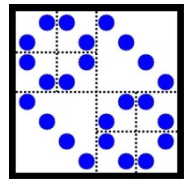
That is exactly a condition for a good preconditioner:

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 G-S preconditioner.

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

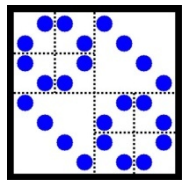
As convex combination of old and new iterate (Jacobi or GS)

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

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

Incomplete LU factorization → ILU

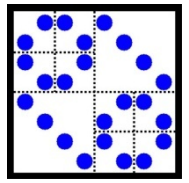
Reduce in the FOR-loops the indices to the indices with
- allowed pattern, e.g. ILU(0) for pattern of A
- values that are not to small, ILUT for ILU with treshold

Leads to approximate LU factorization

$A = L\,U + R$, preconditioner $M = L\,U$

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.

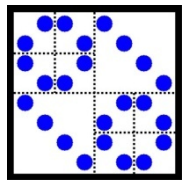
The resulting preconditioners are given by triangular matrices L, that have to be solved in each iteration step: $L^{-1}x_k$!
Strongly sequential!

Jacobi easy to parallelize, but slow convergence.

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

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

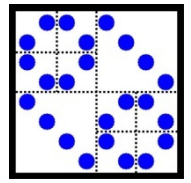
Examples: For GS is (i) and (iii) OK, but not (ii)

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

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

Note that preconditioners also have to be well defined!
Zero on diagonals???

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.

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**

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

Solution: Normalized Chebyshev polynomials

Advantage of polynomial preconditioner: better in parallel

Disadvantage: Not-optimal approximation in Krylov subspace