Plan général

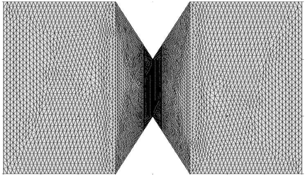
Motivation

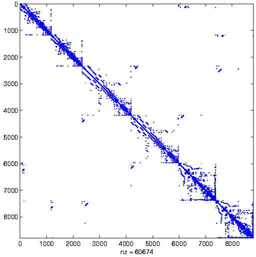
Classical direct methods for solving $Ax = b$ (lectures 1, 2):

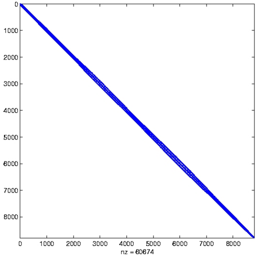
- Designed for dense matrices;
- Require $O(n^3)$ arithmetic operations (except matrices with structure, e.g. band matrices)
- Require (half of) A stored.

Ill-suited, or sub-optimal, for many applications:

- A may be very large ($O(n^2)$ storage, $O(n^3)$ computing work, either potentially prohibitive).
- Frequent situations involving large, sparse sparse matrices
  - $\rightarrow$ ODE/PDE discretization by e.g. finite elements/differences;
  - $\rightarrow$ Factorization (a) fills initially-zero entries, (b) involves unnecessary operations on zero entries.





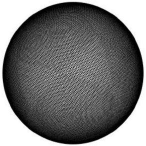


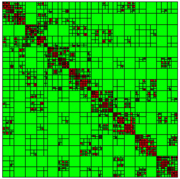
(after mesh numbering)

Motivation

Sometimes: A dense but has accurate enough low-rank approximation

- e.g. fast-multipole or hierarchical-matrix for large boundary element models;





- e.g. dense ill-conditioned matrices with *numerical rank* $\ll n$ (see lecture 6).

Motivation

Alternative approach: iterative methods:

- Define *approximating sequences* $x_0, x_1, x_2, \ldots \to x$ solving $Ax = b$
- In general, limiting process: $\boxed{x = \lim_{k\to\infty} x_k}$.
- Iterate a rule defining next iterate x_{k+1} from current iterate x_k.
- *Direct* vs. *iterative* algorithms:
  - $\to$ Direct methods (e.g. LU, $\text{LDL}^\text{T}\ldots$) give exact result within finitely many operations;
  - $\to$ Iterative methods give exact result as a limit (potentially infinitely many operations)
- In practice, happy to stop at K-th iteration giving a "close enough" approximation x_K.
- Ideal practical goal: $\|x_K - x\|/\|x\| \leq \varepsilon$ for given tolerance ε (impractical since x unknown!)
- Feasible practical goal: $\|b - Ax_K\|/\|b\| \leq \varepsilon$

Expected advantages of iterative methods:

- Availability of (too-large, too-expensive. . .) A not needed; instead, use evaluations $x \mapsto Ax$.
- Take full advantage of matrix sparsity (if present).
- Allows working with $O(n)$ storage.

This course (lectures 3, 4): three kinds of iterative methods for linear systems:

- Classical methods based on matrix splitting (briefly);
- Conjugate gradient method for SPD systems;
- Generalized minimum residuals (GMRES) for general square systems.

Splitting (fixed-point) methods

Basic idea:

- Set $A = A_1 - A_2$ with A_1 chosen as "easily invertible";
- Define iterations $A_1 x_{k+1} = A_2 x_k + b$.
- Algebraically (not computationally!):

$$x_{k+1} = A_1^{-1}\big(b + A_2 x_k\big) = A_1^{-1}\big(b + A_2 A_1^{-1}(b + A_2 x_{k-1})\big) \ldots = A_1^{-1}\Big\{ \sum_{i=0}^{k} (A_2 A_1^{-1})^i b \Big\}$$

- Sufficient convergence condition: $\boxed{\|A_2 A_1^{-1}\| < 1}$, since

$$\Big\| \sum_{i=0}^{k} (A_2 A_1^{-1})^i b \Big\| \leq \Big\{ \sum_{i=0}^{k} \|A_2 A_1^{-1}\|^i \Big\} \|b\| \overset{\lim\limits_{k\to\infty}}{\leq} \frac{\|b\|}{1 - \|A_2 A_1^{-1}\|}$$

- Fixed-point iteration interpretation:

$$x_{k+1} = F(x_k), \qquad F(x) := A_1^{-1}\big(b + A_2 x\big)$$

Iterations $A_1 x_{k+1} = A_2 x_k + b$ converge for any initial guess x_0 iff $\rho(A_2 A_1^{-1}) < 1$ (guarantees $x \mapsto F(x)$ is contracting).

Some well-known splitting-based algorithms.

Split A according to $\boxed{A = D - L - U}$ (diagonal − lower triangle − upper triangle);

- *Jacobi iterations:* $A_1 = D$, $A_2 = L + U$.
- *Gauss-Seidel iterations:* $A_1 = D - L$, $A_2 = U$.
- *Successive over-relaxation (SOR) iterations:* $A_1 = D - \eta L$, $A_2 = \eta U + (\eta - 1)D$ ($\eta \in \mathbb{R}$, $\eta \neq 0$),

$$\big[D - \eta L\big]x_{k+1} = \big[\eta U + (1-\eta)D\big]x_k + \eta b \qquad (x_0\text{: user-chosen initial guess}).$$

→ Gauss-Seidel as special case of SOR ($\eta = 1$).
→ Relaxation parameter η (tunable): helps ensure convergence or enhance convergence rates
→ SOR iterations do not converge for any A, even with best η.

Some properties of splitting-based iterations:

- If A diagonally dominant ($|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for each $1 \leq i \leq n$), Jacobi converges for any x_0.
- If A real SPD, Gauss-Seidel converges for any x_0.
- Necessary convergence condition for SOR: $0 < \eta < 2$ (otherwise, SOR diverges for some x_0).
- If A real SPD, SOR converges for any $0 < \eta < 2$ and any x_0.

(proofs use Householder-John theorem: if A and $A - B - B^{\mathsf{T}}$ are real SPD, then $\rho\big((A-B)^{-1}B\big) < 1$.

General observations:

- Splitting-based iterations very useful for specific classes of problems;
- Convergence not guaranteed for general A;
- Some methods rely on triangular systems, hence unsuitable if A large and dense.

By contrast: CGM and GMRES (i) always converge, and (ii) do not need matrix storage.

Alternative: iterative solution algorithms based on quadratic minimization. This course:

- General observations, inadequacy of steepest-descent
- SPD systems and conjugate gradient method (CGM)
- Krylov spaces
- General square systems and generalized minimal residuals (GMRES, Krylov-based)
- Krylov space interpretation and convergence of CGM

Quadratic minimization viewpoint for SPD systems (uniquely solvable using LDL^{T}/Cholesky).

- Let $A \in \mathbb{R}^{n\times n}$ SPD, $b \in \mathbb{R}^n$. Quadratic minimization problem

  $$\boxed{\min_{x\in\mathbb{R}^n} J(x) := \tfrac{1}{2}x^{\text{T}}Ax - x^{\text{T}}b + c}$$

  has a unique solution $x^\star$ (A SPD $\implies$ $J(x)$ strictly convex).
- Solution found from 1st order necessary condition

  $$\nabla J(x^\star) = 0, \quad \text{i.e.} \quad Ax^\star = b$$
- Since A SPD: eigenvalues $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n > 0$, condition mumber $\kappa_2(A) = \lambda_1/\lambda_n$.
- Since A SPD: energy norm and scalar product

  $$\boxed{\|x\|_A^2 := x^{\text{T}}Ax} \quad \lambda_n\|x\|_2 \leq \|x\|_A \leq \lambda_1\|x\|_2, \qquad \boxed{(y,x)_A := y^{\text{T}}Ax}.$$
- With these definitions:

  $$J(x) = \tfrac{1}{2}\big\|x - A^{-1}b\big\|_A^2 + \text{cste}$$

  (will be used to interpret/assess CGM).

SPD systems and quadratic minimization

General form of unconstrained gradient-based minimization (data: A, b and tolerance ε):

- Initialization: $x = x_0$
- Iterations $k = 0, 1, 2, \ldots$ until convergence:
  - **(i)** Choose *descent direction* $p_k \in \mathbb{R}^n$ (must verify $p_k^{\mathsf{T}} \nabla J(x_k) < 0$);
  - **(ii)** 1D minimization along descent direction (line-search), here in closed form:

    $$t_k = \arg\min_{t \geq 0} \Big\{ j(t) := J(x_k + tp_k) \Big\} = \frac{p_k^{\mathsf{T}} r_k}{p_k^{\mathsf{T}} A p_k} \qquad r_k := b - Ax_k\text{: current } \textit{residual}.$$

  - **(iii)** Solution update:

    $$x_{k+1} := x_k + t_k p_k.$$

- Convergence test, e.g. on relative residual: is $\|b - Ax\|/\|b\| \leq \varepsilon$ reached?

Crucial ingredient: method defining descent direction p_k for each iteration.

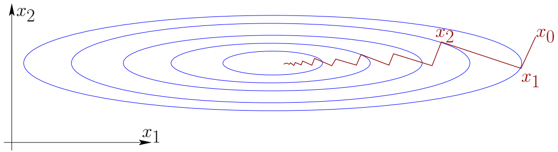
SPD systems and quadratic minimization

- Natural choice: $p_k := -\nabla J(x_k)$ (steepest, i.e. initially-fastest, descent direction from x_k)
- However, turns out to be inefficient:

$$\boxed{e_k \leq \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1} \right)^k e_0, \qquad \text{with} \quad e_k := \|x_k - x^\star\|_A}$$

(proof is quite lengthy)

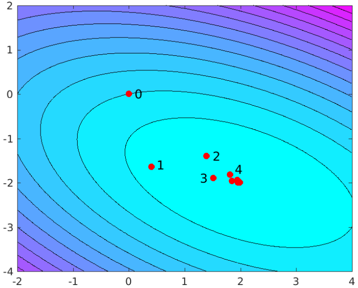
Say $\kappa_2(A) = 100$, then $e_k \leq (99/101)^k e_0$, takes 100s of iterations to reach $e_k \leq 10^{-3} e_0$



SPD systems and quadratic minimization

Example: $\boxed{Ax = b \qquad \text{with} \quad A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, \; b = \begin{Bmatrix} -2 \\ 8 \end{Bmatrix}}$ $(\lambda_1 = 7,\ \lambda_2 = 2, \kappa_2(A) = 7/2)$.

Solution $x^\star = \{2,\ -2\}^\mathsf{T}$: 19 (resp. 43) iterations for $\|b - Ax\|/\|b\| \leq 10^{-4}$ (resp. 10^{-10}).
Theoretical rate $e_k \leq (5/9)^k e_0$, predicts 16 (resp. 39) iterations.



Congugate gradient heuristics.

- Each p_k A-orthogonal ("conjugate") to all previous directions:

  $$\big(p_j, p_k\big)_A = 0 \quad \text{for all } j < k \, .$$

  Heuristic idea: each new p_k in an unexplored direction of $\mathbb{R}^n$
- Algorithm derivation: try $x_0 = 0$ (so $r_0 = b$) and $p_0 = r_0$ ($k=0$), then

  $$p_k = r_k + \beta_k p_{k-1} + \beta_{k-1} p_{k-2} + \ldots + \beta_1 p_0 \qquad (r_k := b - Ax_k = -\nabla J(x_k) : \quad \text{residual}),$$

  p_k: "corrected" steepest-descent made A-orthogonal to all previous ones.
- Induction on k then gives $\beta_1 = \ldots = \beta_{k-1} = 0$ and

$$\boxed{x_{k+1} = x_k + \alpha_{k+1} p_k, \qquad p_k = r_k + \beta_k p_{k-1}, \qquad r_{k+1} = r_k - \alpha_k A p_k} \tag{3}$$

$$\alpha_{k+1} = \frac{r_{k-1}^{\mathsf{T}} r_{k-1}}{(p_{k-1}, p_{k-1})_A}, \qquad \beta_k = \frac{r_k^{\mathsf{T}} r_k}{r_{k-1}^{\mathsf{T}} r_{k-1}}. \tag{4}$$

(see details in lecture notes)

Conjugate gradient method (CGM)

Algorithm 4 Conjugate gradient algorithm for SPD problems (explanatory form)

$A \in \mathbb{R}^{n\times n}$ SPD and $b \in \mathbb{R}^n$ (problem data)
$x_0 = 0,\ r_0 = b,\ p_0 = r_0$ (initialization)
for $k = 1, 2, \ldots$ **do**
    $q_{k-1} = Ap_{k-1}$ (matrix-vector product)
    $\alpha_k = \dfrac{r_{k-1}^{\mathsf{T}} r_{k-1}}{p_{k-1}^{\mathsf{T}} q_{k-1}}$ (optimal step)
    $x_k = x_{k-1} + \alpha_k p_{k-1}$ (solution update)
    $r_k = r_{k-1} - \alpha_k q_{k-1}$ (residual update)
    $\beta_k = \dfrac{r_k^{\mathsf{T}} r_k}{r_{k-1}^{\mathsf{T}} r_{k-1}}$ (conjugacy coefficient)
    $p_k = r_k + \beta_k p_{k-1}$ (descent direction for next iteration)
    Stop if convergence, set $x = x_k$
end for

Optimality property of CG iterates

Residual $r_{k+1} := b - Ax_k$ orthogonal to all previous directions: $p_j^{\mathsf{T}} r_{k+1} = 0$ $(0 \le j \le k)$. Therefore:

- x_{k+1} minimizes $J(x)$ over $\mathrm{span}\big(p_0, \ldots, p_k\big)$ (not just along current p_k).
- CGM must converge in at most n iterations (since $\mathrm{span}\big(p_0, \ldots, p_{n-1}\big) = \mathbb{R}^n$ then)

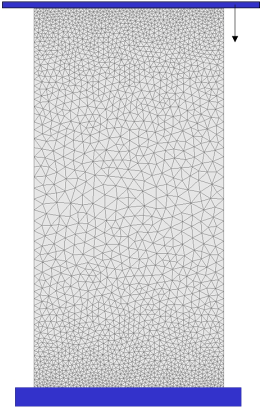
Conjugate gradient method (CGM)

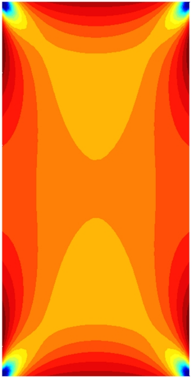
Algorithm 5 Conjugate gradient algorithm for SPD problems (practical form)

$A \in \mathbb{R}^{n\times n}$ SPD and $b \in \mathbb{R}^n$ (problem data)
$x = 0,\ r = b,\ p = r$ (initialization)
for $k = 1, 2, \dots$ **do**
    $q = Ap$ (matrix-vector product)
    $\gamma = p^{\mathsf{T}} q$ (auxiliary coefficient)
    $\alpha = (r^{\mathsf{T}} r)/\gamma$ (optimal step)
    $x = x + \alpha p$ (solution update)
    $r = r - \alpha q$ (residual update)
    $\beta = (r^{\mathsf{T}} r)/\alpha\gamma$ (conjugacy coefficient)
    $p = r + \beta p$ (descent direction for next iteration)
    Stop if convergence, return solution x
end for

- One matrix-vector product / iteration (usually main computing effort)
- If prefer some $x_0 \neq 0$, set $x = x' + x_0$ ($Ax' = b - Ax_0$), start from $x'_0 = 0$

Example: CGM in finite element analysis (plate in compression)

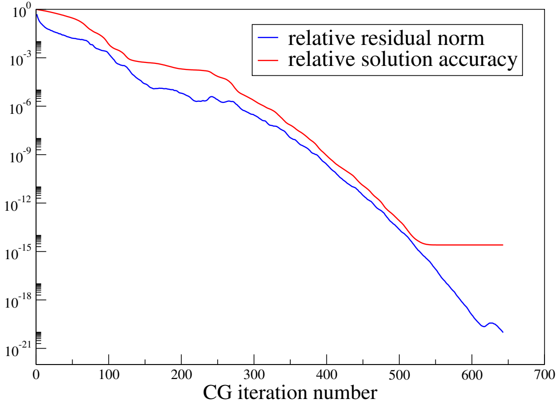




- 4410 DOFs, stiffness matrix K is SPD, sparse, banded with half-bandwidth 147 (after DOF renumbering!)
- nonzeros in K (upper part only): 32263 (net), 386015 (within profile), 641802 (within bandwidth)

MB, A. Frangi, C. Rey, The finite element method in solid mechanics (Mc Graw Hill Education, 2014)

Example: CGM in finite element analysis (plate in compression)



Relative solution accuracy 10^{-6} (relative to direct solver) reached after about 300 iterations (7% of matrix size)

Revisit splitting methods:

- Consider (simplistic) splitting $A = I + (A - I)$ and iterate:
  $x_{k+1} + (A-I)x_k = b$ i.e. $x_{k+1} = b + x_k - Ax_k, \quad x_0 = 0,\ x_1 = b$
  (method not recommended for applications: converges only if $\rho(A-I) < 1$!)
- By induction: $\boxed{x_k \in \text{span}\big(b, Ab, A^2b, \ldots, A^{k-1}b\big), \quad k = 1, 2, \ldots}$
- Similar observations apply to other splitting methods.

Definition: Krylov subspace

Let $A \in \mathbb{K}^{n\times n}$, $b \in \mathbb{K}^n$. For any $k \le n$, define the Krylov subspace $\mathcal{K}_k = \mathcal{K}_k(A, b)$ as

$$\mathcal{K}_k(A, b) := \text{span}\big(b, Ab, A^2b, \ldots, A^{k-1}b\big).$$

- Krylov subspaces are nested: $k < \ell \implies \mathcal{K}_k \subset \mathcal{K}_\ell$.
- Krylov basis vectors generated by matrix-vector products $w \mapsto Aw$;
- $w \in \mathcal{K}_k(A, b) \iff w = p(A)b$ for some polynomial p of degree $k-1$.

Krylov spaces integral to many iterative solution methods (link matrix equations to matrix polynomials)

- Concept of Krylov space underpins many other iterative methods (in particular GMRES)
- Conjugate gradient method (CGM) introduced (previous lecture) following heuristic argument
  Concept of Krylov space will porovide insight into CGM (e.g. convergence rate estimates)

Characteristic polynomial, minimal polynomial

- *Characteristic polynomial* p_A of $A \in \mathbb{K}^{n\times n}$: $\boxed{p_A(\lambda) = \det(A - \lambda I)}$ (p_A has degree n)
- $P_A(A) = 0$ for any $A \in \mathbb{K}^{n\times n}$ *(Cayley-Hamilton theorem)*
- *Minimal polynomial* q_A of $A \in \mathbb{K}^{n\times n}$: polynomial q_A of smallest degree such that $q_A(A) = 0$.

Krylov subspace sequences may stagnate:

- One may have $\mathcal{K}_\ell(A, b) = \mathcal{K}_{\ell+1}(A, b)$ for some $\ell < n$.
- Krylov subspaces then stationary: $\mathcal{K}_\ell(A, b) = \mathcal{K}_{\ell+1}(A, b) = \mathcal{K}_{\ell+2}(A, b) = \ldots = \mathcal{K}_n(A, b)$.
- Grade of b with respect to A: smallest ℓ such that $\mathcal{K}_\ell(A, b) = \mathcal{K}_{\ell+1}(A, b)$
- Exists $q_{A;b}$ such that $q_{A;b}(A)b = 0$ (*minimal polynomial of b with respect to A*), with

  $$\ell = \text{degree}(q_{A;b}) \leq \text{degree}(q_A).$$

We still need/want iterative solvers for general square invertible systems. Krylov spaces will help!

We still need/want iterative solvers for general square invertible systems. Krylov spaces will help!

- Generalized Minimal RESiduals (GMRES) for $Ax = b$, $A \in \mathbb{K}^{n\times n}$ any invertible matrix
- Basic idea:

$$\boxed{x_k = \underset{x\in\mathcal{K}_k(A,b)}{\arg\min} \; \|b - Ax\|_2^2} \qquad (x_1 \in \mathcal{K}_1(A,b),\; x_2 \in \mathcal{K}_2(A,b) \supset \mathcal{K}_1(A,b) \ldots)$$

Iterates (i) minimize system residual, (ii) are sought in Krylov spaces of increasing dimension

- Naive implementation: set

$$x_k = \alpha_0 b + \alpha_1 Ab + \ldots + \alpha_{k-1} A^{k-1} b \qquad y_k := \{\alpha_0, \ldots, \alpha_{k-1}\}^{\mathrm{H}} \text{ unknown)}$$

Define Krylov matrix $K_k := \big[\, b,\ Ab,\ \ldots A^{k-1}b \,\big] \in \mathbb{K}^{n\times k}$, then:

$$y_k := \underset{y\in\mathbb{K}^k}{\arg\min} \|b - (AK_k)y_k\|_2^2 \qquad \text{(least-squares problem of size } n\times k\text{)}$$

- Practical issues:
  - $\to$ Krylov vectors $b, Ab, A^2b \ldots$ increasingly collinear (converge to eigenvector for $\lambda_1(A)$)
    Makes GMRES potentially ill-conditioned
  - $\to$ Least-squares problem for k-th iteration: QR factorization (say) needs $O(nk^2)$ operations.

GMRES

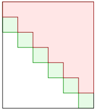
- Practical issues:
  - $\rightarrow$ Krylov vectors $b, Ab, A^2b \ldots$ increasingly collinear (converge to eigenvector for $\lambda_1(A)$)
    Makes GMRES potentially ill-conditioned $\Longrightarrow$ orthogonalization
  - $\rightarrow$ Least-squares problem for k-th iteration: QR factorization (say) needs $O(nk^2)$ operations
    $\Longrightarrow$ decompose A, recursive least squares problems

  Both issues solved by (recursive) **Arnoldi iteration.**

- Principle: construct orthonormal vectors $q_1, q_2, \ldots \in \mathbb{K}^n$ such that for each $k = 1, 2, \ldots$

$$\boxed{AQ_k = Q_{k+1}H_k}, \quad \text{i.e.} \quad A[q_1 \ \ldots \ q_k] = [q_1 \ \ldots \ q_k, q_{k+1}]H_k, \tag{5}$$

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \ldots & h_{1k} \\ h_{21} & h_{22} & & \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{kk} \\ 0 & & & h_{k+1,k} \end{bmatrix} \in \mathbb{K}^{(k+1)\times k} \quad \textit{(upper Hessenberg matrix)} \tag{6}$$



Arnoldi iterations

Column-by-column enforcement of equality

$$A[q_1 \; \ldots \; q_k] = [q_1 \; \ldots \; q_k, q_{k+1}] \begin{bmatrix} h_{11} & h_{12} & \ldots & h_{1k} \\ h_{21} & h_{22} & & \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{kk} \\ 0 & & & h_{k+1,k} \end{bmatrix}$$

q_1 arbitrary unit vector, then enforce orthonormality of $q_1, q_2, q_3, \ldots$

$$\begin{aligned}
&\text{column 1:} \quad Aq_1 = h_{11}q_1 + h_{21}q_2 && \Longrightarrow h_{11}, h_{21}, q_2 \\
& && \quad (q_1^{\mathrm{H}}q_2 = 0, \; \|q_2\| = 1) \\
&\text{column 2:} \quad Aq_2 = h_{12}q_1 + h_{22}q_2 + h_{32}q_3 && \Longrightarrow h_{12}, h_{22}, h_{32}, q_3 \\
& && \quad (q_1^{\mathrm{H}}q_3 = q_2^{\mathrm{H}}q_3 = 0, \; \|q_3\| = 1) \\
&\text{column 3:} \quad Aq_3 = h_{13}q_1 + h_{23}q_2 + h_{33}q_3 + h_{43}q_4 && \Longrightarrow h_{13}, h_{23}, h_{33}, h_{43}, q_4 \\
& && \quad (q_1^{\mathrm{H}}q_4 = q_2^{\mathrm{H}}q_4 = q_3^{\mathrm{H}}q_4 = 0, \; \|q_4\| = 1) \ldots
\end{aligned}$$

By induction: $q_k \in \text{span}(q_1, Aq_1, \ldots, A^{k-1}q_1)$ for $k = 1, 2, 3 \ldots$

$$\boxed{\text{span}(q_1, q_2, \ldots, q_k) = \text{span}(q_1, Aq_1, \ldots, A^{k-1}q_1) = \mathcal{K}_k(A, b)}$$

Arnoldi iterations: algorithm

Algorithm 6 Arnoldi iterations

```
A ∈ 𝕂^{n×n} and b ∈ 𝕂^n                          (problem data)
q_1 = b/‖b‖                                       (initialization)
for k = 1, 2, ... do
    q_{k+1} = A q_k                               (new Krylov vector: initialize q_{k+1})
    for j = 1 to k do
        h_{jk} = q_j^H q_{k+1}                    (entry in k-th column of H_k)
        q_{k+1} = q_{k+1} − h_{jk} q_j            (update (not yet normalized) vector q_{k+1})
    end for
    h_{k+1,k} = ‖q_{k+1}‖                         (last entry in k-th column of H_k)
    q_{k+1} = q_{k+1}/‖h_{k+1,k}‖                 (normalize q_{k+1})
end for
```

Note: to solve $Ax = b$, Arnoldi started with $q_1 = b$. Then:

- $\text{span}(q_1, Aq_1, \ldots, A^{k-1}q_1) = \text{span}(q_1, q_2, \ldots, q_k) = \mathcal{K}_k(A, b)$ for $k = 1, 2, 3 \ldots$
- allows to seek x_k as $x_k = Q_k y_k$

Apply Arnoldi iterations to GMRES

- Set $x_k = Q_k y_k$ for each $k = 1, 2, 3 \ldots$
- Use $AQ_k = Q_{k+1} H_k$ in residual $\|b - Ax_k\|_2$:

$$\begin{aligned}
\|b - Ax_k\|_2^2 &= \|b - AQ_k y_k\|_2^2 \\
&= \|b - Q_{k+1} H_k y_k\|_2^2 \\
&= \|Q_{k+1}^{\mathrm{H}} b - H_k y_k\|_2^2 \\
&= \|\beta e_1 - H_k y_k\|_2^2 \quad (\beta := \|b\|,\ q_1 = b/\beta) \implies \boxed{y_k = \arg\min_{y \in \mathbb{K}^k} \|\beta e_1 - H_k y\|_2^2}
\end{aligned}$$

- $H_k \in \mathbb{K}^{(k+1)\times k}$ (compare with $AK_k \in \mathbb{K}^{n\times k}$)
- step k of GMRES requires q_{k+1} and k-th column of H_k, i.e. one Arnoldi iteration

Algorithm 7 GMRES algorithm with Arnoldi iterations

1: $A \in \mathbb{K}^{n\times n}$ and $b \in \mathbb{K}^n$ (problem data)
2: $x = 0$, $\beta = \|b\|$, $q_1 = b/\beta$ (initialization)
3: **for** $k = 1, 2, \ldots$ **do**
4: Step k of Arnoldi iteration (see Algorithm 6)
5: Find $y \in \mathbb{K}^k$, $\|\beta e_1 - H_k y\|_2^2 \to \min$ (least squares problem)
6: $x = Q_k y$ (current solution)
7: **Stop** if convergence, return x
8: **end for**

GMRES

- Initialization: $x = 0$, $\beta = \|b\|$, $q_1 = b/\beta$
- Iteration 1: Arnoldi step $\Longrightarrow$ h_{11}, h_{21}, q_2, then

$$\left\| \begin{Bmatrix} \beta \\ 0 \end{Bmatrix} - \begin{bmatrix} h_{11} \\ h_{21} \end{bmatrix} \{\alpha_1\} \right\|^2 \to \text{min}; \qquad x_1 := q_1\alpha_1$$

  Iteration 2: Arnoldi step $\Longrightarrow$ $h_{12}, h_{22}, h_{32}, q_3$, then

$$\left\| \begin{Bmatrix} \beta \\ 0 \\ 0 \end{Bmatrix} - \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ 0 & h_{32} \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} \right\|^2 \to \text{min}; \qquad x_2 := q_1\alpha_1 + q_2\alpha_2$$

  Iteration 3: Arnoldi step $\Longrightarrow$ $h_{13}, h_{23}, h_{33}, h_{43}, q_4$, then

$$\left\| \begin{Bmatrix} \beta \\ 0 \\ 0 \\ 0 \end{Bmatrix} - \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \\ 0 & 0 & h_{43} \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{Bmatrix} \right\|^2 \to \text{min}; \qquad x_3 := q_1\alpha_1 + q_2\alpha_2 + q_3\alpha_3$$

  and so on (you get the picture)...
- Size of least-squares problems grows with the iterations
- Restarted GMRES: run m (50-100) iterations, then stop, fetch x_m and restart with $x_0 := x_m$

GMRES as polynomial approximation problem

- Key observation: $x_k \in \mathcal{K}_k(A, b)$, so $\boxed{x_k = q(A)b}$ for some degree-$(k-1)$ polynomial q
- Let $\mathcal{P}_k = \{\, p \text{ polynomial of degree } k,\ \ p(0) = 1 \,\}$
  Then: residual: $r_k := b - Ax_k = \big(1 - Aq(A)\big)b$ and $1 - Xq(X) \in \mathcal{P}_k$
- Defining least-squares problem for GMRES:

$$
\begin{aligned}
&\text{Find } x_k \in \mathbb{K}^n, \quad \|b - Ax_k\|_2^2 \to \text{minimum} \\
\iff \quad &\text{Find } p_k \in \mathcal{P}_k, \quad \|p_k(A)b\|_2^2 \to \text{minimum} \qquad \Big(\text{i.e.} \quad p_k = \arg\min_{p \in \mathcal{P}_k} \|p(A)b\|_2^2\Big)
\end{aligned}
$$

Convergence of GMRES iterates

- Since $p_A(A) = 0$, GMRES must converge after at most n iterations.
- If ℓ is the grade of b relative to A, GMRES converge after ℓ iterations to x_ℓ with $b - Ax_\ell = 0$.

- Minimizing polynomial p_k verifies $\|r_k\| = \|p_k(A)b\| \implies \boxed{\|r_k\| \leq \|p_k(A)\|\|b\|}$.
  How small can $\|p_k(A)\|_2$ be for a given matrix A?

GMRES as polynomial approximation problem

- How small can $\|p_k(A)\|_2$ be for a given matrix A?
- Relatively simple analysis for A diagonalizable: assume

$$A = X\Lambda X^{-1}, \qquad X \in \mathbb{K}^{n\times n},\ \Lambda = \text{diag}\big(\lambda_1, \ldots, \lambda_n\big)$$

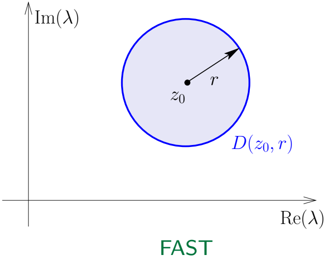
Then:

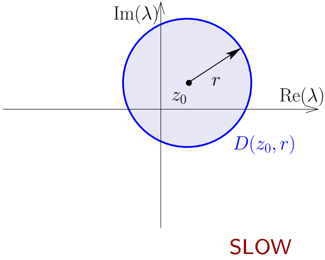
$$p(A) = Xp(\Lambda)X^{-1} \quad \big(\, = X\,\text{diag}\big(p(\lambda_1), \ldots, p(\lambda_n)\big)\, X^{-1}\,\big)$$

$$\begin{aligned}\frac{\|r_k\|}{\|b\|} \leq \|p(A)\| &= \|Xp(\Lambda)X^{-1}\| \\ &\leq \|X\|\|X^{-1}\|\Big\{ \sup_{\lambda=\lambda_1,\ldots,\lambda_n} |p(\lambda)| \Big\} = \kappa(X)\Big\{ \sup_{\lambda=\lambda_1,\ldots,\lambda_n} |p(\lambda)| \Big\}\end{aligned}$$

Fast convergence of GMRES if can find $p_k \in \mathcal{P}_k$ whose size on Λ decreases quickly with degree k.

Favorable situation: assume $\lambda_1, \ldots, \lambda_n$ evenly distributed in $D(z_0, r)$, $|z_0| > r$.





Approximate minimizing polynomial: try $p_k(z) = \big(1 - z/z_0\big)^k$ (satisfies $p(0) = 1$).

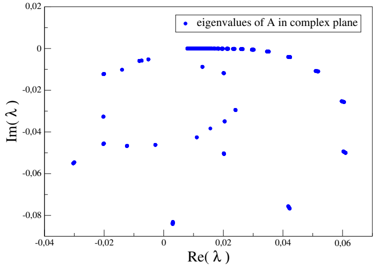
Then: $\boxed{\sup_{z \in D(z_0,r)} |p_k| = \Big(\dfrac{r}{|z_0|}\Big)^k}$: residual reduced by factor $r/|z_0| < 1$ (or less) at each iteration

Eigenvalues of A **tightly** clustered **away from 0** $\implies$ fast convergence of GMRES

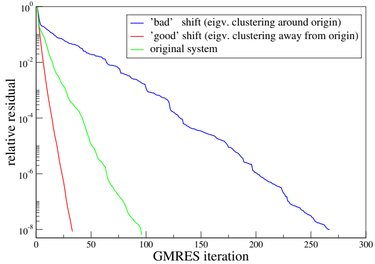
- By contrast: eigenvalues of A clustered around 0 $\implies$ slow convergence of GMRES.
- **Preconditioning:** replace $Ax = b$ by **equivalent** system with eigenvalues clustered away from 0.

Example (complex non-symmetric system)

Complex unsymmetric square system (from BEM model of wave scattering by obstacle), $n = 2520$



Original system $Ax = b$; shifted system $(A + \sigma I)x = b'$ (with $b' = b = \sigma x$ to keep same solution!)



Krylov-space interpretation of conjugate gradient method

- Recall CGM is minimization algorithm for $J(x) := \frac{1}{2}x^{\mathsf{T}}Ax - x^{\mathsf{T}}b + c$
- Recall main steps of CG algorithm: $x_0 = 0$, $r_0 = b$, $p_0 = r_0$, then:

  $$x_k = x_{k-1} + \alpha_k p_{k-1}, \qquad r_k = r_{k-1} - \alpha_k A p_{k-1}, \qquad p_k = r_k + \beta_k p_{k-1}$$

  Induction: $x_k \in \mathcal{K}_k(A, b)$, implies $r_k = b - Ax_k \in \mathcal{K}_{k+1}(A, b)$
- Since A SPD: energy norm and scalar product

  $$\boxed{\|x\|_A^2 := x^{\mathsf{T}}Ax} \quad \lambda_n\|x\|_2 \le \|x\|_A \le \lambda_1\|x\|_2, \qquad \boxed{(y, x)_A := y^{\mathsf{T}}Ax}.$$

  With these definitions, minimize $J(x) \iff$ minimize energy norm of solution error:

  $$J(x) = \frac{1}{2}\left\|x - A^{-1}b\right\|_A^2 + \text{cste} = \frac{1}{2}\left\|x - x^\star\right\|_A^2 + \text{cste}.$$
- Solution error $e_k := x_k - x^\star$ in a Krylov subspace (note $e_0 = -x^\star = -A^{-1}b$):

  $$x_k \in \mathcal{K}_k(A, b) \subset \mathcal{K}_{k+1}(A, A^{-1}b)) = \mathcal{K}_{k+1}(A, e_0) \implies e_k \in \mathcal{K}_{k+1}(A, e_0);$$
- For each iteration k, there exists $p_k \in \mathcal{P}_k$ such that $e_k = p_k(A)e_0$, and

  $$\boxed{\min_{x\in\mathcal{K}_k(A,b)} \left\|x - A^{-1}b\right\|_A^2 \iff \min_{p\in\mathcal{P}_k} \|p(A)e_0\|_A^2}$$

Krylov-space interpretation of conjugate gradient method

$$\boxed{\min_{x\in\mathcal{K}_k(A,b)} \big\|x - A^{-1}b\big\|_A^2 \qquad \Longleftrightarrow \qquad \min_{p\in\mathcal{P}_k} \|p(A)e_0\|_A^2}$$

- $A = Q\Lambda Q^{\mathsf{T}}$ with Q orthogonal (since A SPD).
- Expand $e_0 = q_1\varepsilon_1 + \ldots q_n\varepsilon_n = Q\varepsilon$; then:

$$\begin{aligned} \|p(A)e_0\|_A^2 &= e_0^{\mathsf{T}}\big(Qp(\Lambda)Q^{\mathsf{T}}\big)Q\Lambda Q^{\mathsf{T}}\big(Qp(\Lambda)Q^{\mathsf{T}}\big)e_0 = \varepsilon^{\mathsf{T}}\big(p(\Lambda)\Lambda p(\Lambda)\big)\varepsilon \\ &= \sum_{i=1}^{n} \varepsilon_i^2 \lambda_i \big[p(\lambda_i)\big]^2, \end{aligned}$$

- Upper bound of relative solution error:

$$\frac{\|p(A)e_0\|_A^2}{\|e_0\|_A^2} \leq \lambda_1 \max_{\lambda_1,\ldots,\lambda_n} \big[p(\lambda_i)\big]^2$$

- Estimate convergence rate of CGM via polynomial minimization problem:

$$\min_{p\in\mathcal{P}_k} \Big\{ \max_{\lambda_1\leq\lambda\leq\lambda_n} |p(\lambda)| \Big\},$$

- Solution is known from interpolation theory:

$$\boxed{p_k(\lambda) = \frac{1}{T_k(\gamma)} T_k\Big(\gamma - \frac{2\lambda}{\lambda_1 - \lambda_n}\Big)}, \qquad \gamma := \frac{\lambda_1 + \lambda_n}{\lambda_1 - \lambda_n} > 1$$

($T_k(t)$: Chebyshev polynomial, $T_k(\cos\theta) = \cos(k\theta)$)

Krylov-space interpretation of conjugate gradient method

$$\boxed{p_k(\lambda) = \frac{1}{T_k(\gamma)} T_k\Big(\gamma - \frac{2\lambda}{\lambda_1 - \lambda_n}\Big)}, \qquad \gamma := \frac{\lambda_1 + \lambda_n}{\lambda_1 - \lambda_n} = \frac{\kappa + 1}{\kappa - 1} > 1 \quad (\kappa = \kappa_2(A))$$

- $T_k(t)$ Chebyshev polynomial:

  $|t| \le 1,\ t = \cos\theta \qquad T_k(t) = \cos(k\theta),$

  $|t| \ge 1 \qquad T_k(t) = \frac{1}{2}(z^k + z^{-k}),$ with $t = \frac{1}{2}(z + z^{-1})$

- Majorization: $-1 \le \big(\gamma - \frac{2\lambda}{\lambda_1 - \lambda_n}\big) \le 1$, hence

  $$T_k\Big(\gamma - \frac{2\lambda}{\lambda_1 - \lambda_n}\Big) \le 1$$

- Evaluate $T_k(\gamma)$: $\gamma = (z + z^{-1})/2$ with $z = (\sqrt{\kappa}+1)/(\sqrt{\kappa}-1)$, and hence

  $$T_k(\gamma) = \frac{1}{2}\Big[\Big(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\Big)^k + \Big(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\Big)^{-k}\Big] \ge \frac{1}{2}\Big(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\Big)^k.$$

Convergence rate of the conjugate gradient method

Let A SPD with condition number κ. Run k CG iters. on $Ax = b$. Then: $\dfrac{\|e_k\|_A}{\|e_0\|_A} \le 2\Big(\dfrac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^k$

Recall $\frac{\|e_k\|_A}{\|e_0\|_A} \le 2\big(\frac{\kappa-1}{\kappa+1}\big)^k$ for steepest-descent (compare)

Preconditioning

- Convergence rate of CGM, GMRES…strongly depends on A (e.g. eigenvalue distribution)
- Speeding up convergence essential for efficiently solving large systems
- **Preconditioning** $Ax = b$: reformulations such as

$$\begin{aligned} (M_1 A)x &= M_1 b && \text{(left preconditioning),} \\ (AM_2)z &= b, \quad M_2 z = x && \text{(right preconditioning),} \\ (M_1 A M_2)z &= M_1 b, \quad M_2 z = x && \text{(symmetric preconditioning).} \end{aligned}$$

- Heuristic idea: find (invertible) preconditioners M_1 and/or M_2 producing equivalent systems with better convergence properties.
- Sometimes, preconditioners in inverse form, e.g.

$$\begin{aligned} (M_1^{-1} A)x &= M_1^{-1} b && \text{(left preconditioning),} \\ (AM_2^{-1})z &= b, \quad M_2 x = z && \text{(right preconditioning).} \end{aligned}$$

Preconditioning

Preconditioning largely an "art": no general methodology, tailor to specifics of considered application. Some common preconditioning approaches:

- Diagonal preconditioner, e.g. $M_1 = \mathrm{diag}(a_{11}^{-1}, \ldots, a_{nn}^{-1})$ (left preconditioner).
- Block preconditioners (M_1 or M_2 block-diagonal) sometimes naturally suggested by problem (multi-component physical system).
- Incomplete (e.g. LU, Cholesky) factorizations, triangular factors sparse or otherwise easy to invert;
- Sparse approximate inverse: sparse matrix M such that (e.g.) $\|I - MA\| \to \min$;
- Use less-precise discretizations (e.g. multigrid methods, truncated Fourier/wavelet expansions);

Example: left-preconditioned GMRES algorithm

- Take left-preconditioned system of the form $\boxed{M^{-1}Ax = M^{-1}b}$.
- Each Krylov basis vector of the form $q_{k+1} = M^{-1}Aq_k$, i.e. $Mq_{k+1} = Aq_k$

Algorithm 8 Left-preconditioned GMRES algorithm

$A \in \mathbb{K}^{n\times n}$ and $b \in \mathbb{K}^n$ (problem data)
$x = 0$, $\beta = \|b\|$, $q_1 = b/\beta$ (initialization)
for $k = 1, 2, \ldots$ **do**
  solve $Mq_{k+1} = Aq_k$ for q_{k+1} (new Krylov vector: initialize q_{k+1})
  for $j = 1$ to k **do**
    $h_{jk} = q_j^{\mathsf{H}} q_{k+1}$ (entry in k-th column of H_k)
    $q_{k+1} = q_{k+1} - h_{jk}q_j$ (update (not yet normalized) vector q_{k+1})
  end for
  $h_{k+1,k} = \|q_{k+1}\|$ (last entry in k-th column of H_k)
  $q_{k+1} = q_{k+1}/\|h_{k+1,k}\|$ (normalize q_{k+1})
  Find $y \in \mathbb{K}^k$, $\|be_1 - H_k y\|_2^2 \to \min$ (least squares problem)
  $x = Q_k y$ (current solution)
  Stop if convergence, return x
end for

Example: acoustic scattering by sound-hard obstacle

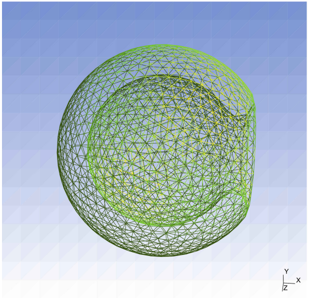
Boundary integral equation (u: (unknown) total acoustic field, u_{inc} (given) incident field)

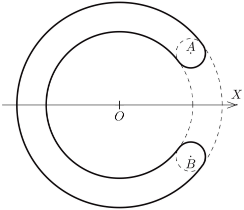
$$\boxed{\tfrac{1}{2}u(\boldsymbol{x}) + \int_S \boldsymbol{n}(\boldsymbol{y})\cdot\boldsymbol{\nabla} G(\boldsymbol{x},\boldsymbol{y})u(\boldsymbol{y})\,\mathrm{d}S(\boldsymbol{y}) = u_{\text{inc}}(\boldsymbol{x}) \qquad \text{for all } \boldsymbol{x}\in S} \qquad (S\text{: obstacle surface})$$

$$G(\boldsymbol{x},\boldsymbol{y}) = \frac{e^{i\omega|\boldsymbol{y}-\boldsymbol{x}|/c}}{4\pi|\boldsymbol{y}-\boldsymbol{x}|} \qquad \text{(acoustic field at } \boldsymbol{y} \text{ created by unit source at } \boldsymbol{x})$$

Discretized using boundary elements on S ("finite elements on surfaces")

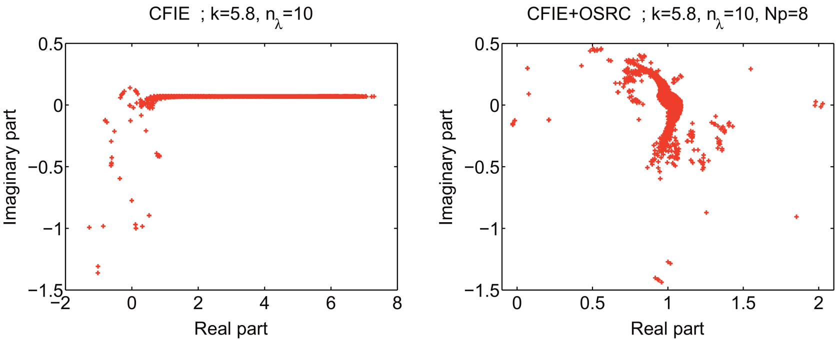
$\boxed{Au = u_{\text{inc}}}$, A dense, compression methods available, solved using GMRES.





2D C-shape contour

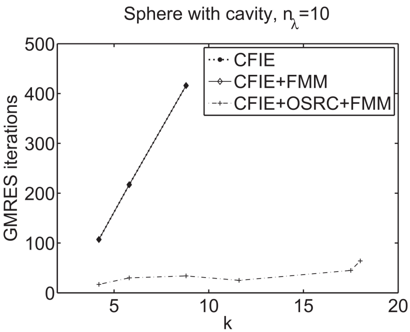
Darbas M., Darrigrand E., Lafranche Y. Combining analytic preconditioner and Fast Multipole Method for the 3D Helmholtz equation. *J. Comput. Phys.*, 236:289–316 (2013).



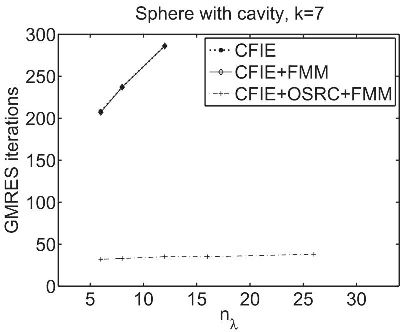
Eigenvalue distribution, non-preconditioned (left) and preconditioned (right) formulations

Darbas M., Darrigrand E., Lafranche Y. Combining analytic preconditioner and Fast Multipole Method for the 3D Helmholtz equation. *J. Comput. Phys.*, 236:289–316 (2013).

Example: preconditioned BEM for scattering problem



(a) GMRES iterations vs k



(b) GMRES iterations vs n_λ

GMRES iteration count with/without preconditioning, vs. frequency (left) or mesh density/wavelength (right).

Darbas M., Darrigrand E., Lafranche Y. Combining analytic preconditioner and Fast Multipole Method for the 3D Helmholtz equation. *J. Comput. Phys.*, 236:289–316 (2013).