Chapter 1

Lectures

1.1 Lecture 1: 19.08.2025

19. August 2025

What is the course about? *Solving linear problems*

$$A\mathbf{x} = \mathbf{b}$$

and *eigenvalue problems*

$$A\mathbf{v} = \lambda\mathbf{v}$$

for A large (e.g., $n \geq 10^4$), and *sparse* (most elements are non-zero).

$N_z(A)$: number of non-zero elements in A.

Stick to $A\mathbf{x} = \mathbf{b}$, $A \in \mathbb{R}^{n\times n}$ and non-singular.

Classical methods:

LU decomposition $A = (P)LU$ (Gaussian elimination, complexity $\mathcal{O}(n^3)$)

If A is symmetric positive definite (SPD), i.e. $A = A^T > 0$, then Cholesky decomposition $A = C^TC$ where C is triangular. Complexity $\mathcal{O}(n^3)$.

Standard test problems: Discrete Laplacian in 2D: Discretization of $\Delta u = f$ in a square domain $\Omega = (0,1)\times(0,1)$ with Dirichlet boundary conditions $u = g$ on $\partial\Omega$.

$$\Delta u = u_{xx} + u_{yy} = f,$$
$$\text{with} \quad \begin{cases} u = g & \text{on } \partial\Omega, \\ h = \frac{1}{N+1}, \\ x_i = ih, \quad y_j = jh, \quad i,j = 0,\dots,N+1 \end{cases}$$

$$U_{ij} \approx u(x_i, y_j) = u_{ij},$$

$$u_{xx}\big|_{(x_i,y_j)} \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2} + \mathcal{O}(h^2),$$

$$u_{yy}\big|_{(x_i,y_j)} \approx \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h^2} + \mathcal{O}(h^2).$$

This leads to the linear system (5-point formula):

$$4U_{ij} - U_{i+1,j} - U_{i-1,j} - U_{i,j+1} - U_{i,j-1} = h^2 f_{ij}, \quad i, j = 1, \dots, N$$

This can be written in matrix form $A\mathbf{U} = \mathbf{f}$, where $\mathbf{U}$ is the vector of unknowns U_{ij} and $\mathbf{f}$ is the vector of right-hand side values f_{ij}. A is a *block tridiagonal matrix* with blocks $B \in \mathbb{R}^{N\times N}$, where B is the discrete Laplacian in one dimension:

$$\begin{aligned}
A\mathbf{U} &= \mathbf{f}, \\
A &= \begin{bmatrix} B & -I_N & 0 & \cdots & 0 \\ -I_N & B & -I_N & \cdots & 0 \\ 0 & -I_N & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -I_N \\ 0 & 0 & 0 & -I_N & B \end{bmatrix}, \qquad B = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \cdots & 0 \\ 0 & -1 & 4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & -1 & 4 \end{bmatrix}, \\
\mathbf{U} &= \begin{bmatrix} U_{11} \\ U_{12} \\ \vdots \\ U_{1N} \\ U_{21} \\ U_{22} \\ \vdots \\ U_{NN} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_{11} \\ f_{12} \\ \vdots \\ f_{1N} \\ f_{21} \\ f_{22} \\ \vdots \\ f_{NN} \end{bmatrix}.
\end{aligned}$$

Properties of A

The matrix A is *symmetric*, *sparse*, and *structured*. In particular, A is a **banded matrix**. Total of N^2 equations.

Banded Matrix

Definition 1.1: Banded Matrix

A is banded with bandwidth:

$$m_u + m_l + 1 \text{ if } a_{ij} \neq 0 \text{ only if } |i - j| \leq m_u + m_l$$

where m_u is the upper bandwidth and m_l is the lower bandwidth.

For the discrete Laplacian, A has bandwidth $2N + 1$.

Even if A is sparse the LU-factorization is not (fill-in), however the banded structure is preserved.

1.1.1 Iterative techniques for solving linear systems

Let $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{n\times n}$.

Instead of solving the system directly (which becomes expensive for large n), we generate a sequence of approximations $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$ that converges to the exact solution $\mathbf{x}^*$.

Classical iterative methods/Fixed-point iterations: The key idea is to split the matrix A into two parts: an "easy" part M and the remainder N.

Basic approach:

$$\begin{aligned} A &= M - N, \\ M\mathbf{x} &= N\mathbf{x} + \mathbf{b}, \\ \mathbf{x} &= M^{-1}N\mathbf{x} + M^{-1}\mathbf{b}, \\ \mathbf{x}_{k+1} &= M^{-1}N\mathbf{x}_k + M^{-1}\mathbf{b}. \end{aligned}$$

Choose M such that:

- $M\mathbf{v} = \mathbf{c}$ is easy to solve
- $\rho(M^{-1}N) < 1$ (spectral radius) for convergence
- $\mathbf{c} = M^{-1}\mathbf{b}$

Standard splitting methods:

Let $A = D + L + U$ where:

- D = diagonal part of A
- L = strictly lower triangular part of A
- U = strictly upper triangular part of A

- **Jacobi:** $M = D$, $N = L + U$
- **Gauss-Seidel:** $M = D + L$, $N = U$
- **SOR (Successive Over-Relaxation):** $M = \frac{1}{\omega}D + L$, $N = \frac{1-\omega}{\omega}D - U$, where $0 < \omega < 2$

1.1.2 Projection methods for solving linear systems

Idea (of one iteration): Choose $\mathcal{L}, \mathcal{K} \subset \mathbb{R}^n$ where $\dim(\mathcal{K}) = \dim(\mathcal{L}) = m \ll n$. Choose some initial guess $\mathbf{x}_0 \in \mathbb{R}^n$:

$$\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}_1, \text{ s.t. the residual } \mathbf{r}_1 = A\mathbf{x}_1 - \mathbf{b} \perp \mathcal{L},$$

Example

Let $\mathcal{K} = \mathcal{L} = \text{span}\{\mathbf{r}_0\}$, where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ is the initial residual.

Then we can write:

$$\begin{aligned} \Delta\mathbf{x}_0 &= \alpha_0\mathbf{r}_0, \alpha_0 \in \mathbb{R}, \\ \mathbf{r}_1 &= \mathbf{b} - A\mathbf{x}_1 = \mathbf{b} - A(\mathbf{x}_0 - \alpha_0\mathbf{r}_0) = \mathbf{r}_0 - \alpha_0 A\mathbf{r}_0. \end{aligned}$$

We can choose α_0 such that $\mathbf{r}_1 \perp \mathcal{L}$, i.e. $\langle \mathbf{r}_1, \mathbf{v}\rangle = 0$ for all $\mathbf{v} \in \mathcal{L}$. This leads to the equation:

$$\langle \mathbf{r}_1, \mathbf{r}_0\rangle = \langle \mathbf{r}_0, \mathbf{r}_0\rangle - \alpha_0\langle A\mathbf{r}_0, \mathbf{r}_0\rangle = 0.$$

Solving for α_0 gives:

$$\alpha_0 = \frac{\langle \mathbf{r}_0, \mathbf{r}_0\rangle}{\langle A\mathbf{r}_0, \mathbf{r}_0\rangle}.$$

This is the first step in a projection method, where we iteratively refine our solution by projecting onto the subspace defined by the initial residual.

1.1.3 How to store sparse matrices?

- **List of lists (LIL)**: Each row is stored as a list of non-zero elements and their column indices.

$$\text{LIL} = \begin{bmatrix} [1,2,3] & [4,5] & [6] \\ [7,8] & [9] & [] \\ [] & [10,11] & [12] \end{bmatrix}$$

- **Compressed Sparse Row (CSR)**: Three arrays: values, column indices, and row pointers.

$$\begin{aligned} \text{values} &= [1,2,3,4,5,6], \\ \text{col\_indices} &= [0,1,2,0,1,2], \\ \text{row\_pointers} &= [0,3,5,6] \end{aligned}$$

- **Compressed Sparse Column (CSC)**: Similar to CSR but column-wise.

$$\begin{aligned} \text{values} &= [1,4,2,5,3,6], \\ \text{row\_indices} &= [0,1,0,1,2,2], \\ \text{col\_pointers} &= [0,2,4,6] \end{aligned}$$

- **Coordinate List (COO)**: Three arrays: row indices, column indices, and values.

$$\begin{aligned} \text{row\_indices} &= [0,0,0,1,1,2], \\ \text{col\_indices} &= [0,1,2,0,1,2], \\ \text{values} &= [1,2,3,4,5,6] \end{aligned}$$

1.2 Lecture 2: 20.08.2025

Let $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$ be two vectors. The *inner product* $(\cdot, \cdot)$ and *norm* (unless otherwise specified) are defined as:

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i \overline{y_i} = \mathbf{x}^H \mathbf{y}, \quad \|\mathbf{x}\|^2 = (\mathbf{x}, \mathbf{x}) = \sum_{i=1}^{n} |x_i|^2 = \mathbf{x}^H \mathbf{x}.$$

1.2.1 Unitary Matrices

A matrix $Q \in \mathbb{C}^{n\times n}$ is *unitary* if $Q^H Q = I_n$, where I_n is the $n \times n$ identity matrix. The columns of Q form an orthonormal set, meaning they are mutually orthogonal and each has unit norm.

Let $Q = [q_1, q_2, \dots, q_n]$. Then the orthonormality condition is:

$$(q_i, q_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Examples of Unitary Matrices

1. **Identity matrix**: I_n is trivially unitary.
2. **2D rotation matrices** (real orthogonal):

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

   Verification: $R(\theta)^T R(\theta) = I_2$ since $\cos^2(\theta) + \sin^2(\theta) = 1$.

3. **Givens rotation**: $G(i, j, \theta)$ rotates components i and j by angle θ:

$$G(i, j, \theta) = \begin{bmatrix} I_{i-1} & & & \\ & c & -s & \\ & s & c & \\ & & & I_{n-j} \end{bmatrix}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$, and the 2×2 rotation block appears at positions (i, i) through (j, j).
4. **Householder reflector**: Given a unit vector $v \in \mathbb{C}^n$ with $\|v\|_2 = 1$:

$$P = I_n - 2vv^H$$

This matrix satisfies $P = P^H = P^{-1}$ (it is Hermitian and unitary).
Verification of unitarity:

$$\begin{aligned} P^H P &= (I_n - 2vv^H)^2 \\ &= I_n - 4vv^H + 4v(v^H v)v^H \\ &= I_n - 4vv^H + 4vv^H = I_n \end{aligned}$$

Geometric interpretation: For any vector $\mathbf{x}$:

$$P\mathbf{x} = \mathbf{x} - 2(v^H \mathbf{x})v = \mathbf{x} - 2(\mathbf{x}, v)v$$

This reflects $\mathbf{x}$ across the hyperplane orthogonal to v.

Key Properties of Unitary Matrices

- **Inner product preservation**: $(Q\mathbf{x}, Q\mathbf{y}) = (\mathbf{x}, \mathbf{y})$
- **Norm preservation**: $\|Q\mathbf{x}\| = \|\mathbf{x}\|$
- **Unit determinant**: $|\det(Q)| = 1$
- **Eigenvalues on unit circle**: All eigenvalues of Q satisfy $|\lambda| = 1$

Applications

- **Spectral decomposition**: If $A = A^H$, then $A = V\Lambda V^H$ where V is unitary and Λ is real diagonal.
- **QR decomposition**: Any matrix A can be factored as $A = QR$ where Q is unitary and R is upper triangular.

1.2.2 QR Decomposition

The QR decomposition is a fundamental matrix factorization that expresses any matrix $A \in \mathbb{C}^{m \times n}$ (with $m \geq n$) as the product $A = QR$, where $Q \in \mathbb{C}^{m \times m}$ is unitary and $R \in \mathbb{C}^{m \times n}$ is upper triangular. When A has full column rank, this decomposition is unique up to signs.

The QR decomposition has numerous applications including:

- Solving least squares problems: $\min_x \|Ax - b\|_2$
- Computing matrix eigenvalues (QR algorithm)
- Orthogonalizing vectors (Gram-Schmidt process)
- Numerical solution of linear systems

There are several algorithms for computing the QR decomposition, with Householder reflections being the most numerically stable and widely used in practice.

Householder Reflections for QR

The key idea is to use a sequence of Householder reflectors to systematically introduce zeros below the diagonal of A. For column k, we construct a Householder matrix P_k that zeros out entries $k+1, k+2, \ldots, m$ in that column, while preserving the upper triangular structure already achieved in previous columns.

The complete factorization is:

$$P_n P_{n-1} \cdots P_2 P_1 A = R$$

where each P_k is a Householder reflector. Since each P_k is unitary, we have:

$$A = \underbrace{P_1^H P_2^H \cdots P_n^H}_{Q} R$$

Algorithm

Given a vector $x \in \mathbb{C}^m$, we construct a Householder reflector P such that $Px = \pm\|x\|_2 e_1$.

Construction of Householder vector:

$$\begin{aligned} \sigma &= \begin{cases} -1 & \text{if } \Re(x_1) > 0 \\ 1 & \text{if } \Re(x_1) \leq 0 \end{cases} \\ u &= x - \sigma\|x\|_2 e_1 \\ v &= \frac{u}{\|u\|_2} \end{aligned}$$

The sign choice prevents cancellation when $|x_1| \approx \|x\|_2$.

Result: $Px = (I - 2vv^H)x = -\sigma\|x\|_2 e_1$

Full QR Algorithm

For $k = 1, 2, \ldots, n$:

1. Extract subcolumn: $x = A_{k:m,k}$
2. Construct Householder vector v_k as above
3. Apply reflection: $A_{k:m,k:n} \leftarrow A_{k:m,k:n} - 2v_k(v_k^H A_{k:m,k:n})$
4. Store v_k in $A_{k+1:m,k}$ (below diagonal)

Complexity: The total computational cost is: $2mn^2 - \frac{2}{3}n^3$ flops for $m \times n$ matrix.

Worked Example

Consider $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$.

Step 1 — First column:

- $x = [1, 1, 1]^T$, $\|x\|_2 = \sqrt{3}$
- $\sigma = -1$ (since $x_1 = 1 > 0$)
- $u = [1, 1, 1]^T + \sqrt{3}[1, 0, 0]^T = [1 + \sqrt{3}, 1, 1]^T$
- $v_1 = u / \|u\|_2$
- $P_1 A = \begin{bmatrix} -\sqrt{3} & -2\sqrt{3} \\ 0 & \star \\ 0 & \star \end{bmatrix}$

Step 2 — Second column (rows 2:3): Apply similar process to zero out the $(3,2)$ entry.

Result: $R = P_2P_1A$ is upper triangular, and $Q = P_1^T P_2^T$.

Implementation Notes

- **Never form P explicitly**: Use the update $A \leftarrow A - 2v(v^H A)$
- **In-place storage**: Store Householder vectors below the diagonal
- **Numerical stability**: The algorithm is backward stable with excellent numerical properties

Visualization of Householder reflection

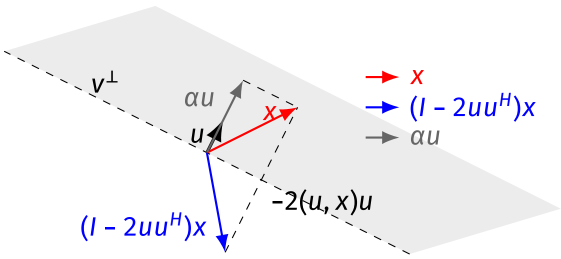
The goal of this figure is to make the algebraic action of a Householder reflector visually transparent. Let u be a unit vector (the reflector normal) and set

$$\alpha = u^H x, \qquad \pi_u(x) = \alpha u, \qquad P = I - 2uu^H.$$

Then we have the decomposition

$$x = \pi_u(x) + (x - \pi_u(x)), \qquad Px = -\pi_u(x) + (x - \pi_u(x)).$$

In words: the component of x parallel to u (the projection $\pi_u(x)$) is negated by P, while the perpendicular component (lying in $u^\perp$) is unchanged. The TikZ picture below illustrates these parts.



Householder reflection of a vector x across the hyperplane orthogonal to u. The projection $\pi_u(x)$ is shown in grey, while the reflected vector Px is shown in blue.

Remarks and interpretation:

- Decomposition: the figure shows x (red), its projection $\pi_u(x)$ (grey), and the reflected vector Px (blue). Algebraically $Px = x - 2\alpha u$.
- Symmetry: the projection point $\pi_u(x)$ lies midway (along the u–direction) between x and Px, which is the geometric content of the reflector.
- Use in QR: algorithmically one chooses u so that Px becomes a (signed) multiple of a basis vector (e.g. $\pm\|x\|_2 e_1$); repeating this across columns zeros subdiagonals and produces an upper triangular R.

1.3 Lecture 3: 26.08.2025

1.3.1 Eigenvalues and Eigenvectors

Let $A \in \mathbb{C}^{n\times n}$ be a square matrix. An **eigenvalue** $\lambda \in \mathbb{C}$ and corresponding **eigenvector** $\mathbf{v} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$ satisfy:

$$A\mathbf{v} = \lambda\mathbf{v}$$

For the conjugate transpose A^H, we have:

$$A^H \mathbf{w} = \bar{\lambda} \mathbf{w}$$

Remark 1
If A is Hermitian (i.e., $A^H = A$), then all eigenvalues are real: $\lambda \in \mathbb{R}$. If A is singular, then $\lambda = 0$ is an eigenvalue.

1.3.2 Matrix Properties and Non-singularity

Definition 1.2: Strictly Diagonally Dominant Matrix

A matrix $A \in \mathbb{C}^{n \times n}$ is **strictly diagonally dominant** if

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \quad \text{for all } i = 1, 2, \dots, n$$

Theorem 1.3: Non-singularity of Strictly Diagonally Dominant Matrices

Every strictly diagonally dominant matrix is non-singular.

Definition 1.4: Irreducible Matrix

A matrix $A \in \mathbb{C}^{n \times n}$ is **irreducible** if for every pair of indices $i, j \in \{1, 2, \dots, n\}$, there exists a sequence of indices $i = m_0, m_1, m_2, \dots, m_k = j$ such that $a_{m_\ell m_{\ell+1}} \neq 0$ for all $\ell = 0, 1, \dots, k - 1$.
Equivalently, the directed graph associated with the matrix is strongly connected.

A matrix A is **reducible** if and only if there exists a permutation matrix P such that

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ \mathbf{0} & A_{22} \end{bmatrix}$$

where A_{11} and A_{22} are square matrices.

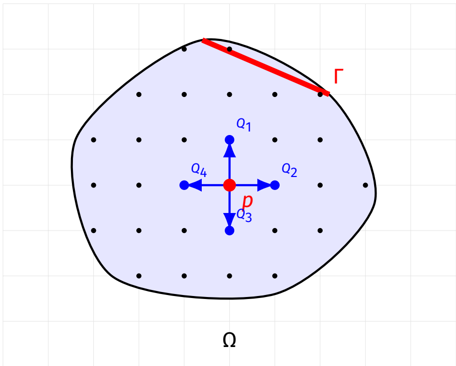
Theorem 1.5: Irreducible Diagonally Dominant Matrices

If A is irreducible and diagonally dominant with at least one row strictly diagonally dominant, then A is non-singular.

Example 1. Finite Difference Discretization
Consider the Poisson equation $\Delta u = u_{xx} + u_{yy} = f(x, y)$ on domain Ω with boundary condition $u = g$ on $\Gamma \subset \partial\Omega$.
The finite difference discretization yields the linear system:

$$\alpha_{pp} U_p + \sum_{\ell=1}^{N_p} \alpha_{pQ_\ell} U_{Q_\ell} = f_p \quad \text{for } p = 1, 2, \dots, M$$

where:
- p is a grid point in the interior domain
- Q_ℓ are the neighboring points of p
- N_p is the number of neighbors of p
- U_p is the approximate solution at grid point p



1.3.3 Gershgorin Circle Theorem

Theorem 1.6: Gershgorin Circle Theorem

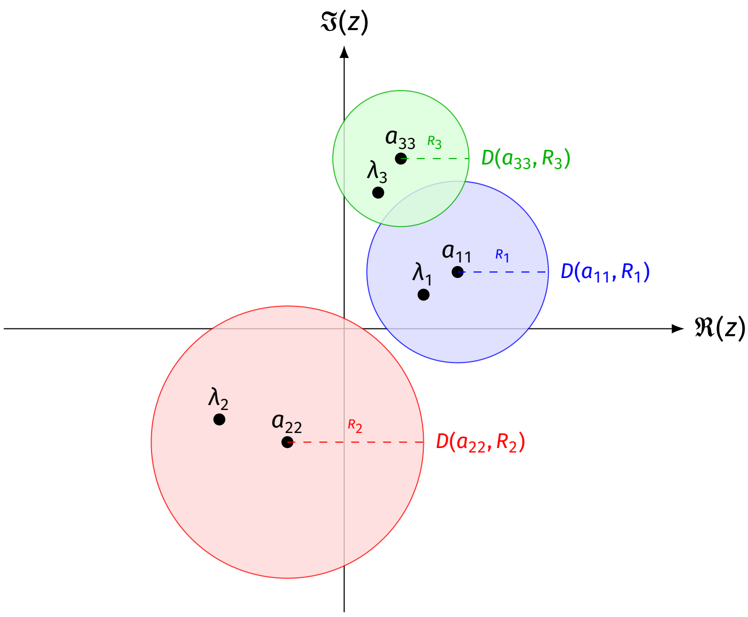
Let $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ and define the **row radii**:

$$R_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \quad \text{for } i = 1, 2, \dots, n$$

Then every eigenvalue of A lies within the union of **Gershgorin discs**:

$$\sigma(A) \subseteq S_R = \bigcup_{i=1}^{n} D(a_{ii}, R_i)$$

where $D(a_{ii}, R_i) = \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i\}$ is the closed disc centered at a_{ii} with radius R_i.



Theorem 1.7: Gershgorin Separation

Let $S_1 = \bigcup_{i=1}^{\ell} D(a_{ii}, R_i)$ and $S_2 = \bigcup_{i=\ell+1}^{n} D(a_{ii}, R_i)$ where $S_1 \cap S_2 = \emptyset$.
Then A has exactly ℓ eigenvalues in S_1 and $n - \ell$ eigenvalues in S_2.

Proof. Let $\lambda \in \sigma(A)$ with corresponding eigenvector $\mathbf{v} = (v_1, v_2, \dots, v_n)^T$. Normalize so that $\|\mathbf{v}\|_\infty = 1$, and let m be an index such that $|v_m| = 1$.
From the eigenvalue equation $A\mathbf{v} = \lambda\mathbf{v}$, the m-th component gives:

$$\sum_{j=1}^{n} a_{mj} v_j = \lambda v_m$$

$$(\lambda - a_{mm}) v_m = \sum_{\substack{j=1 \\ j \neq m}}^{n} a_{mj} v_j$$

Taking absolute values and using $|v_j| \leq 1$ for all j:

$$\begin{aligned} |\lambda - a_{mm}||v_m| &= \left| \sum_{\substack{j=1 \\ j \neq m}}^{n} a_{mj} v_j \right| \\ &\leq \sum_{\substack{j=1 \\ j \neq m}}^{n} |a_{mj}||v_j| \\ &\leq \sum_{\substack{j=1 \\ j \neq m}}^{n} |a_{mj}| = R_m \end{aligned}$$

Since $|v_m| = 1$, we have $|\lambda - a_{mm}| \leq R_m$, so $\lambda \in D(a_{mm}, R_m) \subseteq S_R$. □

Theorem 1.8: Gershgorin for Irreducible Matrices

If A is irreducible and λ lies on the boundary of some Gershgorin disc $\partial D(a_{ii}, R_i)$, then λ lies on the boundary of every Gershgorin disc.

Proof of Theorem ??.
Suppose λ lies on the boundary of $D(a_{mm}, R_m)$. Then equality holds in the previous proof:

$$|\lambda - a_{mm}| = \sum_{\substack{j=1 \\ j \neq m}}^{n} |a_{mj}| \frac{|v_j|}{|v_m|} = R_m$$

This requires $|v_j| = |v_m|$ for all j such that $a_{mj} \neq 0$.
Since A is irreducible, for any indices i, j, there exists a path $i = m_0, m_1, \dots, m_k = j$ with $a_{m_\ell m_{\ell+1}} \neq 0$ for all $\ell = 0, 1, \dots, k-1$.
By the same argument, we get $|v_{m_\ell}| = |v_{m_{\ell+1}}|$ for all ℓ, which implies $|v_i| = |v_j|$ for all i, j.
Therefore, $|\lambda - a_{ii}| = R_i$ for all i, meaning λ lies on the boundary of every Gershgorin disc. □

1.3.4 Continuity of Eigenvalues

Consider the matrix family $A(t) = D + tH$ where D is diagonal and H contains the off-diagonal entries of A, with $t \in [0, 1]$.

$$A(t) = D + tH \quad \text{where} \begin{cases} A(0) = D & \text{(diagonal matrix)} \\ A(1) = A & \text{(original matrix)} \end{cases}$$

The eigenvalues $\lambda(t)$ of $A(t)$ vary continuously with respect to t. The eigenvalues of $A(0) = D$ are simply $a_{11}, a_{22}, \dots, a_{nn}$.

If D has distinct diagonal entries, then as t varies from 0 to 1, each eigenvalue remains within its corresponding Gershgorin disc, providing insight into eigenvalue perturbation.

1.4 Lecture 4: 27.08.2025

1.4.1 Similarity and eigenvectors

Let $A \in \mathbb{C}^{n\times n}$. If $B = X^{-1}AX$ with $\det X \neq 0$, then A and B are similar and have the same eigenvalues. If $Av = \lambda v$, then $X^{-1}v$ is an eigenvector of B with eigenvalue λ.

If A is diagonalizable with eigenbasis $V = [v_1, \dots, v_n]$, then

$$V^{-1}AV = \Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n).$$

If A is defective, there exists invertible X with

$$X^{-1}AX = J = \operatorname{blockdiag}\Big(J_1(\lambda_1), \dots, J_s(\lambda_s)\Big), \qquad J_k(\lambda) = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} \text{ or larger.}$$

1.4.2 Schur decomposition

Theorem 1.9: Schur decomposition

or any $A \in \mathbb{C}^{n\times n}$ there exists a unitary Q and upper triangular T such that

$$A = QTQ^H, \qquad T = Q^HAQ.$$

Proof. Pick a unit eigenvector u of A, complete to a unitary $U = [u\ \tilde{U}]$. Then

$$U^HAU = \begin{bmatrix} \alpha & c^H \\ 0 & \tilde{A} \end{bmatrix}.$$

By induction, choose unitary $\tilde{V}$ with $\tilde{V}^H\tilde{A}\tilde{V} = T_{n-1}$. With $Q = U\ \operatorname{diag}(1, \tilde{V})$,

$$Q^HAQ = \begin{bmatrix} \alpha & b^H \\ 0 & T_{n-1} \end{bmatrix},$$

which is upper triangular. □ □

Hermitian case: If $A = A^H$, then T is normal and upper triangular, hence diagonal with real entries. Thus

$$A = Q\Lambda Q^H, \qquad \Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n) \in \mathbb{R}^{n\times n}.$$

1.4.3 Real Schur form

For $A \in \mathbb{R}^{n\times n}$ there exists orthogonal Q with

$$A = QTQ^T, \qquad T = \begin{bmatrix} T_1 & * \\ 0 & T_2 \end{bmatrix},$$

where each diagonal block T_i is either 1×1 (real eigenvalue) or a real 2×2 block corresponding to a complex conjugate pair.

1.4.4 QR factorization

For $A \in \mathbb{R}^{m\times n}$ with $m \geq n$,

$$A = QR, \qquad Q^TQ = I, \quad R \text{ upper triangular}, \quad R = Q^TA.$$

1.4.5 Eigenvalue perturbation

Let $Au = \lambda u$ and $v^HA = \lambda v^H$ with $\|u\|_2 = \|v\|_2 = 1$. For $A(\varepsilon) = A + \varepsilon E$ with $|\varepsilon| \ll 1$, the first–order eigenvalue change is

$$\delta\lambda = \varepsilon\, v^HEu, \qquad |\delta\lambda| \leq |\varepsilon|\ \|E\|.$$

Condition number of a simple eigenvalue:

$$\kappa(\lambda) = \frac{1}{|v^Hu|}.$$

If $v^Hu \to 0$ (nearly defective), then $\kappa(\lambda) \to \infty$.

1.4.6 Linear system perturbation

Consider

$$(A + \varepsilon E)x(\varepsilon) = b + \varepsilon e, \qquad Ax = b.$$

Let $\delta x = x(\varepsilon) - x$. Then

$$(A + \varepsilon E)\,\delta x = \varepsilon(e - Ex), \qquad \delta x = \varepsilon(A + \varepsilon E)^{-1}(e - Ex).$$

Using $(I + \varepsilon A^{-1}E)^{-1} = I - \varepsilon A^{-1}E + O(\varepsilon^2)$,

$$\delta x = \varepsilon A^{-1}(e - Ex) + O(\varepsilon^2).$$

Relative error bound:

$$\frac{\|\delta x\|}{\|x\|} \lesssim |\varepsilon|\ \kappa(A)\left(\frac{\|e\|}{\|b\|} + \frac{\|E\|}{\|A\|}\right), \qquad \kappa(A) = \|A\|\ \|A^{-1}\|.$$

1.4.7 Projection methods

A projector $P : \mathbb{C}^n \to \mathbb{C}^n$ satisfies $P^2 = P$. Then $\mathrm{Range}(P) = M$ and $\mathrm{Range}(I - P) = \ker(P)$.

Oblique projection: Let $M = \mathrm{span}\{v_1, \ldots, v_m\}$ and $W = \mathrm{span}\{w_1, \ldots, w_m\}$. With $V = [v_1, \ldots, v_m]$ and $W = [w_1, \ldots, w_m]$,

$$P = V(W^*V)^{-1}W^*, \qquad Px \in M, \quad W^*(x - Px) = 0.$$

Orthogonal projection: Take $W = V$. Then

$$P_M = V(V^*V)^{-1}V^*, \qquad P_M^* = P_M, \quad P_M^2 = P_M,$$

and the best-approximation property holds:

$$\|x - P_M x\|_2 = \min_{y \in M} \|x - y\|_2.$$

1.5 Lecture 5: 02.09.2025

Projection Methods

Problem: Solve $A\mathbf{x} = \mathbf{b}$ where $A \in \mathbb{R}^{n \times n}$ is invertible, with solution $\mathbf{x}^\star$.

1. Given $\mathbf{x}_0$, choose $\mathcal{K}, \mathcal{L} \subset \mathbb{R}^n$ with $\dim(\mathcal{K}) = \dim(\mathcal{L}) = m$.
   - $\mathcal{K}$ is the *search space* (or *trial space*)
   - $\mathcal{L}$ is the *constraint space* (or *test space*)
2. Find $\tilde{\mathbf{x}} \in \mathbf{x}_0 + \mathcal{K}$ s.t. $\tilde{\mathbf{r}} = \mathbf{b} - A\tilde{\mathbf{x}} \perp \mathcal{L}$.

Alternative:

1. Let $\delta = \tilde{\mathbf{x}} - \mathbf{x}_0$, $\tilde{\mathbf{r}} = \mathbf{b} - A\tilde{\mathbf{x}} = \mathbf{b} - A(\mathbf{x}_0 + \delta) = \mathbf{r}_0 - A\delta$.
2. Find $\delta \in \mathcal{K}$ s.t. $\mathbf{r}_0 - A\delta \perp \mathcal{L}$.

   $$\tilde{\mathbf{x}} = \mathbf{x}_0 + \delta, \quad \delta \in \mathcal{K}, \quad \mathbf{r}_0 - A\delta \perp \mathcal{L}$$

   In matrix form:

   $$\mathcal{K} = \operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_m\} = \operatorname{span}(V)$$
   $$\mathcal{L} = \operatorname{span}\{\mathbf{w}_1, \dots, \mathbf{w}_m\} = \operatorname{span}(W)$$

   Then:

   $$\delta = V\mathbf{y}, \quad y \in \mathbb{R}^m, \quad \mathbf{r}_0 - AV\mathbf{y} \perp \operatorname{span}(W)$$
   $$W^\top(\mathbf{r}_0 - AV\mathbf{y}) = 0$$
   $$y = (W^\top AV)^{-1} W^\top \mathbf{r}_0$$
   $$\tilde{\mathbf{x}} = \mathbf{x}_0 + V(W^\top AV)^{-1} W^\top \mathbf{r}_0$$

Remarks: Standard choices for $\mathcal{L}$ (A is non-singular):

- if A is SPD, choose $\mathcal{L} = \mathcal{K}$ (Galerkin condition)
- otherwise, choose $\mathcal{L} = A\mathcal{K}$ (Petrov-Galerkin condition)

Questions?

1. Will the method converge?

   $$\|\tilde{\mathbf{x}} - \mathbf{x}^\star\| \le \|\mathbf{x}_0 - \mathbf{x}^\star\|$$
   $$\|\tilde{\mathbf{r}}\| \le \|\mathbf{r}_0\|$$

2. Is $W^\top AV$ invertible?

- if $A = A^\top$ is SPD and $\mathcal{L} = \mathcal{K}$, then:

$$\begin{aligned}
&A \text{ SPD}\\
&A = C^\top C\\
&W = VG, \quad G \in \mathbb{R}^{m\times m} \text{ invertible}\\
&W^\top AV = G^\top V^\top AV = G^\top (C^\top C)^\top (C^\top C)\\
&C^\top V \text{ has rank } m \text{ since } V \text{ has rank } m\\
\Rightarrow\ &W^\top AV \text{ is SPD } \Rightarrow \text{ invertible}
\end{aligned}$$

- if A invertible and $\mathcal{L} = A\mathcal{K}$, then:

$$\begin{aligned}
&W = AVG, \quad G \in \mathbb{R}^{m\times m} \text{ invertible}\\
&W^\top AV = G^\top (AV)^\top (AV)\\
&AV \text{ has rank } m \text{ since } V \text{ has rank } m\\
\Rightarrow\ &W^\top AV \text{ is SPD } \Rightarrow \text{ invertible}
\end{aligned}$$

Optimality results

$$\begin{aligned}
\tilde{\mathbf{x}} &\in \mathbf{x}_0 + \mathcal{K},\\
\tilde{\mathbf{r}} &= \mathbf{b} - A\tilde{\mathbf{x}} \perp \mathcal{L},\\
\delta &= \tilde{\mathbf{x}} - \mathbf{x}_0 \in \mathcal{K},\\
\tilde{\mathbf{r}} &= \mathbf{r}_0 - A\delta \perp \mathcal{L},\\
Ax_\star &= \mathbf{b}.
\end{aligned}$$

(a) If A is SPD and $\mathcal{L} = \mathcal{K}$, then

$$\|\tilde{\mathbf{x}} - x_\star\|_A = \min_{\mathbf{x}\in\mathbf{x}_0+\mathcal{K}} \|\mathbf{x} - x_\star\|_A$$

(b) If A is invertible and $\mathcal{L} = A\mathcal{K}$, then

$$\|\tilde{\mathbf{r}}\|_2 = \|\mathbf{b} - A\tilde{\mathbf{x}}\|_2 = \min_{\mathbf{x}\in\mathbf{x}_0+\mathcal{K}} \|\mathbf{b} - A\mathbf{x}\|_2$$

If these are used iteratively, then:

$$\begin{aligned}
\|x_\star - x_{k+1}\|_A &\le \|x_\star - x_k\|_A\\
\|r_{k+1}\|_2 &\le \|r_k\|_2
\end{aligned}$$

Can we find a constant $C < 1$ such that:

$$\begin{aligned}
\|x_\star - x_{k+1}\|_A &\le C\|x_\star - x_k\|_A\\
\|r_{k+1}\|_2 &\le C\|r_k\|_2
\end{aligned}$$

Example: Steepest Descent

If A is SPD, with $\mathcal{L} = \mathcal{K} = \operatorname{span}\{r_k\}$, then:

$$\begin{aligned}
x_{k+1} &= x_k + \alpha_k r_k, \quad \alpha_k \in \mathbb{R}\\
r_{k+1} &= \mathbf{b} - Ax_{k+1} = r_k - \alpha_k A r_k\\
r_{k+1} &\perp r_k \Rightarrow r_k^\top (r_k - \alpha_k A r_k) = 0 \quad \Rightarrow \alpha_k = \frac{r_k^\top r_k}{r_k^\top A r_k}\\
d_k &= x_\star - x_k\\
r_k &= \mathbf{b} - Ax_k = Ax_\star - Ax_k = Ad_k
\end{aligned}$$

We want to estimate $\|d_{k+1}\|_A \leq C\|d_k\|_A$ for some $C < 1$.

$$\begin{aligned}
r_{k+1} &= \mathbf{b} - Ax_{k+1} = A(x_\star - x_{k+1}) = Ad_{k+1} = Ad_k - \alpha_k Ar_k \\
d_{k+1} &= d_{k+1}^\top Ad_{k+1} = d_{k+1}^\top r_{k+1} \\
&= (d_k - \alpha_k r_k)^\top r_{k+1} = d_k^\top r_{k+1} \\
&= d_k^\top (r_k - \alpha_k Ar_k) = d_k^\top r_k - \alpha_k d_k^\top Ar_k \\
&= d_k^\top Ad_k - \alpha_k r_k^\top r_k \\
&= \|d_k\|_A^2 - \alpha_k \|r_k|^2 \\
&= \|d_k\|_A^2 - \frac{\|r_k|^4}{\|r_k\|_A^2} \\
\|d_{k+1}\|_A^2 &= \|d_k\|_A^2 \left(1 - \frac{\|r_k\|^4}{\|r_k\|_A^2 \|r_k\|_{A^{-1}}^2}\right)
\end{aligned}$$

1.6 Lecture 6: 03.09.2025

Steepest Descent (SD)

Let $A = A^\top > 0$ (SPD). Given $\mathbf{x}_0$ with $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$.

$$\begin{aligned}
\mathcal{K} &= \operatorname{span}\{\mathbf{r}\} \\
\mathcal{L} &= \mathcal{K} \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{r}_k, \quad \alpha_k = \|\mathbf{r}_k\|_2^2 / \mathbf{r}_k^\top A\mathbf{r}_k \\
\mathbf{d}_k &= \mathbf{x}_\star - \mathbf{x}_k \\
\|\mathbf{d}_{k+1}\|_A &\leq \|\mathbf{d}_k\|_A \\
\|\mathbf{d}_{k+1}\|_A^2 &= \|\mathbf{d}_k\|_A^2 \left(1 - \frac{(\mathbf{r}_k^\top \mathbf{r}_k)^2}{\mathbf{r}_k^\top A\mathbf{r}_k \mathbf{r}_k^\top A^{-1}\mathbf{r}_k}\right)
\end{aligned}$$

Using Kantorovich inequality: Let $B \in \mathbb{R}^{n\times n}$ be SPD then for all $\mathbf{x} \in \mathbb{R}^n$:

$$\frac{\|\mathbf{x}\|_B^2 \|\mathbf{x}\|_{B^{-1}}^2}{\|\mathbf{x}\|_2^4} \leq \frac{1}{4} \cdot \frac{(\lambda_1 + \lambda_n)^2}{\lambda_1 \lambda_n}, \qquad \lambda_1 \geq \ldots \geq \lambda_n > 0$$

B is SPD so there exists Q orthogonal and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ such that $B = Q^\top \Lambda Q$. Choose $\|\mathbf{x}\|_2 = 1$ where $\|Q\mathbf{x}\|_2 = \|\mathbf{x}\|_2 = 1$. Then:

$$\begin{aligned}
B^{-1} &= Q^\top \Lambda^{-1} Q \\
\|\mathbf{x}\|_B^2 &= \mathbf{x}^\top B\mathbf{x} = (Q\mathbf{x})^\top \Lambda (Q\mathbf{x}) = \sum_{i=1}^n \lambda_i y_i^2, \quad y = Q\mathbf{x} \\
\|\mathbf{x}\|_{B^{-1}}^2 &= \mathbf{x}^\top B^{-1}\mathbf{x} = (Q\mathbf{x})^\top \Lambda^{-1} (Q\mathbf{x}) = \sum_{i=1}^n \lambda_i^{-1} y_i^2
\end{aligned}$$

$(\bar{\lambda}, \bar{\lambda}^{-1})$ as a weighted discre center of gravity for the point $(\lambda_i, \frac{1}{\lambda_i})$ for $i = 1, \ldots, n$.

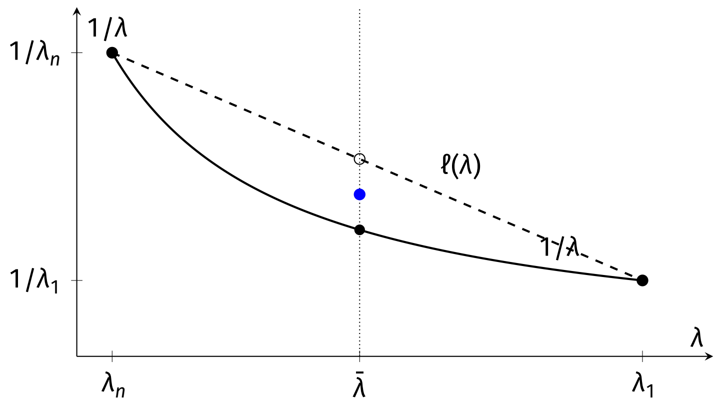
$$\ell(\lambda) = \frac{1}{\lambda_1} + \frac{1}{\lambda_n} - \frac{\lambda}{\lambda_1 \lambda_n}, \qquad \ell(\lambda_1) = \frac{1}{\lambda_1}, \qquad \ell(\lambda_n) = \frac{1}{\lambda_n}$$

Then $(\bar\lambda, \bar\lambda^{-1})$ is below $\ell(\lambda)$:

$$\bar\lambda^{-1} \le \ell(\bar\lambda)$$

which has maximum at $\lambda = \frac{1}{2}(\lambda_1 + \lambda_n)$.

$$\bar\lambda\bar\lambda^{-1} \le \frac{(\lambda_1+\lambda_n)^2}{4\lambda_1\lambda_n} = \bar\lambda\left(\frac{1}{\lambda_1} + \frac{1}{\lambda_n}\right)$$



If A has the eigenvalues $0 < \lambda_1 \le \ldots \le \lambda_n$, then:

$$\begin{aligned}\frac{\|\mathbf{r}_k\|_2^4}{\|\mathbf{r}_k\|_A^2\|\mathbf{r}_k\|_{A^{-1}}^2} &\ge \frac{4\lambda_1\lambda_n}{(\lambda_1+\lambda_n)^2}\\ \|\mathbf{d}_{k+1}\|_A^2 &\le \|\mathbf{d}_k\|_A^2\left(1 - 4\frac{\lambda_1\lambda_n}{(\lambda_1+\lambda_n)^2}\right)\\ &= \|\mathbf{d}_k\|_A^2\left(\frac{\lambda_n-\lambda_1}{\lambda_n+\lambda_1}\right)^2\end{aligned}$$

Example: Discrete Laplacian

$$A = \begin{bmatrix} B & -I & & & 0\\ -I & B & -I & & \\ & -I & \ddots & \ddots & \\ & & \ddots & \ddots & -I\\ 0 & & & -I & B\end{bmatrix} \in \mathbb{R}^{N^2\times N^2}, \begin{bmatrix} 4 & -1 & & 0\\ -1 & 4 & -1 & \\ & -1 & \ddots & \\ 0 & & & 4\end{bmatrix} \in \mathbb{R}^{N\times N}$$

Eigenvalues of A:

$$\lambda_{ij} = 4 - 2\left(\cos\left(\frac{i\pi}{N+1}\right) + \cos\left(\frac{j\pi}{N+1}\right)\right), \quad i,j = 1,\ldots,N$$

$$\begin{aligned}\lambda_{\max} &= 4 \text{ if } N \text{ odd}\\ \lambda_{\min} &= 4 - 4\cos\left(\frac{\pi}{N+1}\right)\\ \frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}} &= \frac{4\cos\left(\frac{\pi}{N+1}\right)}{8 - 4\cos\left(\frac{\pi}{N+1}\right)} \approx 1 - \frac{1}{2}\left(\frac{\pi}{N+1}\right)^2 + \ldots\end{aligned}$$

So for N large, convergence is slow.

Other 1D projection methods

Let $\mathcal{K} = \text{span}\{\mathbf{v}\}$, $\mathcal{L} = \text{span}\{\mathbf{w}\}$. One step, starting from $\mathbf{x}_0$:

$$\begin{aligned} \tilde{\mathbf{x}} &= \mathbf{x}_0 + \alpha \mathbf{v}, \quad \alpha = \frac{\mathbf{w}^\top \mathbf{r}_0}{\mathbf{w}^\top A \mathbf{v}} \\ \tilde{\mathbf{r}} &= \mathbf{b} - A\tilde{\mathbf{x}} = \mathbf{r}_0 - \alpha A \mathbf{v} \end{aligned}$$

if SD: $\mathbf{v} = \mathbf{w} = \mathbf{r}_0$.

Minimim residual (MR)

$\mathbf{v} = \mathbf{r}_0$, $\mathbf{w} = A\mathbf{r}_0$. Converges if

$$\frac{1}{2}\left(A + A^\top\right) > 0 \text{ (SPD)}$$

This is the definition of A being *positive definite*.

$$\begin{aligned} \|\mathbf{r}_{k+1}\|_2^2 &\leq \left(1 - \frac{\mu^2}{\sigma^2}\right) \|\mathbf{r}_k\|_2^2 \\ \mu &= \lambda_{\min}\left(\frac{1}{2}(A + A^\top)\right) \\ \sigma &= \|A\|_2 \end{aligned}$$

If we have the system $A\mathbf{x} = \mathbf{b}$ where A is not positive definite, then we can solve the equivalent system:

$$(A^\top A)\mathbf{x} = A^\top \mathbf{b}$$

and do SD.

$$\begin{aligned} \mathbf{v} &= A^\top \mathbf{r}_0 \\ \mathbf{w} &= A\mathbf{r}_0 \end{aligned}$$

residual norm, steepest descent.

Block Methods

Block methods extend basic iterative techniques to handle systems where variables are grouped into blocks, improving convergence for certain problems.

Block Jacobi

For a matrix A partitioned into blocks A_{ij}, $i, j = 1, \dots, p$, and vectors $\mathbf{x}$ and $\mathbf{b}$ partitioned accordingly:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_p \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_p \end{bmatrix}, \quad V_i = \begin{bmatrix} 0 \\ \vdots \\ I \\ \vdots \\ 0 \end{bmatrix} \text{ (identity at block } i\text{)}$$

The block Jacobi iteration is:

$$\begin{aligned} A_{ii}\tilde{\mathbf{x}}_i &= \mathbf{b}_i - \sum_{j \neq i} A_{ij}\mathbf{x}_j^{(k)} \\ \mathbf{x}_i^{(k+1)} &= A_{ii}^{-1}\left(\mathbf{b}_i - \sum_{j \neq i} A_{ij}\mathbf{x}_j^{(k)}\right), \quad i = 1, \dots, p \end{aligned}$$

Convergence requires diagonal blocks to be invertible and the method to satisfy spectral radius conditions.

Key Takeaways (Exam)

- What is a projection method.
- How can we implement it.
- The optimaly result $\mathcal{L} = \mathcal{K}$ and $\mathcal{L} = A\mathcal{K}$.
- Derive one dimensional projection methods, and how to find convergence results.

1.7 Lecture 9: 09.09.2025

1.7.1 Krylov Subspace Methods (Saad Ch. 6)

Motivation: Solve $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$.

Projection Methods: Given $\mathbf{x}_0$ (initial guess), define the residual $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$. Choose $\mathcal{K}$ and $\mathcal{L}$ subspaces (same dimension) where you want to find

$$\tilde{\mathbf{x}} - \mathbf{x}_0 \in \mathcal{K}, \quad \text{and} \quad \mathbf{b} - A\tilde{\mathbf{x}} \perp \mathcal{L}.$$

One-dimensional methods: (SD, MR)

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{r}_k, \quad \mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k.$$

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{x}_0 + \alpha_0 \mathbf{r}_0, \quad \mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1 = \mathbf{r}_0 - \alpha_0 A\mathbf{r}_0 \\
\mathbf{x}_2 &= \mathbf{x}_1 + \alpha_1 \mathbf{r}_1, \quad \mathbf{r}_2 = \mathbf{b} - A\mathbf{x}_2 = \mathbf{r}_1 - \alpha_1 A\mathbf{r}_1 \\
&\vdots \\
\mathbf{x}_k &= \mathbf{x}_0 + \tilde{\alpha}_0 \mathbf{r}_0 + \tilde{\alpha}_1 A\mathbf{r}_0 + \dots + \tilde{\alpha}_{k-1} A^{k-1}\mathbf{r}_0, \\
&= \mathbf{x}_0 + q_{k-1}(A)\mathbf{r}_0 \\
q_{k-1} &\in \mathbb{P}_{k-1} \\
\mathbf{x}_k &\in \mathbf{x}_0 + \operatorname{span}\{\mathbf{r}_0, A\mathbf{r}_0, \dots, A^{k-1}\mathbf{r}_0\} =: \mathbf{x}_0 + \mathcal{K}_k(A, \mathbf{r}_0).
\end{aligned}$$

We now define the **Krylov subspace**:

Definition 1.10: Krylov Subspace

Given $A : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^n$, the m-th Krylov subspace is

$$\mathcal{K}_m(A, \mathbf{v}) := \operatorname{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \dots, A^{m-1}\mathbf{v}\} = \mathcal{K}_m.$$

Note that $\dim(\mathcal{K}_k(A, \mathbf{v})) \leq k$ and $\dim(\mathcal{K}_k(A, \mathbf{v})) \leq n$.

1.7.2 Important Properties of Krylov Subspaces

1st Property: What is the smallest m s.t. $A\mathcal{K}_m = \mathcal{K}_m$? (i.e. $\mathcal{K}_m$ is invariant under A meaning $A\mathbf{v} \in \mathcal{K}_m$ for all $\mathbf{v} \in \mathcal{K}_m$)

Definition 1.11: minimal polynomial

The minimal polynomial of $\mathbf{v}$ with respect to A is the monic polynomial of the lowest possible degree s.t.

$$A^{\mu}\mathbf{v} + \sum_{i=0}^{\mu-1} d_i A^i \mathbf{v} = p_A(A)\mathbf{v} = 0.$$

μ is the grade of $\mathbf{v}$ with respect to A.

Example 2

Let $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\mathbf{v}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

$$A\mathbf{v}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad A^2\mathbf{v}_1 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \quad A\mathbf{v}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad A^2\mathbf{v}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Then $\text{grade}(\mathbf{v}_1) = 2$ and $\text{grade}(\mathbf{v}_2) = 1$.

2nd Property: is that $\text{grade}(\mathbf{v}) \le n$ where $\mu = \text{grade}(\mathbf{v})$ and n is the size of the matrix A.

1.7.3 Cayley-Hamilton Theorem

Theorem 1.12: Cayley-Hamilton Theorem

Let $A \in \mathbb{R}^{n\times n}$ and

$$p_A(\lambda) = \det(\lambda I - A), \quad p_A \in \mathbb{P}_n$$

be the characteristic polynomial of A. Then

$$p_A(A) = 0.$$

Assume $\mathbf{x} \in \mathcal{K}_m(A, \mathbf{v})$, where $m \ge \mu = \text{grade}(\mathbf{v})$. Then

$$\begin{aligned}
\mathbf{x} &= q_{m-1}(A)\mathbf{v}, \quad q_{m-1} \in \mathbb{P}_{m-1} \\
q(t) &= q_1(t)p_A(t) + q_2(t), \qquad p_A \in \mathbb{P}_{\mu},\ q_2 \in \mathbb{P}_{\mu-1} \\
\mathbf{x} &= q_{m-1}(A)\mathbf{v} \\
&= q_1(A)p_A(A)\mathbf{v} + q_2(A)\mathbf{v} \\
&= q_2(A)\mathbf{v}
\end{aligned}$$

3rd Property: If $\mu = \text{grade}(\mathbf{v})$, then

$$A\mathcal{K}_{\mu} = \mathcal{K}_{\mu}, \quad \text{and} \quad \mathcal{K}_m = \mathcal{K}_{\mu}\ \forall m \ge \mu.$$

4th Property:

$$\dim(\mathcal{K}_m) = \min(m, \text{grade}(\mathbf{v})).$$

If

$$\dim(\mathcal{K}_m) = \dim(\mathcal{L}_m)\begin{cases} \tilde{\mathbf{x}} & \in \mathbf{x} + \mathcal{K}_m \\ \mathbf{b} - A\tilde{\mathbf{x}} & \perp \mathcal{L}_m \end{cases}$$

For simplicity, let $\mathbf{x}_0 = 0$. If $A\mathcal{K}_m = \mathcal{K}_m$, and $\mathbf{b} \in \mathcal{K}_m$, then the exact solution $\mathbf{x}_{\star} = \tilde{\mathbf{x}}$ (independent of $\mathcal{L}_m$)[1].

[1] see lemma 1.36 in Saad

Proof. Let $\tilde{\mathbf{x}} \in \mathcal{K}$, $A\tilde{\mathbf{x}} \in \mathcal{K}$, and $\mathbf{b} \in \mathcal{K} = A\mathcal{K}$. Then

$$
\begin{aligned}
&\mathbf{b} - A\tilde{\mathbf{x}} \in \mathcal{K} \\
&\mathbf{b} - A\tilde{\mathbf{x}} \perp \mathcal{L} \\
&\mathbf{b} - A\tilde{\mathbf{x}} \in \mathcal{K} \cap \mathcal{L}^{\perp} = \{0\} \quad \Rightarrow \quad \mathbf{b} - A\tilde{\mathbf{x}} = 0 \\
&\quad \Leftrightarrow \tilde{\mathbf{x}} = \mathbf{x}_{\star}
\end{aligned}
$$

□ □

Lemma 1: Lemma 1.36 Given two subspaces M and L of the same dimension m, the following two conditions are mathematically equivalent.

1. No nonzero vector of M is orthogonal to L;
2. For any $x \in \mathbb{C}^n$ there is a unique vector u which satisfies the conditions:

$$u \in M \quad x - u \perp L$$

1.7.4 Practical implementation of Krylov Subspace Methods

Let

$$
\begin{aligned}
A\mathbf{x} = \mathbf{b}, \quad \exists \mathbf{x}_0, \quad \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 \\
\mathcal{K}_m(A, \mathbf{r}_0) = \operatorname{span}\{\mathbf{r}_0, A\mathbf{r}_0, \dots, A^{m-1}\mathbf{r}_0\}
\end{aligned}
$$

FOM: Full Orthogonalization Method

$$
\begin{aligned}
\mathcal{K} &= \mathcal{K}_m(A, \mathbf{r}_0) \\
\mathcal{L} &= \mathcal{K} \\
\mathbf{x}_m &= \mathbf{x}_0 + V_m \left(V_m^T A V_m\right)^{-1} V_m^T \mathbf{r}_0 \\
V_m &= [\mathbf{v}_1, \dots, \mathbf{v}_m] \text{ with } V_m^T V_m = I
\end{aligned}
$$

1. How to find an orthogonal basis for $\mathcal{K}_m$?
2. What is $V_m^T A V_m$?
3. When to stop?

$$\|\mathbf{r}_m\|_2 \leq \text{tol}$$

1. Arnoldi Algorithm

What do we get from the Arnoldi algorithm?

$$
\begin{aligned}
V_{m+1} &= [\mathbf{v}_1, \dots, \mathbf{v}_{m+1}] \in \mathbb{R}^{n\times(m+1)}, \quad V_m = [\mathbf{v}_1, \dots, \mathbf{v}_m] \in \mathbb{R}^{n\times m} \\
\overline{H}_m &= (h_{ij}) \in \mathbb{R}^{(m+1)\times m} \text{ upper Hessenberg matrix}, \qquad H_m := \overline{H}_m(1:m, 1:m) \in \mathbb{R}^{m\times m}
\end{aligned}
$$

s.t.

$$
\begin{aligned}
AV_m &= V_{m+1}\overline{H}_m = V_m H_m + h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^T, \\
V_m^T A V_m &= H_m.
\end{aligned}
$$

Using the Galerkin condition for FOM (take $\mathcal{L} = \mathcal{K}_m$) we obtain the small system

$$H_m \mathbf{y}_m = V_m^T \mathbf{r}_0 = \beta \mathbf{e}_1, \qquad \beta = \|\mathbf{r}_0\|_2,$$

Algorithm 1 Arnoldi Algorithm

Require:
  $A \in \mathbb{R}^{n\times n}$
  $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|_2}$
Ensure:
  $V_{m+1} = [\mathbf{v}_1, \dots, \mathbf{v}_{m+1}]$ orthonormal basis for $\mathcal{K}_{m+1}(A, \mathbf{r}_0)$
  $\overline{H}_m = (h_{ij}) \in \mathbb{R}^{(m+1)\times m}$ upper Hessenberg matrix
  for $j = 1, 2, \dots, m$ **do**
    Compute $w = A\mathbf{v}_j$
    for $i = 1, \dots, j$ **do**
      $h_{ij} = \langle w, \mathbf{v}_i\rangle$
      $w = w - h_{ij}\mathbf{v}_i$
    $h_{j+1,j} = \|w\|_2$
    if $h_{j+1,j} = 0$ **then**
      Stop (breakdown)
    $\mathbf{v}_{j+1} = w/h_{j+1,j}$

so

$$\mathbf{x}_m = \mathbf{x}_0 + V_m\mathbf{y}_m, \qquad \mathbf{y}_m = H_m^{-1}(\beta\mathbf{e}_1).$$

The residual can be computed cheaply from the Arnoldi relation:

$$\begin{aligned}\mathbf{r}_m &= \mathbf{r}_0 - AV_m\mathbf{y}_m = \beta\mathbf{v}_1 - V_{m+1}\overline{H}_m\mathbf{y}_m \\ &= \beta\mathbf{v}_1 - V_mH_m\mathbf{y}_m - h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^\top\mathbf{y}_m = -h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^\top\mathbf{y}_m,\end{aligned}$$

since $H_m\mathbf{y}_m = \beta\mathbf{e}_1$. Hence

$$\|\mathbf{r}_m\|_2 = |h_{m+1,m}|\,|\mathbf{e}_m^\top\mathbf{y}_m|.$$

Thus we get the FOM algorithm (Arnoldi performed incrementally; solve the small system at each step and check residual):

Algorithm 2 Full Orthogonalization Method (FOM)

Require:
  $A \in \mathbb{R}^{n\times n}, \mathbf{b} \in \mathbb{R}^n, \mathbf{x}_0 \in \mathbb{R}^n, m_{\max} \in \mathbb{N}, \mathrm{tol} > 0$
  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0, \quad \beta = \|\mathbf{r}_0\|_2$
  $\mathbf{v}_1 = \mathbf{r}_0/\beta$
Ensure:
  $\mathbf{x}_j$ approximations, stop when converged or breakdown
  for $j = 1, 2, \dots, m_{\max}$ **do**
    Perform one Arnoldi step to compute $h_{1:j+1,j}$ and $\mathbf{v}_{j+1}$ (see Alg. 1)
    Let $H_j = \overline{H}_j(1:j, 1:j)$ and $V_j = [\mathbf{v}_1, \dots, \mathbf{v}_j]$
    Solve $H_j\mathbf{y}_j = \beta\mathbf{e}_1$
    $\mathbf{x}_j = \mathbf{x}_0 + V_j\mathbf{y}_j$
    $\mathbf{r}_j = -h_{j+1,j}\mathbf{v}_{j+1}\mathbf{e}_j^\top\mathbf{y}_j$
    if $\|\mathbf{r}_j\|_2 \le \mathrm{tol}$ **then**
      Return $\mathbf{x}_j$
    if $h_{j+1,j} = 0$ **then**
      Breakdown: exact solution in $\mathcal{K}_j$ (stop)

1.8 Lecture 10: 10.09.2025

1.8.1 Krylov space

$$\mathcal{K}_m(A, \mathbf{v}) = \text{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \dots, A^{m-1}\mathbf{v}\}$$

Arnoldi Algorithm

Algorithm 3 Arnoldi Algorithm

$\mathbf{v}_1 = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$
for $j = 1, 2, \dots, m$ **do**
  $\mathbf{w}_j = A\mathbf{v}_j$
  for $i = 1, 2, \dots, j$ **do**
    $h_{ij} = \langle \mathbf{v}_i, \mathbf{w}_j \rangle$
    $\mathbf{w}_j = \mathbf{w}_j - h_{ij}\mathbf{v}_i$
  $h_{j+1,j} = \|\mathbf{w}_j\|_2$
  if $h_{j+1,j} = 0$ **then**
    Stop
  $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}}$

Out of the algorithm we get:

$$\begin{aligned}
V_{m+1} &= [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{m+1}] \in \mathbb{R}^{n\times(m+1)} \\
V_{m+1}^\top V_{m+1} &= 0 \\
\overline{H}_m &= \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ 0 & h_{3,2} & \cdots & h_{3,m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{m+1,m} \end{bmatrix} = \begin{bmatrix} H_m \\ 0 \end{bmatrix} \in \mathbb{R}^{(m+1)\times m}
\end{aligned}$$

With the relations:

$$\begin{aligned}
AV_m &= V_{m+1}\overline{H}_m = V_m H_m + h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^\top \\
V_m^\top A V_m &= H_m
\end{aligned}$$

1.8.2 FOM: Full Orthogonalization Method

Let $\mathcal{L}_m = \mathcal{K}_m(A, \mathbf{r}_0)$, where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$. Find $\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{L}_m$ such that

$$\begin{aligned}
\mathbf{x}_m &= \mathbf{x}_0 + V_m^\top \mathbf{y}_m \\
\mathbf{y}_m &= \beta H_m^{-1}\mathbf{e}_1 \\
\beta &= \|\mathbf{r}_0\|_2 \\
\|\mathbf{r}_m\|_2 &= |h_{m+1,m}||\mathbf{e}_m^\top \mathbf{y}_m|
\end{aligned}$$

Complexity:

- Arnoldi:
  - 1 Av per iteration: $\mathcal{O}(N_z(A) \cdot m)$ flops.

- Inner products and update of $\mathbf{w}$: $\mathcal{O}(nm)$ flops.
- Sol. of $H_m \mathbf{y}_m = \beta \mathbf{e}_1$: $\mathcal{O}(m^2)$ flops.
- Total $V_m^\top \mathbf{y}_m$: $\mathcal{O}(nm)$ flops.

Remedies:

- Restart after a given m iterations: $\mathbf{x}_0 \leftarrow \mathbf{x}_m$.
- Ortogonalize only towards the last k vectors of $\mathbf{v}_j$.
- incomplete orthogonalization.

1.8.3 GMRES (Generalized Minimum Residual Method)

Let $\mathcal{K} = \mathcal{K}_m(A, \mathbf{r}_0)$, and $\mathcal{L}_m = A\mathcal{K}$.

$$
\begin{aligned}
\mathbf{r}_m &= \mathbf{b} - A\mathbf{x}_m \\
\|\mathbf{b} - A\mathbf{x}_m\|_2 &= \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m} \|\mathbf{b} - A\mathbf{x}\|_2 \\
\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m \quad &\Rightarrow \quad \mathbf{x} = \mathbf{x}_0 + V_m \mathbf{y}_m, \quad \mathbf{y}_m \in \mathbb{R}^m \\
\mathbf{r} = \mathbf{b} - A\mathbf{x} &= \mathbf{b} - A(\mathbf{x}_0 + V_m \mathbf{y}_m) = \mathbf{r}_0 - AV_m \mathbf{y}_m \\
&= \mathbf{r}_0 - V_{m+1} \overline{H}_m \mathbf{y}_m \\
&= V_{m+1}(\beta \mathbf{e}_1 - \overline{H}_m \mathbf{y}_m) \\
\|\mathbf{r}\|^2 &= \|V_{m+1}(\beta \mathbf{e}_1 - \overline{H}_m \mathbf{y}_m)\|_2 = \|\beta \mathbf{e}_1 - \overline{H}_m \mathbf{y}_m\|_2, \quad \text{since } \|V_{m+1}\|_2 = 1 \text{ (orthonormal columns)} \\
\mathbf{y}_m &= \arg\min_{\mathbf{y} \in \mathbb{R}^m} \|\beta \mathbf{e}_1 - \overline{H}_m \mathbf{y}\|_2 \\
\mathbf{x}_m &= \mathbf{x}_0 + V_m \mathbf{y}_m
\end{aligned}
$$

Want to solve the overdetermined system:

$$\overline{H}_m \mathbf{y} \approx \beta \mathbf{e}_1$$

We solve this least squares problem using QR factorization of $\overline{H}_m$ with Givens rotations.

QR Factorization Approach

Since $\overline{H}_m \in \mathbb{R}^{(m+1)\times m}$ is upper Hessenberg, we can efficiently compute its QR factorization using Givens rotations. Let

$$\overline{H}_m = Q_{m+1} R_m$$

where $Q_{m+1} \in \mathbb{R}^{(m+1)\times(m+1)}$ is orthogonal and $R_m \in \mathbb{R}^{(m+1)\times m}$ has the structure:

$$\tilde{R}_m = \begin{bmatrix} R_m \\ \mathbf{0}^\top \end{bmatrix}$$

with $R_m \in \mathbb{R}^{m\times m}$ upper triangular.

Let

$$\bar{\mathbf{g}}_m = Q_{m+1}^\top \beta \mathbf{e}_1 = [\gamma_1, \gamma_2, \dots, \gamma_{m+1}]^\top$$

The least squares problem becomes:

$$
\begin{aligned}
Q_m \left(\beta \mathbf{e}_1 - \overline{H}_m \mathbf{y}\right) = \beta Q_m \mathbf{e}_1 - Q_m \overline{H}_m \mathbf{y} &= \overbrace{\beta Q_m \mathbf{e}_1}^{\bar{\mathbf{g}}_m} - \begin{bmatrix} R_m \\ \mathbf{0}^\top \end{bmatrix} \mathbf{y}_m \\
&= \begin{bmatrix} \mathbf{g}_{1:m} \\ g_{m+1} \end{bmatrix} - \begin{bmatrix} R_m \\ \mathbf{0}^\top \end{bmatrix} \mathbf{y}_m \\
&= \begin{bmatrix} \mathbf{g}_{1:m} - R_m \mathbf{y}_m \\ g_{m+1} \end{bmatrix}
\end{aligned}
$$

Then:

$$\begin{aligned}\|\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y}\|^2 = \|\bar{\mathbf{g}}_m - \tilde{R}_m\mathbf{y}\|^2 &= \|\mathbf{g}_{1:m} - R_m\mathbf{y}\|^2 + |g_{m+1}|^2 \\ \mathbf{y}_m &= R_m^{-1}\mathbf{g}_{1:m} \\ \|\mathbf{r}_m\|_2 &= |\gamma_{m+1}|\end{aligned}$$

Then we do QR factorization by Givens rotations:

$$\begin{aligned} h &= \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}, \quad \Omega = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad c^2 + s^2 = 1 \\ \Omega h &= \begin{bmatrix} \|h\| \\ 0 \end{bmatrix} \\ \begin{bmatrix} c & s \\ -s & c \end{bmatrix}\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} &= \begin{bmatrix} r \\ 0 \end{bmatrix} \\ \Rightarrow \quad \|h\| &= \sqrt{h_1^2 + h_2^2}, \quad c = \frac{h_1}{r}, \quad s = \frac{h_2}{r} \end{aligned}$$

In the Arnoldi process for $k = 1$:

$$\begin{aligned} H_1 &= \begin{bmatrix} h_{1,1} \\ h_{2,1} \end{bmatrix} \xrightarrow{\Omega_1} \begin{bmatrix} \tilde{h}_{1,1} \\ 0 \end{bmatrix} \\ \beta\mathbf{e}_1 &= \begin{bmatrix} \beta \\ 0 \end{bmatrix} \xrightarrow{\Omega_1} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \end{aligned}$$

For $k = 2$:

$$\begin{aligned} H_2 &= \begin{bmatrix} \tilde{h}_{1,1} & h_{1,2} \\ 0 & h_{2,2} \\ 0 & h_{3,2} \end{bmatrix} \xrightarrow{\Omega_2} \begin{bmatrix} \tilde{h}_{1,1} & \tilde{h}_{1,2} \\ 0 & \tilde{h}_{2,2} \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} &\xrightarrow{\Omega_2} \begin{bmatrix} \gamma_1 \\ \tilde{\gamma}_2 \\ \tilde{\gamma}_3 \end{bmatrix} \end{aligned}$$

after m iterations we have:

$$\begin{aligned} \tilde{R}_m &= \begin{bmatrix} \tilde{h}_{1,1} & \tilde{h}_{1,2} & \cdots & \tilde{h}_{1,m} \\ 0 & \tilde{h}_{2,2} & \cdots & \tilde{h}_{2,m} \\ 0 & 0 & \cdots & \tilde{h}_{3,m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \\ \bar{\mathbf{g}}_m &= \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{m+1} \end{bmatrix} = \begin{bmatrix} g_{1:m} \\ g_{m+1} \end{bmatrix} \end{aligned}$$

Afer k iterates:

$$\begin{bmatrix} h_{1,k} \\ h_{2,k} \\ \vdots \\ h_{k,k} \\ h_{k+1,k} \end{bmatrix} \xrightarrow{\Omega_k} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_k \\ 0 \end{bmatrix}$$

before applying Givens rotations.

$$\|r_{k-1}\| = |\gamma_k|$$

Then Givens:

$$\begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \begin{bmatrix} \gamma_k \\ 0 \end{bmatrix} = \begin{bmatrix} c_k\gamma_k \\ -s_k\gamma_k \end{bmatrix}$$
$$\|r_k\| = |-s_k\gamma_k| = |s_k|\,\|r_{k-1}\|$$

Then

$$|s_k| \le 1$$
$$\text{If } |s_k| < 1, \text{ then } \|r_k\| < \|r_{k-1}\|$$
$$\text{If } |s_k| = 1, \text{ then stagnation, but then } c_k = 0 \text{ which means } h_{k,k} = 0 \text{ or } A \text{ is singular.}$$
$$c_k = \frac{h_{k,k}}{\sqrt{h_{k,k}^2 + h_{k+1,k}^2}}, \qquad s_k = \frac{h_{k+1,k}}{\sqrt{h_{k,k}^2 + h_{k+1,k}^2}}$$

GMRES Algorithm

Algorithm 4 GMRES Algorithm

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
$\beta = \|\mathbf{r}_0\|_2$
$\mathbf{v}_1 = \frac{\mathbf{r}_0}{\beta}$
for $j = 1, 2, \dots, m$ **do**
  $\mathbf{w}_j = A\mathbf{v}_j$
  for $i = 1, 2, \dots, j$ **do**
    $h_{ij} = \langle \mathbf{w}_j, \mathbf{v}_i \rangle$
    $\mathbf{w}_j = \mathbf{w}_j - h_{ij}\mathbf{v}_i$
  $h_{j+1,j} = \|\mathbf{w}_j\|_2$
  if $h_{j+1,j} = 0$ **then Stop**
  $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}}$
$V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m] \in \mathbb{R}^{n\times m}$
$V_m^T V_m = I$
$H_m \in \mathbb{R}^{m\times m}$
$\overline{H}_j \in \mathbb{R}^{(m+1)\times m}$ (upper Hessenberg matrix)
Compute minimizer $\mathbf{y}_m$ of $\|\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y}\|_2$
$\mathbf{x}_m = \mathbf{x}_0 + V_m\mathbf{y}_m$ (Solution)

1.9 Lecture 11: 16.09.2025

Go from Arnoldi → *Lanczos* (symmetric case) → *conjugate gradient* (CG).

We first start with the assumption that A is *symmetric and positive definite* (SPD), i.e., $A = A^T > 0$.

1.9.1 Recap: Arnoldi iteration

Algorithm 5 Arnoldi iteration where A is SPD

Require: $A, \mathbf{b}, \mathbf{x}_0, m$
  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
  $\beta = \|\mathbf{r}_0\|_2$
  $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\beta}$
  for $j = 1, 2, \dots, m$ **do**
    $h_{ij} = \langle A\mathbf{v}_j, \mathbf{v}_i \rangle$ for $i = 1, 2, \dots, j$
    $\mathbf{w}_j = A\mathbf{v}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i$
    $h_{j+1,j} = \|\mathbf{w}_j\|_2$
    if $h_{j+1,j} = 0$ **then**
      Stop
    $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}}$
    return $V_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$, $\bar{H}_m = \begin{bmatrix} H_m \\ h_{m+1,m}\mathbf{e}_m^T \end{bmatrix}$

Then we have the Arnoldi relation

$$\begin{aligned} AV_m &= V_{m+1}\bar{H}_m \\ V_m^\top A V_m &= H_m \end{aligned}$$

Where we solve the reduced linear system:

$$\begin{aligned} \mathbf{x}_m &= \mathbf{x}_0 + V_m H_m^{-1} V_m^\top \mathbf{r}_0 \\ &= \mathbf{x}_0 + V_m H_m^{-1} \beta \mathbf{e}_1, \quad \beta = \|\mathbf{r}_0\|_2 \\ \mathbf{x}_m &= \mathbf{x}_0 + V_m \mathbf{y}_m \end{aligned}$$

How can this be simplified if $A = A^\top$?

In this case $H_m = V_m^\top A V_m = H_m^\top$ is symmetric, and since it is upper Hessenberg it must be tridiagonal. H_m is then tridiagonal and symmetric, i.e., H_m has the form:

$$H_m = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & \cdots & 0 \\ 0 & \beta_3 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \beta_m \\ 0 & 0 & 0 & \beta_m & \alpha_m \end{bmatrix}$$

1.9.2 Lanczos iteration

Algorithm 6 Lanczos: Arnoldi for symmetric $A = A^{\mathsf{T}}$

Require: $A, \mathbf{b}, \mathbf{x}_0, m$

$\beta_1 = 0$
$\mathbf{v}_0 = 0$
$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
$\beta = \|\mathbf{r}_0\|_2$
$\mathbf{v}_1 = \dfrac{\mathbf{r}_0}{\beta}$
for $j = 1, 2, \dots, m$ **do**
  $\mathbf{w}_j = A\mathbf{v}_j - \beta_j \mathbf{v}_{j-1}$, where $\beta_1 \mathbf{v}_0 = 0$
  $\alpha_j = \langle \mathbf{w}_j, \mathbf{v}_j \rangle$
  $\mathbf{w}_j = \mathbf{w}_j - \alpha_j \mathbf{v}_j$
  $\beta_{j+1} = \|\mathbf{w}_j\|_2$
  if $\beta_{j+1} = 0$ **then Stop**
  $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{\beta_{j+1}}$
return $V_{m+1} = [\mathbf{v}_1, \dots, \mathbf{v}_{m+1}]$
$T_m = \text{tridiag}(\beta_i, \alpha_i, \beta_{i+1}), \quad i = 1, \dots, m$
$\mathbf{x}_m = \mathbf{x}_0 + V_m T_m^{-1} \beta \mathbf{e}_1$
Solve: $T_m \mathbf{y}_m = \beta \mathbf{e}_1$

We solve the tridiagonal system:

$$T_m \mathbf{y}_m = \beta \mathbf{e}_1$$

using LU factorization:

$$T_m = L_m U_m$$

$$\begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & \cdots & 0 \\ 0 & \beta_3 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \beta_m \\ 0 & 0 & 0 & \beta_m & \alpha_m \end{bmatrix} = \overbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \lambda_2 & 1 & 0 & \cdots & 0 \\ 0 & \lambda_3 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \lambda_m & 1 \end{bmatrix}}^{L_m} \overbrace{\begin{bmatrix} \eta_1 & \beta_2 & 0 & \cdots & 0 \\ 0 & \eta_2 & \beta_3 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \beta_m \\ 0 & 0 & 0 & 0 & \eta_m \end{bmatrix}}^{U_m}$$

Now we rewrite the approximation using L_m and U_m:

$$\mathbf{x}_m = \mathbf{x}_0 + \underbrace{V_m U_m^{-1}}_{P_m} \underbrace{L_m^{-1} \beta \mathbf{e}_1}_{\mathbf{z}_m}, \quad \mathbf{z}_m = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_m \end{bmatrix}, \quad P_m = [\mathbf{p}_1, \dots, \mathbf{p}_m]$$

$$\begin{aligned} L_m \mathbf{z}_m &= \beta \mathbf{e}_1 \\ \zeta_1 &= \beta \\ \lambda_2 \zeta_1 + \zeta_2 &= 0 \\ &\vdots \\ \lambda_{i+1} \zeta_i + \zeta_{i+1} &= 0, \quad i = 1, \dots, m-1 \end{aligned}$$

$$
\begin{aligned}
P_m U_m &= V_m \\
\eta_1 \mathbf{p}_1 &= \mathbf{v}_1 \\
\beta_2 \mathbf{p}_1 + \eta_2 \mathbf{p}_2 &= \mathbf{v}_2 \\
&\vdots \\
\beta_i \mathbf{p}_{i-1} + \eta_i \mathbf{p}_i &= \mathbf{v}_i, \quad i = 2, \dots, m \\
\mathbf{p}_i = \frac{1}{\eta_i}(\mathbf{v}_i - \beta_i \mathbf{p}_{i-1})
\end{aligned}
$$

Then

$$
\begin{aligned}
\mathbf{x}_m &= \mathbf{x}_0 + P_m \mathbf{z}_m \\
&= \mathbf{x}_0 + \sum_{i=1}^{m} \mathbf{p}_i \zeta_i = \mathbf{x}_0 + \sum_{i=1}^{m-1} \mathbf{p}_i \zeta_i + \mathbf{p}_m \zeta_m \\
&= \mathbf{x}_{m-1} + \zeta_m \mathbf{p}_m
\end{aligned}
$$

If we incorporate this into the Lanczos algorithm we get the *conjugate gradient* (CG) method.

1.9.3 Conjugate gradient (CG) method

Proposition 1

$$
\begin{aligned}
\mathbf{r}_j &= \mathbf{b} - A\mathbf{x}_j, \quad j = 0, 1, \dots, m \\
\mathbf{p}_j &= \frac{1}{\eta_j}(\mathbf{v}_j - \beta_j \mathbf{p}_{j-1}), \quad j = 1, 2, \dots, m
\end{aligned}
$$

Then:
(a) $\langle \mathbf{r}_i, \mathbf{r}_j \rangle = 0$ for $i \neq j$ (residuals are orthogonal)
(b) $\langle \mathbf{p}_i, A\mathbf{p}_j \rangle = 0$ for $i \neq j$ (A-orthogonal search directions)

For a) The residual:

$$
\begin{aligned}
\mathbf{r}_j &= \mathbf{b} - A\mathbf{x}_j \\
&= -\beta_{j+1} \mathbf{e}_j^\top \mathbf{y}_j \mathbf{v}_{j+1}, \quad j = 1, 2, \dots, m \\
&= \sigma \mathbf{v}_{j+1}, \quad \sigma = -\beta_{j+1} \mathbf{e}_j^\top \mathbf{y}_j
\end{aligned}
$$

Since $\mathbf{v}_j$ are orthogonal by construction, so are the residuals $\mathbf{r}_j$ for $j = 0, 1, \dots, m$.

For b) We have

$$
\begin{aligned}
P_m &= \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_m \end{bmatrix} \\
P_m^\top A P_m &= D \text{ (diagonal)} \\
U_m^{-\top} \overbrace{V_m^\top A V_m}^{T_m = L_m U_m} U_m^{-1} &= D \\
P_m^\top A P_m &= U_m^{-\top} L_m U_m U_m^{-1} = U_m^{-\top} L_m = D
\end{aligned}
$$

Obviously, $P_m^\top A P_m$ is symmetric.

- $U_m^{-\top}$ and L_m are lower bidiagonal:

$$U_m^{-\top} = \begin{bmatrix} \frac{1}{\eta_1} & 0 & 0 & \cdots & 0 \\ -\frac{\beta_2}{\eta_1\eta_2} & \frac{1}{\eta_2} & 0 & \cdots & 0 \\ 0 & -\frac{\beta_3}{\eta_2\eta_3} & \frac{1}{\eta_3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & -\frac{\beta_m}{\eta_{m-1}\eta_m} & \frac{1}{\eta_m} \end{bmatrix}, \quad L_m = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \lambda_2 & 1 & 0 & \cdots & 0 \\ 0 & \lambda_3 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \lambda_m & 1 \end{bmatrix}$$

- $U_m^{-\top} L_m$ is lower triangular:

$$U_m^{-\top} L_m = \begin{bmatrix} \frac{1}{\eta_1} & 0 & 0 & \cdots & 0 \\ -\frac{\beta_2}{\eta_1\eta_2} & \frac{1}{\eta_2} & 0 & \cdots & 0 \\ 0 & -\frac{\beta_3}{\eta_2\eta_3} & \frac{1}{\eta_3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & -\frac{\beta_m}{\eta_{m-1}\eta_m} & \frac{1}{\eta_m} \end{bmatrix}$$

- So: A lower triangular symmetric matrix is diagonal.

$$P_m^\top A P_m = U_m^{-\top} L_m = D$$

$$\begin{aligned} \mathbf{x}_m &= \mathbf{x}_0 + V_m \left(V_m^\top A V_m\right)^{-1} V_m^\top \mathbf{r}_0 \\ &= \mathbf{x}_0 + V_m T_m^{-1} \beta \mathbf{e}_1, \quad \beta = \|\mathbf{r}_0\|_2 \\ &= \mathbf{x}_0 + P_m \mathbf{z}_m = \mathbf{x}_{m-1} + \zeta_m \mathbf{p}_m \\ T_m &= L_m U_m \\ P_m &= V_m U_m^{-1} \\ \mathbf{z}_m &= L_m^{-1} \beta \mathbf{e}_1 \end{aligned}$$

For each iteration j with $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = \mathbf{p}_0$:

$$\begin{aligned} \mathbf{x}_{j+1} &= \mathbf{x}_j + \alpha_j \mathbf{p}_j \Rightarrow \mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j A \mathbf{p}_j \\ \mathbf{p}_{j+1} &= \mathbf{r}_{j+1} + \beta_j \mathbf{p}_j \end{aligned}$$

We know that:

$$\begin{aligned} \langle \mathbf{r}_{j+1}, \mathbf{r}_j \rangle = 0 &\Rightarrow \alpha_j = \frac{\langle \mathbf{r}_j, \mathbf{r}_j \rangle}{\langle A\mathbf{p}_j, \mathbf{p}_j \rangle} = \frac{\|\mathbf{r}_j\|_2^2}{\langle \mathbf{p}_j, A\mathbf{p}_j \rangle} \\ \langle \mathbf{r}_{j+1}, \mathbf{r}_j \rangle = 0 &\Rightarrow \beta_j = \frac{\langle \mathbf{r}_{j+1}, \mathbf{r}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{r}_j \rangle} = \frac{\|\mathbf{r}_{j+1}\|_2^2}{\|\mathbf{r}_j\|_2^2} \end{aligned}$$

Then the CG algorithm is:

Algorithm 7 Conjugate gradient (CG) method

Require: $A, \mathbf{b}, \mathbf{x}_0, m$
  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
  $\mathbf{p}_0 = \mathbf{r}_0$
  for $j = 0, 1, \dots, m-1$ **do**
    $\alpha_j = \dfrac{\|\mathbf{r}_j\|_2^2}{\langle \mathbf{p}_j, A\mathbf{p}_j \rangle}$
    $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j$
    $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j A \mathbf{p}_j$
    $\beta_{j+1} = \dfrac{\|\mathbf{r}_{j+1}\|_2^2}{\|\mathbf{r}_j\|_2^2}$
    $\mathbf{p}_{j+1} = \mathbf{r}_{j+1} + \beta_j \mathbf{p}_j$
    if $\|\mathbf{r}_{j+1}\|_2 < \text{tol}$ **then Stop**
  return $\mathbf{x}_m$

Complexity. For every iteration j we need to compute:

1. One matrix-vector product $A\mathbf{p}_j$ (if A is sparse, $\mathcal{O}(Nz(A))$) ($Nz(A)$ = number of nonzeros elements in A)
2. 3 vector updates (axpy), $\mathcal{O}(n)$
3. 2 inner products, $\mathcal{O}(n)$

Total: $m \cdot \mathcal{O}(Nz(A) + n) = \mathcal{O}(m \cdot Nz(A) + m \cdot n)$ for m iterations.

Memory. We need to store $(\mathbf{x}_j, \mathbf{r}_j, \mathbf{p}_j)$, i.e., $3n$ entries, and A (if sparse, $\mathcal{O}(Nz(A))$).

Relation to Orthogonal polynomials.

$$
\begin{aligned}
\langle f, g \rangle &= \int_a^b w(x) f(x) g(x)\, dx, \quad w(x) > 0 \text{ (weight function)} \\
p_0(x) &= 1 \\
p_1(x) &= x \\
p_n(x) &= (x - a_n) p_{n-1}(x) - b_n p_{n-2}(x), \quad n \geq 2 \\
a_n &= \frac{\langle x p_{n-1}, p_{n-1} \rangle}{\langle p_{n-1}, p_{n-1} \rangle} \\
b_n &= \frac{\langle x p_{n-2}, p_{n-2} \rangle}{\langle p_{n-2}, p_{n-2} \rangle}
\end{aligned}
$$

1.10 Lecture 12: 17.09.2025

Projection idea: Find $\mathbf{x}_m - \mathbf{x}_0 \in \mathcal{K}_m$, with $\mathbf{b} - A\mathbf{x}_m \perp \mathcal{L}_m$ for some subspace $\mathcal{L}_m$.

- A is SPD, $\mathcal{L}_m = \mathcal{K}_m \implies$ CG method.

$$\|\mathbf{x}_\star - \mathbf{x}_m\|_A = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m} \|\mathbf{x}_\star - \mathbf{x}\|_A$$

- A is general, $\mathcal{L}_m = A\mathcal{K}_m \implies$ GMRES method.

$$\|\mathbf{b} - A\mathbf{x}_m\|_2 = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m} \|\mathbf{b} - A\mathbf{x}\|_2$$

What we want:

$$\|\mathbf{x}_\star - \mathbf{x}_m\|_A \leq C_m \|\mathbf{x}_\star - \mathbf{x}_0\|_A$$
$$\|\mathbf{b} - A\mathbf{x}_m\|_A \leq \tilde{C}_m \|\mathbf{b} - A\mathbf{x}_0\|_A$$

where $\mathbf{x} \in \mathbf{x}_0 \in \mathcal{K}_m$, $\mathbf{x} = \mathbf{x}_0 + q_m(A)\mathbf{r}_0$ where $q_m \in \mathbb{P}_{m-1}$.

$$\begin{aligned}
\mathbf{x}_\star - \mathbf{x}_m &= \mathbf{x}_\star - \mathbf{x}_0 - q_m(A)\mathbf{r}_0 = (I - Aq_m(A))(\mathbf{x}_\star - \mathbf{x}_0) \\
&= p_m(A)(\mathbf{x}_\star - \mathbf{x}_0) \quad \text{where } p_m \in \mathbb{P}_m,\ p_m(0) = 1 \\
\mathbf{r} &= \mathbf{b} - A\mathbf{x} = \mathbf{b} - A(\mathbf{x}_0 + q_m(A)\mathbf{r}_0) \\
&= (I - Aq_m(A))\mathbf{r}_0
\end{aligned}$$

For the residual:

$$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = A(\mathbf{x}_\star - \mathbf{x}_0)$$

We have:

$$\|\mathbf{x} - \mathbf{x}_m\|_A = \|p_m(A)(\mathbf{x}_\star - \mathbf{x}_0)\|_A = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m} \|p_m(A)(\mathbf{x}_\star - \mathbf{x}_0)\|_A$$
$$\|\mathbf{b} - A\mathbf{x}_m\|_2 = \|p_m(A)\mathbf{r}_0\|_2 = \min_{\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m} \|p_m(A)\mathbf{r}_0\|_2$$

Consider only the CG case (A SPD): Then the eigenvalues are $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ and a full set of orthogonal eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ s.t.

$$V = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad V^T V = I$$

and

$$\begin{aligned}
A\mathbf{v}_i &= \lambda_i \mathbf{v}_i, \quad i = 1, \ldots, n \\
p(A)\mathbf{v}_i &= p(\lambda_i)\mathbf{v}_i
\end{aligned}$$

Then we define $\mathbf{y} \in \mathbb{R}^n$ s.t.

$$\begin{aligned}
\mathbf{y} = V\boldsymbol{\alpha} &= \sum_{i=1}^n \boldsymbol{\alpha}_i \mathbf{v}_i \\
\|\mathbf{y}\|_A^2 &= \sum_{i=1}^n \lambda_i \boldsymbol{\alpha}_i^2 \\
\|\mathbf{y}\|_A^2 &= \mathbf{y}^\top A\mathbf{y} = \mathbf{y}^\top V\Lambda V^\top \mathbf{y} \\
\|p(A)\mathbf{y}\|_A^2 &= \sum_{i=1}^n p(\lambda_i)^2 \lambda_i \boldsymbol{\alpha}_i^2
\end{aligned}$$

If $\mathbf{x}_\star - \mathbf{x}_0 = \sum_{i=1}^n \xi_i \mathbf{v}_i$, then:

$$\|\mathbf{x}_\star - \mathbf{x}_m\|_A^2 = \sum_{i=1}^n p_m(\lambda_i)^2 \lambda_i \xi_i^2$$

And p_m is the solution of:

$$\min_{p_m \in \mathbb{P}_m, p_m(0)=1} \max_{1 \leq i \leq n} |p_m(\lambda_i)|$$

Chebyshev polynomials:

$$C_k(t) = \cos(k \arccos(t)) = \frac{1}{2}\left(\left(t - \sqrt{t^2-1}\right)^k + \left(t + \sqrt{t^2-1}\right)^k\right), \quad |t| \geq 1$$

$$C_0(t) = 1, \quad C_1(t) = t, \quad C_{k+1}(t) = 2tC_k(t) - C_{k-1}(t)$$

They are orthogonal on the inner product:

$$\langle f, g\rangle = \int_{-1}^{1} \frac{1}{\sqrt{1-t^2}} f(t)g(t)\, dt$$

We will search for a polynomial $p \in \mathbb{P}_m$ s.t. $p(0) = 1$ satisfying:

$$p^\star = \arg \min_{\substack{p \in \mathbb{P}_m \\ p(0)=1}} \max_{\lambda_{\min} \leq \lambda \leq \lambda_{\max}} |p(\lambda)|$$

1.10.1 Theorem (Saad 6.11.4)

$C_k(t)$ is the solution of:

$$\min_{\substack{p \in \mathbb{P}_k \\ p(0)=1}} \max_{-1 \leq t \leq 1} |p(t)|$$

Map $[-1, 1] \to [\lambda_{\min}, \lambda_{\max}]$ by scaling it so $p_k(0) = 1$:

$$p_k(\lambda) = \frac{C_k\left(\frac{2\lambda - \lambda_{\max} - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}\right)}{C_k\left(\frac{-\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}\right)}, \quad \lambda \in [\lambda_{\min}, \lambda_{\max}]$$

Then the maximum value of $|p_k(\lambda)|$ on $[\lambda_{\min}, \lambda_{\max}]$ is:

$$|p_k(\lambda)| \leq \frac{1}{|C_k\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)|}$$

Thus we have found the bound for $\|\mathbf{x}_\star - \mathbf{x}_m\|_A = \|p_m(A)(\mathbf{x}_\star - \mathbf{x}_0)\|_A$:

$$\begin{aligned}
\|\mathbf{x}_\star - \mathbf{x}_m\|_A &\leq \frac{1}{|C_m\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)|} \|\mathbf{x}_\star - \mathbf{x}_0\|_A \\
&= \frac{1}{|C_m\left(\frac{\kappa+1}{\kappa-1}\right)|} \|\mathbf{x}_\star - \mathbf{x}_0\|_A
\end{aligned}$$

Where $\kappa_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \frac{\lambda_{\max}}{\lambda_{\min}}$ is the condition number of A. Let $t = \frac{\kappa+1}{\kappa-1}$ then plugging this into the formula for $C_m(t)$ gives:

$$C_m\left(\frac{\kappa+1}{\kappa-1}\right) = \frac{1}{2}\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^m + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m\right] \geq \frac{1}{2}\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^m$$

Thus we have the final bound for CG:

$$\boxed{\|\mathbf{x}_\star - \mathbf{x}_m\|_A^2 \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \|\mathbf{x}_\star - \mathbf{x}_0\|_A^2}$$

1.10.2 Practical remarks

Lets generate a random SPD matrix $A \in \mathbb{R}^{n\times n}$, and do CG on $A^{N_{\text{pot}}}$ for some $N_{\text{pot}} \in (0, 1)$.

1.10.3 Convergence of CG and GMRES (Saad 6.11)

$A \in \mathbb{R}^{n\times n}$, $\mathbf{b}, \mathbf{x}_0 \in \mathbb{R}^n$, and $A \underbrace{\mathbf{x}_\star}_{\text{Exact solution}} = \mathbf{b}$ where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ with the Krylov subspace:

$$\mathcal{K}_k(A, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^{k-1}\mathbf{r}_0\}.$$

1.11 Lecture 13: 23.09.2025

1.11.1 Convergence properties of GMRES (Generalized Minimal Residual Method)

- $\mathbf{x}_\star$ exact solution of $A\mathbf{x} = \mathbf{b}$.
- $\mathbf{x}_m$ numerical solution after m iterations with some *krylov-space method*.

$$\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$$

$$\begin{aligned}
\mathbf{x}_\star - \mathbf{x}_m &= p_m(A)(\mathbf{x}_\star - \mathbf{x}_0), \quad p_m \in \mathcal{P}_m,\ p_m(0) = 1 \\
\mathbf{b} - A\mathbf{x}_m &= p_m(A)(\mathbf{b} - A\mathbf{x}_0) \\
\mathbf{r}_m &= p_m(A)\mathbf{r}_0
\end{aligned}$$

CG (Conjugate Gradient Method)

A is *SPD*, with $\mathcal{L}_m = \mathcal{K}_m(A, \mathbf{r}_0)$.

$$\|\mathbf{x}_\star - \mathbf{x}_m\|_A = \min_{\mathbf{x}\in\mathbf{x}_0+\mathcal{K}_m} \|\mathbf{x}_\star - \mathbf{x}\|_A$$

Used that A is diagonalizable, with orthogonal eigenvectors:

$$\begin{aligned}
A &= V\Lambda V^T, \quad V^TV = I, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n) \\
p(A) &= Vp(\Lambda)V^T \\
\|\mathbf{x}_\star - \mathbf{x}_m\|_A &= \sum_{i=1}^n \lambda_i p_m^2(\lambda_i)\lambda_i\xi_i^2, \quad \xi = V^T(\mathbf{x}_\star - \mathbf{x}_0) \\
&\le \max_i p_m^2(\lambda_i)\sum_{i=1}^n \lambda_i\xi_i^2 = \max_i p_m^2(\lambda_i)\|\mathbf{x}_\star - \mathbf{x}_0\|_A^2
\end{aligned}$$

We solve the min-max problem:

$$\min_{\substack{p\in\mathcal{P}_m \\ p(0)=1}} \max_{1\le i\le n} |p(\lambda_i)|$$

Using Chebyshev polynomials, we get the bound $[-1, 1] \to [\lambda_{\min}, \lambda_{\max}]$ with scale $p(0) = 1$.

1.11.2 GMRES

$\mathcal{L}_m = A\mathcal{K}_m$.

$$\|\mathbf{r}_m\|_2 = \min_{\mathbf{x}\in\mathbf{x}_0+\mathcal{K}_m} \|\mathbf{b} - A\mathbf{x}\|_2$$

$$\|\mathbf{r}_m\|_2 \le \|\mathbf{r}_{m-1}\|_2 \le \dots \le \|\mathbf{r}_0\|_2$$

For each $\|\mathbf{r}_0\|_2$ it is possible to find an A s.t.

$$\|\mathbf{r}_m\|_2 = \|\mathbf{r}_{m-1}\|_2 = \dots = \|\mathbf{r}_0\|_2$$

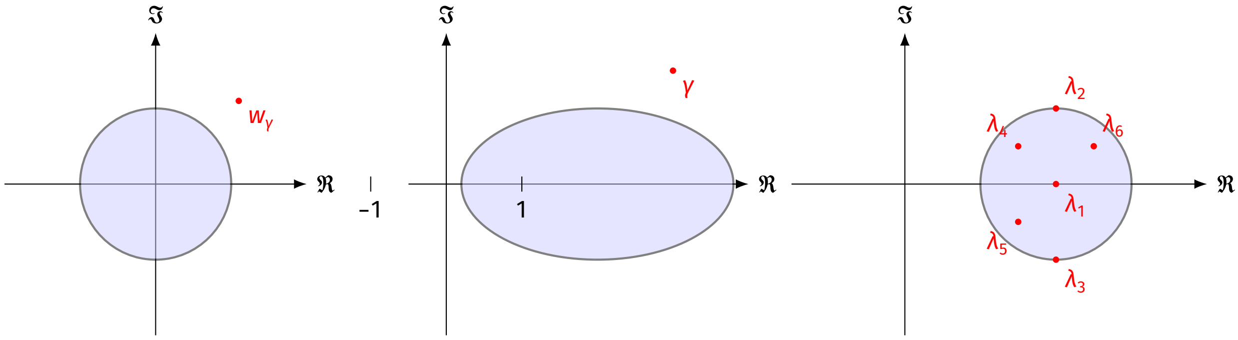
A may not be diagonalizable.

Now assume A is diagonalizable:

$$A = X\Lambda X^{-1}, \qquad \Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n), \quad \text{(eigenvalues)} \qquad X = [\mathbf{x}_1, \dots, \mathbf{x}_n] \quad \text{(eigenvectors)}$$

but X is not orthogonal anymore.

$$\begin{aligned}
p(A) &= Xp(\Lambda)X^{-1} \\
\mathbf{r}_m &= p_m(A)\mathbf{r}_0 = Xp_m(\Lambda)X^{-1}\mathbf{r}_0 \\
\|\mathbf{r}_m\|_2 &\leq \|X\|_2\|X^{-1}\|_2 \max_{1\leq i\leq n} |p_m(\lambda_i)| \, \|\mathbf{r}_0\|_2 \\
&= \sqrt{\lambda_{\max}(A^HA)\cdot\lambda_{\min}((A^HA)^{-1})} \max_{1\leq i\leq n} |p_m(\lambda_i)| \, \|\mathbf{r}_0\|_2 \\
&= \kappa_2(X) \max_{1\leq i\leq n} |p_m(\lambda_i)| \, \|\mathbf{r}_0\|_2 \\
\kappa_2(X) &= \|X\|_2\|X^{-1}\|_2 = \sqrt{\lambda_{\max}(A^HA)\cdot\lambda_{\min}((A^HA)^{-1})} = \frac{\sigma_{\max}(X)}{\sigma_{\min}(X)}
\end{aligned}$$



Let $\lambda_i \in \mathrm{E}$ for $i = 1, \dots, n$, where E is a closed ellipse, and $D_\rho := \{w \in \mathbb{C} : |w| = \rho\}$. We search for some $p^\star$ solving the min-max problem:

$$\min_{\substack{p\in\mathcal{P}_m \\ p(0)=1}} \max_{\lambda_i\in\mathrm{E}} |p(\lambda)|$$

Chebyshev polynomials in $\mathbb{C}$

let $z \in \mathbb{C}$:

$$\begin{aligned}
C_m(z) &= \cosh(m\cdot\rho), \quad \rho = \cosh^{-1}(z) \\
w &= e^\rho \\
C_m(z) &= \frac{1}{2}(e^{m\rho} + e^{-m\rho}) = \frac{1}{2}(w^m + w^{-m}) \\
C_{m+1}(z) &= 2zC_m(z) - C_{m-1}(z), \quad C_0(z) = 1, \; C_1(z) = z \\
z &= \frac{1}{2}(w + w^{-1})
\end{aligned}$$

Lemma 2: Zarantonello Let $\gamma \in \mathbb{C}$, $|\gamma| > \rho$, then:

$$\min_{\substack{p \in \mathcal{P}_m \\ p(\gamma)=1}} \max_{w \in D_\rho} = \left(\frac{\rho}{|\gamma|}\right)^m$$

Minimal polynomial is given by:

$$p(z) = \left(\frac{z}{\gamma}\right)^m$$

Max is obtained when $z = \rho$.

Joukowsky mapping

$$J(w) = \frac{1}{2}(w + w^{-1}), \quad w \in \mathbb{C},\ w \neq 0$$
$$J(D_\rho) = \mathrm{E}(0, 1, \frac{1}{2}(\rho + \rho^{-1}))$$

Theorem 1.13: Elman

Let $J(D_\rho) = \mathrm{E}_\rho$ and choose γ outside E_ρ, and let $w_\gamma = J^{-1}(\gamma)$ (the biggest), then:

$$\frac{\rho^m}{|w_\gamma|^m} \leq \min_{\substack{p \in \mathcal{P}_m \\ p(\gamma)=1}} \max_{z \in \mathrm{E}_\rho} |p(z)| \leq \frac{\rho^m + \rho^{-m}}{|w_\gamma^m + w_\gamma^{-m}|}$$

Then the optimal polynomial $p^\star$ is given by:

$$p^\star(w) = \frac{w^m + w^{-m}}{w_\gamma^m + w_\gamma^{-m}}, \quad w \in \mathbb{C}$$

is close to our optimal polynomial when m is large.

$$C_m(z) = \frac{1}{2}(w^m + w^{-m}), \quad z = \frac{1}{2}(w + w^{-1})$$
$$p^\star(z) = \frac{C_m(w)}{C_m(w_\gamma)}$$
$$\hat{C}_m(z) = \frac{C_m(\frac{z-c}{d})}{C_m(-\frac{c}{d})}, \begin{cases} \mathrm{E}(c, d, a), \\ \hat{C}_m(0) = 1 \end{cases}$$
$$\max_{z \in \mathrm{E}(c,d,a)} |\hat{C}_m(z)| = \frac{C_m(\frac{a}{d})}{|C_m(-\frac{c}{d})|}$$

$$\mathbf{r}_m \leq \kappa_2(X)\varepsilon^m \|\mathbf{r}_0\|_2 = \kappa_2(X)\frac{C_m(\frac{a}{d})}{|C_m(-\frac{c}{d})|}\|\mathbf{r}_0\|_2$$

$$C_m(z) = \frac{1}{2}\left[\left(z + \sqrt{z^2 - 1}\right)^m + \left(z - \sqrt{z^2 - 1}\right)^m\right]$$
$$\varepsilon^m = \frac{C_m(\frac{a}{d})}{|C_m(-\frac{c}{d})|} \approx \left(\frac{a + \sqrt{a^2 - d^2}}{c + \sqrt{c^2 - d^2}}\right)^m$$

The ellipse enclosing the eigenvalues can not include 0, because then $p(0) = 1$ can not be satisfied. If $a < c$, then we have convergene for sure.

1.12 Lecture 14: 24.09.2025

1.12.1 Convergence

Let $A = X\Lambda X^{-1}$, $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ and $\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\lambda_{\max}}{\lambda_{\min}}$ if A is *SPD*.

- **CG:**

$$\|\mathbf{x}_\star - \mathbf{x}_m\|_A \leq 2\left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right)^m \|\mathbf{x}_\star - \mathbf{x}_0\|_A$$

- **GMRES:** $\lambda(A) \subset \mathrm{E}(c, d, a)$: The set of eigenvalues is enclosed in an ellipse with center c, focal distance d and semi-major axis a. Then:

$$\|\mathbf{r}_m\|_2 \leq \|X\|_2 \|X^{-1}\|_2 \min_{\substack{p \in \mathbb{P}_m \\ p(0)=1}} \max_{z \in \mathrm{E}(c,d,a)} |p(z)| \, \|\mathbf{r}_0\|_2.$$

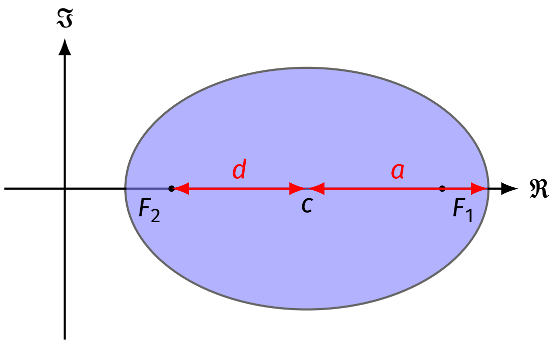
  For an ellipse one can use Chebyshev-type estimates to get an explicit geometric rate. Defining

$$q := \frac{a - \sqrt{a^2 - d^2}}{a + \sqrt{a^2 - d^2}} \quad (0 < q < 1),$$

  one convenient bound is

$$\|\mathbf{r}_m\|_2 \leq 2\, \|X\|_2 \|X^{-1}\|_2\, q^m\, \|\mathbf{r}_0\|_2,$$

  where the factor 2 depends on the normalization of the minimax polynomial and can be omitted in some formulations.



1.12.2 Preconditioning (Saad, Chap. 9)

$$A\mathbf{x} = \mathbf{b}$$

Rewrite the system by choosing $M \in \mathbb{R}^{n \times n}$.

- **Left preconditioning (LPC):** $M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$, solve for $\mathbf{x}$.
- **Right preconditioning (RPC):** $AM^{-1}\mathbf{u} = \mathbf{b}$, solve for $\mathbf{u} = M\mathbf{x}$ or $\mathbf{x} = M^{-1}\mathbf{u}$.

$$\tilde{A} = AM^{-1}$$

Apply **RPC** for A, $\mathbf{b}$ and $\mathbf{x}_0$ where:

$$\mathbf{u}_0 = M\mathbf{x}_0, \quad \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = \mathbf{b} - AM^{-1}\mathbf{u}_0 = \mathbf{b} - AM^{-1}M\mathbf{x}_0$$

Start with the *Arnoldi process* with right preconditioning:

Algorithm 8 Arnoldi process with RPC

```
β = ‖r_0‖_2, v_1 = r_0/β
w_j = A M^{-1} v_j
for i = 1, 2, …, j do
    h_ij = ⟨w_j, v_i⟩
    w_j = w_j − h_ij v_i
h_{j+1,j} = ‖w_j‖_2
if h_{j+1,j} = 0 then
    Stop
v_{j+1} = w_j / h_{j+1,j} return H̄_m, V_m
```

Now we solve for:

$$\begin{aligned} \overline{H}_m\mathbf{y}_m &= \beta\mathbf{e}_1 \\ \mathbf{u}_m &= \mathbf{u}_0 + V_m\mathbf{y}_m \\ \mathbf{x}_m &= M^{-1}\mathbf{u}_0 + M^{-1}V_m\mathbf{y}_m = \mathbf{x}_0 + M^{-1}V_m\mathbf{y}_m \end{aligned}$$

So $\mathbf{u}_m$ is never computed explicitly, we only need to compute:

$$M^{-1}\mathbf{v}_j = \mathbf{z}_j \quad \Rightarrow \quad M\mathbf{z}_j = \mathbf{v}_j$$

This has to be solved for each iteration (and store $\mathbf{z}_j$ instead of $\mathbf{v}_j$).

For the **LPC** we have:

$$\mathbf{r}_j = M^{-1}(\mathbf{b} - A\mathbf{x}_j)$$

1.12.3 Conjugate Gradient

A is *SPD* and $\tilde{A} = M^{-1}A$ is *SPD*, choose $M = LL^T$ SPD (Cholesky factorization).

$$\begin{aligned} M &= LL^T \\ M^{-1} &= L^{-T}L^{-1} \\ M^{-1}A\mathbf{x} &= M^{-1}\mathbf{b} \\ (L^{-T}L^{-1})AI\mathbf{x} &= L^{-T}L^{-1}\mathbf{b} \\ (L^{-T}L^{-1})A(L^{-T}L^{T})\mathbf{x} &= L^{-T}L^{-1}\mathbf{b} \\ L^{-T}(L^{-1}AL^{-T})(L^{T}\mathbf{x}) &= L^{-T}(L^{-1}\mathbf{b}) \\ (L^{-1}AL^{-T})(L^{T}\mathbf{x}) &= L^{-1}\mathbf{b} \\ \tilde{A}\tilde{\mathbf{x}} &= \tilde{\mathbf{b}} \quad \text{with } \tilde{A} = L^{-1}AL^{-T}, \tilde{\mathbf{x}} = L^T\mathbf{x}, \tilde{\mathbf{b}} = L^{-1}\mathbf{b} \end{aligned}$$

Algorithm 9 Preconditioned Conjugate Gradient (PCG) on $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$

Choose initial guess $\tilde{\mathbf{x}}_0$ (e.g. $\tilde{\mathbf{x}}_0 = L^T\mathbf{x}_0$)
$\tilde{\mathbf{r}}_0 = \tilde{\mathbf{b}} - \tilde{A}\tilde{\mathbf{x}}_0$
$\tilde{\mathbf{p}}_0 = \tilde{\mathbf{r}}_0$
for $j = 0, 1, 2, \ldots$ **do**
  $\alpha_j = \dfrac{\langle \tilde{\mathbf{r}}_j, \tilde{\mathbf{r}}_j \rangle}{\langle \tilde{A}\tilde{\mathbf{p}}_j, \tilde{\mathbf{p}}_j \rangle}$
  $\tilde{\mathbf{x}}_{j+1} = \tilde{\mathbf{x}}_j + \alpha_j \tilde{\mathbf{p}}_j$
  $\tilde{\mathbf{r}}_{j+1} = \tilde{\mathbf{r}}_j - \alpha_j \tilde{A}\tilde{\mathbf{p}}_j$
  if $\|\tilde{\mathbf{r}}_{j+1}\|_2 < \text{tol}$ **then**
    Stop
  $\beta_j = \dfrac{\langle \tilde{\mathbf{r}}_{j+1}, \tilde{\mathbf{r}}_{j+1} \rangle}{\langle \tilde{\mathbf{r}}_j, \tilde{\mathbf{r}}_j \rangle}$
  $\tilde{\mathbf{p}}_{j+1} = \tilde{\mathbf{r}}_{j+1} + \beta_j \tilde{\mathbf{p}}_j$
Return $\tilde{\mathbf{x}}_m$ and transform back $\mathbf{x}_m = L^{-T}\tilde{\mathbf{x}}_m$

We see that the inner products in α_j and β_j can be rewritten:

$$\begin{aligned}
\langle \tilde{\mathbf{r}}_j, \tilde{\mathbf{r}}_j \rangle &= \tilde{\mathbf{r}}_j^T \tilde{\mathbf{r}}_j \\
&= \langle L^{-1}\mathbf{r}_j, L^{-1}\mathbf{r}_j \rangle \\
&= \langle \mathbf{r}_j, L^{-T}L^{-1}\mathbf{r}_j \rangle \\
&= \langle \mathbf{r}_j, M^{-1}\mathbf{r}_j \rangle \\
&= \|\mathbf{r}_j\|_M^2 \\
\langle \tilde{A}\tilde{\mathbf{p}}_j, \tilde{\mathbf{p}}_j \rangle &= \langle L^{-1}AL^{-T}\tilde{\mathbf{p}}_j, \tilde{\mathbf{p}}_j \rangle \\
&= \langle L^{-1}A\mathbf{p}_j, \tilde{\mathbf{p}}_j \rangle \\
&= \langle A\mathbf{p}_j, L^{-T}\tilde{\mathbf{p}}_j \rangle \\
&= \langle A\mathbf{p}_j, \mathbf{p}_j \rangle
\end{aligned}$$

Then the iterations become:

$$\begin{aligned}
\mathbf{x}_{j+1} &= \mathbf{x}_j + \alpha_j L^{-T}L^T\mathbf{p}_j = \mathbf{x}_j + \alpha_j \mathbf{p}_j \\
\mathbf{r}_{j+1} &= \mathbf{r}_j - \alpha_j \overbrace{LL^{-1}AL^{-T}L^T}^{\tilde{A}} \mathbf{p}_j = \mathbf{r}_j - \alpha_j A\mathbf{p}_j \\
\mathbf{p}_{j+1} &= L^{-T}L^{-1}\mathbf{r}_{j+1} + \beta_j \mathbf{p}_j = M^{-1}\mathbf{r}_{j+1} + \beta_j \mathbf{p}_j
\end{aligned}$$

Then we have the **Preconditioned Conjugate Gradient (PCG) algorithm**:

Algorithm 10 Preconditioned Conjugate Gradient (PCG)

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\mathbf{z}_0 = M^{-1}\mathbf{r}_0$, $\mathbf{p}_0 = \mathbf{z}_0$
for $j = 0, 1, 2, \dots$ **do**
  $\alpha_j = \dfrac{\langle \mathbf{r}_j, \mathbf{z}_j \rangle}{\langle A\mathbf{p}_j, \mathbf{p}_j \rangle}$
  $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j$
  $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j A\mathbf{p}_j$
  if $\|\mathbf{r}_{j+1}\|_2 < \text{tol}$ **then**
    Stop
  $\mathbf{z}_{j+1} = M^{-1}\mathbf{r}_{j+1}$
  $\beta_j = \dfrac{\langle \mathbf{r}_{j+1}, \mathbf{z}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{z}_j \rangle}$
  $\mathbf{p}_{j+1} = \mathbf{z}_{j+1} + \beta_j \mathbf{p}_j$
return $\mathbf{x}_m$

The price we pay for preconditioning with M is that we have to solve a linear system $M\mathbf{z}_j = \mathbf{r}_j$ at each iteration, and store $\mathbf{z}_j$.

How to choose M?

- M should be *SPD* if A is *SPD* (when using PCG).
- M should be a good approximation of A (in some sense), i.e. $M \approx A$ so that $\kappa(\tilde{A}) < \kappa(A)$.
- M should be cheap to apply, i.e. solving $M\mathbf{z} = \mathbf{r}$ should be cheap.
- M should be sparse (if A is sparse).
- if A is *SPD*, then M should also be *SPD*.

In this course:

1. Use one iteration of one of the *stationary methods* (e.g. Jacobi, Gauss-Seidel, SOR).
   - Jacobi: $M = D$ (diagonal of A).
   - Gauss-Seidel: $M = D + L$ (lower triangular part of A).
   - SOR: $M = \frac{1}{\omega} D + L$.
2. Incomplete factorizations
   - Incomplete LU (ILU) for general $A \approx LU$. LU keeps the sparsity structure of A.
   - Incomplete Cholesky (IC) for *SPD* $A \approx LL^T$.
3. *Multigrid methods*

1.13 Lecture 15: 25/09/2025

1.13.1 The principles of preconditioning

Let $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{n\times n}$ has slow convergence.

We rewrite the system by choosing $M \in \mathbb{R}^{n\times n}$:

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b} \tag{LPC}$$

$$AM^{-1}\mathbf{y} = \mathbf{b}, \quad \mathbf{y} = M\mathbf{x} \text{ or } \mathbf{x} = M^{-1}\mathbf{y} \tag{RPC}$$

Apply (RPC) to GMRES:

Algorithm 11 Right-preconditioned GMRES

Require:
$A, \mathbf{b}, \mathbf{x}_0$
$\mathbf{u}_0 = M\mathbf{x}_0$
$\mathbf{r}_0 = \mathbf{b} - AM^{-1}\mathbf{u}_0 = \mathbf{b} - A\mathbf{x}_0$
$\beta = \|\mathbf{r}_0\|_2$
$\mathbf{v}_1 = \mathbf{r}_0/\beta$
for $j = 1, 2, \ldots$ until convergence **do**
  $\mathbf{w}_j = AM^{-1}\mathbf{v}_j$
  for $i = 1, \ldots, j$ **do**
    $h_{ij} = \mathbf{w}_j^T\mathbf{v}_i$
    $\mathbf{w}_j = \mathbf{w}_j - h_{ij}\mathbf{v}_i$
  $h_{j+1,j} = \|\mathbf{w}_j\|_2$
  $\mathbf{v}_{j+1} = \mathbf{w}_j/h_{j+1,j}$
  Compute $\mathbf{y}_j$ that minimizes $\|H_j\mathbf{y} - \beta e_1\|_2$
  $\mathbf{x}_j = M^{-1}(\mathbf{u}_0 + V_j\mathbf{y}_j)$

Solve $\bar{H}_m\mathbf{y} = \beta\mathbf{e}_1$ in least squares sense.

$$\begin{aligned}\mathbf{u}_m &= \mathbf{u}_0 + V_m\mathbf{y}_m \\ \mathbf{x}_m &= M^{-1}\mathbf{u}_0 + M^{-1}V_m\mathbf{y}_m = \mathbf{x}_0 + M^{-1}V_m\mathbf{y}_m\end{aligned}$$

We never use $\mathbf{u}_m$ explicitly. We need to compute:

$$\begin{aligned}\mathbf{z}_j &= M^{-1}\mathbf{v}_j \\ \mathbf{v}_j &= A\mathbf{z}_j\end{aligned}$$

for each iteration j.

The residual is the same as unconditioned GMRES:

$$\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m = \mathbf{b} - AM^{-1}\mathbf{u}_m = \mathbf{b} - AM^{-1}(\mathbf{u}_0 + V_m\mathbf{y}_m) = \mathbf{r}_0 - AM^{-1}V_m\mathbf{y}_m = \mathbf{r}_0 - W_m\mathbf{y}_m$$

Using (LPC) changes the residual.

We want $M^{-1} \approx A^{-1}$ s.t. $M^{-1}A \approx I_n$.

1.13.2 Preconditioning the CG method

A is SPD and $\tilde{A} = AM^{-1}$ also must be SPD, then M is SPD, with $M = LL^T$.

$$\begin{aligned}\tilde{A} &= AM^{-1}, \text{ or } \tilde{A} &= M^{-1}A \\ M^{-1}A\mathbf{x} &= M^{-1}\mathbf{b} \\ L^{-T}\underbrace{L^{-1}AL^{-T}}_{\tilde{A}}\underbrace{L^T\mathbf{x}}_{\tilde{\mathbf{x}}} &= L^{-T}\underbrace{L^{-1}\mathbf{b}}_{\tilde{\mathbf{b}}}\end{aligned}$$

Now $\tilde{A}$ is SPD:

$$\tilde{A} = L^{-1}AL^{-T} = (L^{-1}AL^{-T})^T = L^{-1}A^TL^{-T} = L^{-1}AL^{-T}, \quad \tilde{\mathbf{x}} = L^T\mathbf{x}, \quad \tilde{\mathbf{b}} = L^{-1}\mathbf{b}$$

with residual

$$\tilde{\mathbf{r}} = \tilde{\mathbf{b}} - \tilde{A}\tilde{\mathbf{x}} = L^{-1}(\mathbf{b} - A\mathbf{x})$$

CG on $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$:

Algorithm 12 Preconditioned CG

Require:
$A, \mathbf{b}, \mathbf{x}_0$
$\tilde{\mathbf{r}}_0 = L^{-1}(\mathbf{b} - A\mathbf{x}_0)$
$\tilde{\mathbf{p}}_0 = \tilde{\mathbf{r}}_0$
for $j = 0, 1, 2, \ldots$ until convergence **do**
  $\alpha_j = \frac{\tilde{\mathbf{r}}_j^T \tilde{\mathbf{r}}_j}{\tilde{\mathbf{p}}_j^T \tilde{A} \tilde{\mathbf{p}}_j} = \frac{\|\tilde{\mathbf{r}}_j\|_2^2}{\|\tilde{p}_j\|_{\tilde{A}}^2}$
  $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \tilde{\mathbf{p}}_j$
  $\tilde{\mathbf{r}}_{j+1} = \tilde{\mathbf{r}}_j - \alpha_j \tilde{A} \tilde{\mathbf{p}}_j$
  $\beta_j = \frac{\tilde{\mathbf{r}}_{j+1}^T \tilde{\mathbf{r}}_{j+1}}{\tilde{\mathbf{r}}_j^T \tilde{\mathbf{r}}_j} = \frac{\|\tilde{\mathbf{r}}_{j+1}\|_2^2}{\|\tilde{\mathbf{r}}_j\|_2^2}$
  $\tilde{\mathbf{p}}_{j+1} = \tilde{\mathbf{r}}_{j+1} + \beta_j \tilde{\mathbf{p}}_j$

For α_j we have:

$$\langle \tilde{\mathbf{r}}_j, \tilde{\mathbf{r}}_j \rangle = \mathbf{r}_j^T \mathbf{r}_j = \langle L^{-1}\mathbf{r}_j, L^{-1}\mathbf{r}_j \rangle = \langle \mathbf{r}_j, L^{-T}L^{-1}\mathbf{r}_j \rangle = \langle \mathbf{r}_j, M^{-1}\mathbf{r}_j \rangle = \|\mathbf{r}_j\|_{M^{-1}}^2$$
$$\langle \tilde{A}\tilde{\mathbf{p}}_j, \tilde{\mathbf{p}}_j \rangle = \langle L^{-1}AL^{-T}\tilde{\mathbf{p}}_j, \tilde{\mathbf{p}}_j \rangle = \langle A \underbrace{L^{-T}\tilde{\mathbf{p}}_j}_{\mathbf{p}_j}, L^{-T}\tilde{\mathbf{p}}_j \rangle = \langle A\mathbf{p}_j, \mathbf{p}_j \rangle = \|\mathbf{p}_j\|_A^2$$

We multiply $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{p}}_j$ with L^{-T}, and $\tilde{\mathbf{r}}_j$ with L to get:

$$L^{-T}\tilde{\mathbf{x}}_{j+1} = L^{-T}\tilde{\mathbf{x}}_j + \alpha_j L^{-T}\tilde{\mathbf{p}}_j = \mathbf{x}_j + \alpha_j \mathbf{p}_j$$
$$L\tilde{\mathbf{r}}_{j+1} = L\tilde{\mathbf{r}}_j - \alpha_j L\tilde{A}\tilde{\mathbf{p}}_j = \mathbf{r}_j - \alpha_j A\mathbf{p}_j$$
$$L^{-T}\tilde{\mathbf{p}}_{j+1} = L^{-T}\tilde{\mathbf{r}}_{j+1} + \beta_j L^{-T}\tilde{\mathbf{p}}_j = M^{-1}\mathbf{r}_{j+1} + \beta_j \mathbf{p}_j$$

We have a new $\mathbf{p}_j$ and a new α_j:

Algorithm 13 Preconditioned CG

Require:
$A, \mathbf{b}, \mathbf{x}_0$
$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
Solve $M\mathbf{z}_0 = \mathbf{r}_0$
$\mathbf{p}_0 = \mathbf{z}_0$
for $j = 0, 1, 2, \ldots$ until convergence **do**
  $\alpha_j = \frac{\mathbf{r}_j^T \mathbf{z}_j}{\mathbf{p}_j^T A \mathbf{p}_j} = \frac{\langle \mathbf{r}_j, \mathbf{z}_j \rangle}{\|\mathbf{p}_j\|_A^2}$
  $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j$
  $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j A \mathbf{p}_j$
  $\mathbf{z}_{j+1} = M^{-1}\mathbf{r}_{j+1}$ (solve $M\mathbf{z}_{j+1} = \mathbf{r}_{j+1}$)
  $\beta_j = \frac{\mathbf{r}_{j+1}^T \mathbf{z}_{j+1}}{\mathbf{r}_j^T \mathbf{z}_j} = \frac{\langle \mathbf{r}_{j+1}, \mathbf{z}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{z}_j \rangle}$
  $\mathbf{p}_{j+1} = \mathbf{z}_{j+1} + \beta_j \mathbf{p}_j$

Price: solve $M\mathbf{z}_j = \mathbf{r}_j$ for each iteration j, only store $\mathbf{z}_j$.

1.13.3 Choosing a preconditioner

We want $M \approx A$ s.t. $AM^{-1} \approx I_n$.

Chapter 2

Exercises

2.1 Exercise 1

2.1.1 Problem 2

$$\|A\|_{pq} = \max_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_q}$$

Let $A = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$.

(a) Computing $\|A\|_{1\infty,\mathbb{R}}$

We need to find the maximum of $\frac{\|A\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty}$ over all real vectors $\mathbf{x} \neq 0$.

Let $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$ be a real vector. Then:

$$\begin{aligned} A\mathbf{x} &= \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} x_1 - x_2 \\ x_1 + x_2 \end{bmatrix} \end{aligned}$$

The 1-norm of $A\mathbf{x}$ is:

$$\|A\mathbf{x}\|_1 = |x_1 - x_2| + |x_1 + x_2|$$

The ∞-norm of $\mathbf{x}$ is:

$$\|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|)$$

To analyze $|x_1 - x_2| + |x_1 + x_2|$, we consider different cases based on the signs of $x_1 \pm x_2$:

Case 1: $x_1 + x_2 \geq 0$ and $x_1 - x_2 \geq 0$ (i.e., $x_1 \geq |x_2|$)

$$|x_1 - x_2| + |x_1 + x_2| = (x_1 - x_2) + (x_1 + x_2) = 2x_1 = 2|x_1|$$

Since $x_1 \geq |x_2|$, we have $\|\mathbf{x}\|_\infty = |x_1|$, so:

$$\frac{\|A\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty} = \frac{2|x_1|}{|x_1|} = 2$$

Case 2: $x_1 + x_2 \geq 0$ and $x_1 - x_2 \leq 0$ (i.e., $x_2 \geq x_1 \geq -x_2$)

$$|x_1 - x_2| + |x_1 + x_2| = -(x_1 - x_2) + (x_1 + x_2) = 2x_2 = 2|x_2|$$

Since $x_2 \geq |x_1|$, we have $\|\mathbf{x}\|_\infty = |x_2|$, so:

$$\frac{\|A\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty} = \frac{2|x_2|}{|x_2|} = 2$$

Case 3: $x_1 + x_2 \leq 0$ and $x_1 - x_2 \geq 0$ (i.e., $-x_2 \geq x_1 \geq x_2$)

$$|x_1 - x_2| + |x_1 + x_2| = (x_1 - x_2) - (x_1 + x_2) = -2x_2 = 2|x_2|$$

Since $|x_2| \geq |x_1|$, we have $\|\mathbf{x}\|_\infty = |x_2|$, so:

$$\frac{\|A\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty} = \frac{2|x_2|}{|x_2|} = 2$$

Case 4: $x_1 + x_2 \leq 0$ and $x_1 - x_2 \leq 0$ (i.e., $x_1 \leq -|x_2|$)

$$|x_1 - x_2| + |x_1 + x_2| = -(x_1 - x_2) - (x_1 + x_2) = -2x_1 = 2|x_1|$$

Since $|x_1| \geq |x_2|$, we have $\|\mathbf{x}\|_\infty = |x_1|$, so:

$$\frac{\|A\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty} = \frac{2|x_1|}{|x_1|} = 2$$

In all cases, we get $\frac{\|A\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty} = 2$. Therefore:

$$\|A\|_{1\infty,\mathbb{R}} = 2$$

(b) Computing $\|A\|_{1\infty}$ over complex vectors

Now we consider complex vectors. Let $\mathbf{x} = \begin{bmatrix} 1+i & 1-i \end{bmatrix}^\mathsf{T}$.

First, compute $A\mathbf{x}$:

$$\begin{aligned} A\mathbf{x} &= \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1+i \\ 1-i \end{bmatrix} \\ &= \begin{bmatrix} (1+i) - (1-i) \\ (1+i) + (1-i) \end{bmatrix} \\ &= \begin{bmatrix} 1+i-1+i \\ 1+i+1-i \end{bmatrix} \\ &= \begin{bmatrix} 2i \\ 2 \end{bmatrix} \end{aligned}$$

Compute the norms:

$$
\begin{aligned}
\|A\mathbf{x}\|_1 &= |2i| + |2| = 4 \\
\|\mathbf{x}\|_\infty &= \max(|1+i|, |1-i|) \\
&= \max(\sqrt{1^2+1^2}, \sqrt{1^2+(-1)^2}) \\
&= \max(\sqrt{2}, \sqrt{2}) \\
&= \sqrt{2} \\
\frac{\|A\mathbf{x}\|_1}{\|\mathbf{x}\|_\infty} &= \frac{4}{\sqrt{2}} \\
&= 2\sqrt{2}
\end{aligned}
$$

Thus:

$$\|A\|_{1\infty,\mathbb{R}} = 2 < \|A\|_{1\infty} \geq 2\sqrt{2}$$

This shows that allowing complex vectors can increase the norm value.

2.1.2 Problem 3

(a) 2-norm of rank-1 matrix $E = uv^H$

Let $E = uv^H$ where $\mathbf{u}, \mathbf{v} \in \mathbb{C}^n$. We compute $\|E\|_2 = \sqrt{\rho(E^H E)}$.

First, find E^H:

$$E^H = (uv^H)^H = (\mathbf{v}^H)^H \mathbf{u}^H = \mathbf{v}\mathbf{u}^H$$

Next, compute $E^H E$:

$$
\begin{aligned}
E^H E &= (\mathbf{v}\mathbf{u}^H)(\mathbf{u}\mathbf{v}^H) \\
&= \mathbf{v}(\mathbf{u}^H\mathbf{u})\mathbf{v}^H \quad \text{(associativity)} \\
&= \mathbf{v}(\|\mathbf{u}\|_2^2)\mathbf{v}^H \quad (\text{since } \mathbf{u}^H\mathbf{u} = \|\mathbf{u}\|_2^2) \\
&= \|\mathbf{u}\|_2^2(\mathbf{v}\mathbf{v}^H)
\end{aligned}
$$

Now we find the spectral radius. Note that vv^H is a rank-1 matrix with:

- Eigenvalue $\mathbf{v}^H\mathbf{v} = \|\mathbf{v}\|_2^2$, with $m_g(\|\mathbf{v}\|_2^2) = 1$
- Eigenvalue 0 with multiplicity $m_g(0) = n - 1$

Therefore:

$$
\begin{aligned}
\rho(E^H E) &= \rho(\|\mathbf{u}\|_2^2 \cdot \mathbf{v}\mathbf{v}^H) \\
&= \|\mathbf{u}\|_2^2 \cdot \rho(\mathbf{v}\mathbf{v}^H) \\
&= \|\mathbf{u}\|_2^2 \cdot \|\mathbf{v}\|_2^2
\end{aligned}
$$

Thus:

$$\|E\|_2 = \sqrt{\rho(E^H E)} = \sqrt{\|\mathbf{u}\|_2^2 \|\mathbf{v}\|_2^2} = \|\mathbf{u}\|_2 \|\mathbf{v}\|_2$$

(b) Frobenius norm of rank-1 matrix

The Frobenius norm is defined as $\|A\|_F = \sqrt{\text{trace}(A^H A)}$.

Using our previous calculation of $E^H E$:

$$\begin{aligned}\|E\|_F^2 &= \text{trace}(E^H E) \\ &= \text{trace}(\|\mathbf{u}\|_2^2 \cdot \mathbf{v}\mathbf{v}^H) \\ &= \|\mathbf{u}\|_2^2 \,\text{trace}(\mathbf{v}\mathbf{v}^H)\end{aligned}$$

Now, $\mathbf{v}\mathbf{v}^H$ is an $n \times n$ matrix where $(\mathbf{v}\mathbf{v}^H)_{ij} = v_i \overline{v_j}$. The diagonal entries are:

$$(\mathbf{v}\mathbf{v}^H)_{ii} = v_i \overline{v_i} = |v_i|^2$$

Therefore:

$$\text{trace}(\mathbf{v}\mathbf{v}^H) = \sum_{i=1}^{n} |v_i|^2 = \|\mathbf{v}\|_2^2$$

Substituting back:

$$\|E\|_F^2 = \|\mathbf{u}\|_2^2 \|\mathbf{v}\|_2^2$$

Thus:

$$\|E\|_F = \|\mathbf{u}\|_2 \|\mathbf{v}\|_2$$

The result holds for both the 2-norm and Frobenius norm.

2.1.3 Problem 4

(a) Eigenvalues and Jordan form of $A = uv^H$: Let $A = uv^H$ where $\mathbf{u}, \mathbf{v} \in \mathbb{C}^n$. To find eigenvalues, we solve $A\mathbf{x} = \lambda\mathbf{x}$.

$$\begin{aligned}uv^H\mathbf{x} &= \lambda\mathbf{x} \\ \mathbf{u}(\mathbf{v}^H\mathbf{x}) &= \lambda\mathbf{x}\end{aligned}$$

Since $\mathbf{v}^H\mathbf{x}$ is a scalar, let $\alpha = \mathbf{v}^H\mathbf{x}$. Then:

$$\alpha\mathbf{u} = \lambda\mathbf{x}$$

1. $\mathbf{x}$ is parallel to $\mathbf{u}$, i.e., $\mathbf{x} = \beta\mathbf{u}$ for some $\beta \neq 0$.

$$\begin{aligned}\mathbf{u}(\mathbf{v}^H(\beta\mathbf{u})) &= \lambda(\beta\mathbf{u}) \\ \beta\mathbf{u}(\mathbf{v}^H\mathbf{u}) &= \lambda\beta\mathbf{u} \\ \beta(\mathbf{v}^H\mathbf{u})\mathbf{u} &= \lambda\beta\mathbf{u}\end{aligned}$$

For $\beta \neq 0$, dividing by $\beta\mathbf{u}$ gives:

$$\lambda = \mathbf{v}^H\mathbf{u}$$

So $\lambda_1 = \mathbf{v}^H\mathbf{u}$ is an eigenvalue with eigenvector in the direction of $\mathbf{u}$.

2. **x** is orthogonal to **u**.
   If $\mathbf{x} \perp \mathbf{u}$, then from $\alpha\mathbf{u} = \lambda\mathbf{x}$ and $\alpha = \mathbf{v}^H\mathbf{x}$, we need $\alpha = 0$ (since **u** and **x** are orthogonal and $\mathbf{u} \neq 0$). This means $\mathbf{v}^H\mathbf{x} = 0$, so **x** is orthogonal to **v**. From $\alpha\mathbf{u} = \lambda\mathbf{x}$ with $\alpha = 0$, we get $\lambda\mathbf{x} = 0$, which implies $\lambda = 0$. The eigenspace for $\lambda = 0$ consists of all vectors orthogonal to **v**, which has dimension $n - 1$ (assuming $\mathbf{v} \neq 0$).

- $\lambda_1 = \mathbf{v}^H\mathbf{u}$ with geometric multiplicity 1
- $\lambda_2 = \cdots = \lambda_n = 0$ with geometric multiplicity $n - 1$

Jordan Normal Form: Since rank$(A) = 1$ and the geometric multiplicity of eigenvalue 0 is $n - 1$, we have:

$$\text{nullity}\,(A - 0 \cdot I) = \text{nullity}(A) = n - \text{rank}(A) = n - 1$$

This equals the algebraic multiplicity of eigenvalue 0, so all Jordan blocks for eigenvalue 0 have size 1. The Jordan normal form is:

$$J = \begin{bmatrix} \mathbf{v}^H\mathbf{u} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

(b) eigenpair relation between A and $A+\lambda I$: Let $A \in \mathbb{C}^{n\times n}$ and let μ be an eigenvalue of A with eigenvector $\mathbf{x} \neq 0$, so $A\mathbf{x} = \mu\mathbf{x}$.

Consider the matrix $B = A + \lambda I$. Then:

$$\begin{aligned} B\mathbf{x} &= (A + \lambda I)\,\mathbf{x} \\ &= A\mathbf{x} + \lambda I\mathbf{x} \\ &= A\mathbf{x} + \lambda\mathbf{x} \\ &= \mu\mathbf{x} + \lambda\mathbf{x} \\ &= (\mu + \lambda)\mathbf{x} \end{aligned}$$

Therefore, $\mu + \lambda$ is an eigenvalue of $B = A + \lambda I$ with the same eigenvector **x**.

Conversely, if ν is an eigenvalue of $B = A + \lambda I$ with eigenvector **y**, then:

$$\begin{aligned} (A + \lambda I)\,\mathbf{y} &= \nu\mathbf{y} \\ A\mathbf{y} + \lambda\mathbf{y} &= \nu\mathbf{y} \\ A\mathbf{y} &= (\nu - \lambda)\mathbf{y} \end{aligned}$$

So $\nu - \lambda$ is an eigenvalue of A, which means $\nu = (\nu - \lambda) + \lambda$ where $\nu - \lambda \in \sigma(A)$.

This establishes the bijection:

$$\sigma\,(A + \lambda I) = \{\mu + \lambda \,:\, \mu \in \sigma(A)\}$$

The Jordan structure remains unchanged because the eigenvector relationships are preserved.

(c) Eigenvalues and eigenbasis of a given matrix

$$\begin{aligned} A &= \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \\ &= \mathbf{e}\mathbf{e}^\top + I_4 \\ A &= J + I_4 \end{aligned}$$

From part (a), the eigenvalues of $J = \mathbf{e}\mathbf{e}^T$ are:

$$\begin{aligned} \lambda_1 &= \mathbf{e}^T\mathbf{e} = 1^2 + 1^2 + 1^2 + 1^2 = 4, & \mathbf{v}_1 &= \mathbf{e}, \\ \lambda_2 &= \lambda_3 = \lambda_4 = 0 & m_g(0) &= 3 \end{aligned}$$

From part (b), the eigenvalues of $A = J + I_4$ are:

$$\begin{aligned} \mu_1 &= 1 + 4 = 5, & \mathbf{v}_1 &= \mathbf{e}, \\ \mu_2 &= \mu_3 = \mu_4 = 1 + 0 = 1, & m_g(1) &= 3 \end{aligned}$$

To find the eigenspace for $\mu = 1$, we solve:

$$\begin{aligned} (A - 1 \cdot I)\mathbf{x} &= \mathbf{0} \\ (J + I - I)\mathbf{x} &= \mathbf{0} \\ J\mathbf{x} &= \mathbf{0} \end{aligned}$$

The null space of $J = \mathbf{e}\mathbf{e}^T$ consists of all vectors orthogonal to $\mathbf{e}$:

$$\begin{aligned} \mathbf{e}^T\mathbf{x} &= 0 \\ x_1 + x_2 + x_3 + x_4 &= 0 \end{aligned}$$

A basis for this 3-dimensional space is:

$$\mathbf{v}_2 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

We can verify these are eigenvectors:

$$A\mathbf{v}_2 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{v}_2$$

Therefore, the complete eigenbasis and eigenvalues of A are:

$$\begin{aligned} \text{eig}(A) = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\} &= \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} \right\} \\ \sigma(A) &= \{5, 1, 1, 1\} \end{aligned}$$

2.1.4 Problem 5

Given matrix:

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 0 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(a) Gram-Schmidt process: The columns of A are:

$$\mathbf{a}_1 = \begin{bmatrix}1\\1\\2\\1\end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix}2\\0\\2\\1\end{bmatrix}, \quad \mathbf{a}_3 = \begin{bmatrix}0\\1\\1\\1\end{bmatrix}$$

Step 1: Compute $\mathbf{q}_1$

$$\begin{aligned}\mathbf{u}_1 &= \mathbf{a}_1 = \begin{bmatrix}1\\1\\2\\1\end{bmatrix}\\ \|\mathbf{u}_1\|_2 &= \sqrt{1^2+1^2+2^2+1^2} = \sqrt{7}\\ \mathbf{q}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|_2} = \frac{1}{\sqrt{7}}\begin{bmatrix}1\\1\\2\\1\end{bmatrix}\end{aligned}$$

Step 2: Compute $\mathbf{q}_2$

$$\begin{aligned}\mathbf{q}_1^T\mathbf{a}_2 &= \frac{1}{\sqrt{7}}\begin{bmatrix}1 & 1 & 2 & 1\end{bmatrix}\begin{bmatrix}2\\0\\2\\1\end{bmatrix} = \frac{1}{\sqrt{7}}(2+0+4+1) = \sqrt{7}\\ \mathbf{u}_2 &= \mathbf{a}_2 - (\mathbf{q}_1^T\mathbf{a}_2)\mathbf{q}_1 = \begin{bmatrix}2\\0\\2\\1\end{bmatrix} - \sqrt{7}\cdot\frac{1}{\sqrt{7}}\begin{bmatrix}1\\1\\2\\1\end{bmatrix} = \begin{bmatrix}2\\0\\2\\1\end{bmatrix} - \begin{bmatrix}1\\1\\2\\1\end{bmatrix} = \begin{bmatrix}1\\-1\\0\\0\end{bmatrix}\\ \|\mathbf{u}_2\|_2 &= \sqrt{1^2+(-1)^2+0^2+0^2} = \sqrt{2}\\ \mathbf{q}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|_2} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\\0\\0\end{bmatrix}\end{aligned}$$

Step 3: Compute $\mathbf{q}_3$

$$\mathbf{q}_1^T\mathbf{a}_3 = \frac{1}{\sqrt{7}}\begin{bmatrix}1 & 1 & 2 & 1\end{bmatrix}\begin{bmatrix}0\\1\\1\\1\end{bmatrix} = \frac{1}{\sqrt{7}}(0+1+2+1) = \frac{4}{\sqrt{7}},$$

$$\mathbf{q}_2^T\mathbf{a}_3 = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}0\\1\\1\\1\end{bmatrix} = \frac{1}{\sqrt{2}}(0-1+0+0) = -\frac{1}{\sqrt{2}},$$

$$\mathbf{u}_3 = \mathbf{a}_3 - (\mathbf{q}_1^T\mathbf{a}_3)\mathbf{q}_1 - (\mathbf{q}_2^T\mathbf{a}_3)\mathbf{q}_2 = \begin{bmatrix}0\\1\\1\\1\end{bmatrix} - \frac{4}{\sqrt{7}}\cdot\frac{1}{\sqrt{7}}\begin{bmatrix}1\\1\\2\\1\end{bmatrix} - \left(-\frac{1}{\sqrt{2}}\right)\cdot\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\\0\\0\end{bmatrix} = \begin{bmatrix}-1/14\\-1/14\\-1/7\\3/7\end{bmatrix}$$

$$\|\mathbf{u}_3\|_2 = \sqrt{\left(-\frac{1}{14}\right)^2 + \left(-\frac{1}{14}\right)^2 + \left(-\frac{1}{7}\right)^2 + \left(\frac{3}{7}\right)^2} = \frac{\sqrt{42}}{14}$$

$$\mathbf{q}_3 = \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|_2} = \frac{14}{\sqrt{42}}\begin{bmatrix}-1/14\\-1/14\\-1/7\\3/7\end{bmatrix} = \frac{1}{\sqrt{42}}\begin{bmatrix}-1\\-1\\-2\\6\end{bmatrix}$$

Final QR factorization:

$$Q = \begin{bmatrix}1/\sqrt{7} & 1/\sqrt{2} & -1/\sqrt{42}\\ 1/\sqrt{7} & -1/\sqrt{2} & -1/\sqrt{42}\\ 2/\sqrt{7} & 0 & -2/\sqrt{42}\\ 1/\sqrt{7} & 0 & 6/\sqrt{42}\end{bmatrix}, \qquad R = \begin{bmatrix}\sqrt{7} & \sqrt{7} & 4/\sqrt{7}\\ 0 & \sqrt{2} & -1/\sqrt{2}\\ 0 & 0 & \sqrt{42}/14\end{bmatrix}$$

(b) Householder reflections

1. Eliminate first column below diagonal. Let $\mathbf{x} = \begin{bmatrix}1 & 1 & 2 & 1\end{bmatrix}^T$.
   We want to find the Householder matrix $H_1\mathbf{x} = \|\mathbf{x}\|_2\mathbf{e}_1$:

$$\mathbf{v} = \mathbf{x} - \|\mathbf{x}\|_2\mathbf{e}_1 = \begin{bmatrix}1\\1\\2\\1\end{bmatrix} - \sqrt{7}\begin{bmatrix}1\\0\\0\\0\end{bmatrix} = \begin{bmatrix}1-\sqrt{7}\\1\\2\\1\end{bmatrix}$$

$$\|\mathbf{v}\|_2^2 = (1-\sqrt{7})^2 + 1^2 + 2^2 + 1^2 = 1 - 2\sqrt{7} + 7 + 1 + 4 + 1 = 14 - 2\sqrt{7}$$

$$\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$$

$$H_1 = I - 2\mathbf{u}\mathbf{u}^T$$

$$H_1A = \begin{bmatrix}\sqrt{7} & * & *\\ 0 & * & *\\ 0 & * & *\\ 0 & * & *\end{bmatrix}$$

2. Apply Householder to eliminate second column below diagonal. Let **y** be the second column of:

$$H_1A = \begin{bmatrix} * & 2\sqrt{7} & * \\ 0 & -\sqrt{2} & * \\ 0 & \frac{1}{\sqrt{2}} & * \\ 0 & -\frac{1}{\sqrt{2}} & * \end{bmatrix}$$

We want to find the Householder matrix $H_2\mathbf{y} = \|\mathbf{y}\|_2\mathbf{e}_1$:

$$\mathbf{y} = \begin{bmatrix} 2\sqrt{7} \\ -\sqrt{2} \\ 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$$

$$\|\mathbf{y}\|_2 = 4\sqrt{2}$$

$$\mathbf{w} = \mathbf{y} - \|\mathbf{y}\|_2\mathbf{e}_1 = \begin{bmatrix} 2\sqrt{7} - 4\sqrt{2} \\ -\sqrt{2} \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\|\mathbf{w}\|_2^2 = 35 - 16\sqrt{14}$$

$$\mathbf{u}_2 = \frac{\mathbf{w}}{\|\mathbf{w}\|_2}$$

$$H_2 = I - 2\mathbf{u}_2\mathbf{u}_2^T$$

$$H_2H_1A = \begin{bmatrix} \sqrt{7} & * & * \\ 0 & \sqrt{2} & * \\ 0 & 0 & * \\ 0 & 0 & * \end{bmatrix}$$

After computing both Householder transformations, we get:

$$H_2H_1A = R = \begin{bmatrix} \sqrt{7} & \sqrt{7} & \frac{4}{\sqrt{7}} \\ 0 & \sqrt{2} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{\sqrt{42}}{14} \\ 0 & 0 & 0 \end{bmatrix}$$

$$Q = (H_2H_1)^T = H_1^TH_2^T$$

The final reduced QR factorization gives the same result as the Gram-Schmidt method.

2.2 Exercise 2

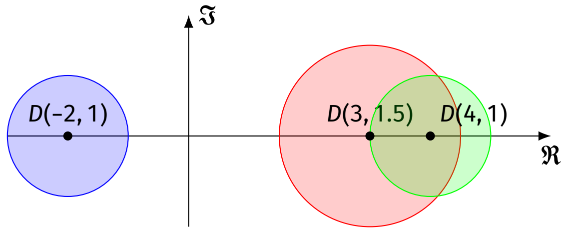
2.2.1 Problem 1: Gershgorin Disks

$$A = \begin{bmatrix} -2 & 1 & 0 \\ 1 & 3 & 0.5 \\ 0.5 & -0.5 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0.5i & 0.5i \\ 0.5 & i & 0.5 \\ -0.5i & -0.5i & 1 + 2i \end{bmatrix}$$

Compute centers and radii (row-form Gershgorin):

For A:

$$\begin{aligned}
a_{11} &= -2, \quad r_1 = |1| + |0| = 1 && \Rightarrow D(-2,1)\\
a_{22} &= 3, \quad r_2 = |1| + |0.5| = 1.5 && \Rightarrow D(3,1.5)\\
a_{33} &= 4, \quad r_3 = |0.5| + |-0.5| = 1 && \Rightarrow D(4,1)
\end{aligned}$$

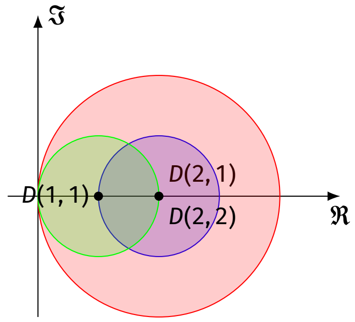


Eigenvalues:

$$\begin{aligned}
\lambda_1 &\in D(-2,1)\\
\lambda_{2,3} &\in D(3,1.5) \cup D(4,1)
\end{aligned}$$

For B:

$$\begin{aligned}
b_{11} &= 2, \quad s_1 = |-1| + |0| = 1 && \Rightarrow D(2,1)\\
b_{22} &= 2, \quad s_2 = |-1| + |-1| = 2 && \Rightarrow D(2,2)\\
b_{33} &= 1, \quad s_3 = |0| + |-1| = 1 && \Rightarrow D(1,1)
\end{aligned}$$

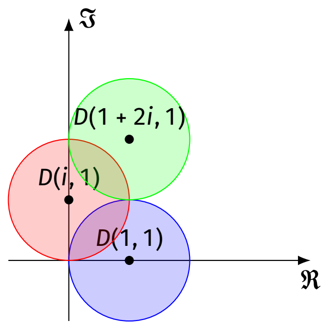


Eigenvalues:

$$\lambda_{1,2,3} \in D(2,2)$$

For C (centers are complex; plot in complex plane):

$$\begin{aligned}
c_{11} &= 1, \quad t_1 = |0.5i| + |0.5i| = 1 && \Rightarrow D(1,1)\\
c_{22} &= i, \quad t_2 = |0.5| + |0.5| = 1 && \Rightarrow D(i,1)\\
c_{33} &= 1+2i, \quad t_3 = |-0.5i| + |-0.5i| = 1 && \Rightarrow D(1+2i,1)
\end{aligned}$$



Eigenvalues:

$$\lambda_{1,2,3} \in D(1,1) \cup D(i,1) \cup D(1+2i,1)$$

2.3 Exercise 3

Problem 2

Let $A \in \mathbb{R}^{n\times n}$ with k distinct eigenvalues $\lambda_1, \ldots, \lambda_k$.

Show that:

$$\operatorname{grade}_A(v) \leq k, \quad \forall v \in \mathbb{R}^n$$

2.4 Exercise 4

2.4.1 Problem 3

Assume that $A \in \mathbb{R}^{n\times n}$ is SPD and that we use the CG method for solving the system $Ax = b$. Assume moreover that the eigenvalues $\lambda_1, \ldots, \lambda_{n-1}$ are distributed in an interval $[\lambda_{\min}, \lambda_{\max}] \subset \mathbb{R}_{>0}$, while the eigenvalue λ_n is "very different" from the others (that is, either much larger than $\lambda_{\max}$ or much closer than $\lambda_{\min}$ to 0).

Find an estimate for the error reduction $\|\mathbf{x}_m - \mathbf{x}^\star\|_A / \|\mathbf{x}_0 - \mathbf{x}^\star\|_A$ after m steps of the CG method. Here $\mathbf{x}^\star = A^{-1}b$ is the exact solution of the system. The estimate should only depend on $\lambda_{\max}, \lambda_{\min}, \lambda_n$, and m.

Solution

We use the estimate:

$$\frac{\|\mathbf{x}_m - \mathbf{x}^\star\|_A}{\|\mathbf{x}_0 - \mathbf{x}^\star\|_A} \leq \max_{i=1,\dots,n} |r(\lambda_i)|$$

for any polynomial r with $\deg(r) \leq m$ and $r(0) = 1$.