TMA4205

Numerical Linear Algebra

Compendium in Numerical Linear Algebra

Author
Trym Sæther
Email: trym.saether@ntnu.no

Semester
Autumn 2025
Last updated: February 28, 2026

Norwegian University of Science and Technology
Department of Mathematical Sciences

Contents

Part I

Foundations of Numerical Linear Algebra

Chapter 1

Introduction

This course covers algorithms for solving linear systems $Ax = b$, eigenvalue problems, and matrix factorizations on computers with finite precision arithmetic.

We study three main categories of *problems*:

- Direct methods: Gaussian elimination, LU/QR/Cholesky factorizations
- Iterative methods: Krylov subspace methods, multigrid, domain decomposition
- Eigenvalue computation: Power method, QR algorithm, Arnoldi/Lanczos methods

The key challenge is balancing computational cost with numerical accuracy. Round-off errors accumulate differently across algorithms, and problem conditioning determines which methods remain stable. We emphasize implementation details and complexity analysis. Most real problems involve sparse matrices where structure must be exploited—dense matrix algorithms often fail due to memory and time constraints.

The material follows Saad (2003) with focus on methods used in practice for large-scale scientific computing.

1.1 Large Sparse Problems

We focus on matrices that are large ($n \gtrsim 10^4$) and sparse. Let $N_z(A)$ denote the number of nonzeros. Storage and matrix–vector products scale with $O(N_z(A))$, whereas dense factorizations cost $O(n^3)$ flops and $O(n^2)$ memory. Consequently, iterative methods driven by sparse matrix–vector products are central to modern scientific computing.

Chapter 2

Preliminaries

2.1 Notation and Conventions

We work over the field $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. Statements that require $\mathbb{C}$ are indicated explicitly.

Vectors are column vectors unless otherwise stated. For $\mathbf{x}, \mathbf{y} \in \mathbb{K}^n$ the inner product is

$$\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{y}^H \mathbf{x},$$

which is conjugate-linear in the first argument and linear in the second.

We define the associated Euclidean 2-norm as our standard norm $\|\cdot\| = \|\cdot\|_2$, unless otherwise specified:

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}.$$

For matrices $A \in \mathbb{K}^{m \times n}$ we use

- A^H for the conjugate transpose, and A^T for the (real) transpose,
- $\overline{A}$ or $\overline{z}$ for elementwise complex conjugation,
- I_n for the $n \times n$ identity and 0 for a suitably sized zero matrix/vector,
- $A_{i,j}$ (or $[A]_{ij}$) for the (i,j) entry and $A_{p:q,r:s}$ for the submatrix with row indices $p, \dots, q$ and column indices $r, \dots, s$,
- $\operatorname{diag}(d_1, \dots, d_n)$ for a diagonal matrix with the given diagonal entries.

Matrix norms and spectral quantities:

$$\|A\|_2 := \max_{\|\mathbf{x}\|_2 = 1} \|A\mathbf{x}\|_2 \qquad \text{(spectral/operator 2-norm)},$$

$$\|A\|_F := \sqrt{\sum_{i,j} |a_{ij}|^2} \qquad \text{(Frobenius norm)},$$

$$\rho(A) := \max_i |\lambda_i(A)| \qquad \text{(spectral radius)}.$$

Other common notation:

- e_i denotes the ith standard basis vector in $\mathbb{K}^n$ and $\mathbf{e} = (1, \dots, 1)^\top$,
- $\operatorname{tr}(A)$ denotes the trace, $\det(A)$ the determinant, and $\operatorname{rank}(A)$ the rank,
- $\Re(z)$ and $\Im(z)$ denote the real and imaginary parts of a complex number z,
- for sequences/functions we use standard asymptotic notation ($O(\cdot)$, $o(\cdot)$) when needed.

These conventions are used throughout the text; any deviation will be stated where it occurs.

2.2 Matrices

2.2.1 Eigenvalues and Eigenvectors

Let $A \in \mathbb{C}^{n\times n}$. A scalar $\lambda \in \mathbb{C}$ and nonzero vector $\mathbf{v} \in \mathbb{C}^n$ satisfy

$$A\mathbf{v} = \lambda\mathbf{v} \qquad \text{(right eigenpair).}$$

Left eigenvectors $\mathbf{w}$ satisfy $\mathbf{w}^H A = \lambda\,\mathbf{w}^H$, equivalently $A^H\mathbf{w} = \bar{\lambda}\,\mathbf{w}$. If A is Hermitian ($A^H = A$), all eigenvalues are real; if A is singular, 0 is an eigenvalue.

2.2.2 Image (Range) and Kernel (Nullspace)

Definition 2.1: Image / Range

The *image* (or *range*) of $A \in \mathbb{R}^{m\times n}$ is

$$\mathrm{Im}(A) = \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} = \mathrm{span}\{\mathbf{a}_1, \dots, \mathbf{a}_n\},$$

where $\mathbf{a}_j$ are the columns of A. The *rank* of A is

$$\mathrm{rank}(A) = \dim(\mathrm{Im}(A)).$$

Definition 2.2: Kernel / Null space

The *kernel* (or *null space*) of $A \in \mathbb{R}^{m\times n}$ is

$$\ker(A) = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\}.$$

Its dimension is the *nullity* of A:

$$\mathrm{nullity}(A) = \dim(\ker(A)).$$

Theorem 2.3: Rank-Nullity Theorem

For any matrix $A \in \mathbb{R}^{m\times n}$,

$$\mathrm{rank}(A) + \mathrm{nullity}(A) = n.$$

Proof sketch. Consider the linear map $x \mapsto Ax$. Choose a basis of $\ker(A)$ and extend it to a basis of $\mathbb{R}^n$. The remaining basis vectors map to a basis of $\mathrm{Im}(A)$, giving the stated equality. □

Immediate consequences of the rank-nullity theorem are:

- A has full column rank iff $\ker(A) = \{0\}$.
- If $m < n$ then $\ker(A) \neq \{0\}$ for rank-deficient A (pigeonhole).
- Solutions of $A\mathbf{x} = \mathbf{b}$ exist iff $\mathbf{b} \in \mathrm{Im}(A)$; when solutions exist they form an affine space $\mathbf{x}_0 + \ker(A)$.

2.2.3 Normal Matrices

A matrix $A \in \mathbb{C}^{n\times n}$ is *normal* if

$$AA^H = A^H A,$$

For real matrices, this becomes $AA^T = A^T A$.

Normal matrices are special because they *commute* with their conjugate transpose; meaning $AA^H = A^H A$. This property guarantees that the matrix has a complete set of orthogonal eigenvectors.

Remark 1. Intuition for Normal Matrices
Think of Normal Matrices like this: most matrices will stretch, rotate, AND skew vectors in complicated ways. But normal matrices are *well-behaved*, they only stretch or shrink along specific perpendicular directions, without mixing them up. This makes them much easier to understand and work with.

Theorem 2.4: Spectral Theorem for Normal Matrices

A matrix $A \in \mathbb{C}^{n\times n}$ is normal if and only if it admits a unitary diagonalization:

$$A = UDU^H,$$

where $U \in \mathbb{C}^{n\times n}$ is unitary ($U^HU = I$) and $D = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ contains the eigenvalues.

This characterization shows that normal matrices are precisely those with a complete orthonormal basis of eigenvectors. The geometric intuition is that normal matrices preserve orthogonality when acting on their eigenspaces.

Important subclasses of normal matrices:

Hermitian matrices: $A = A^H$, which implies all eigenvalues are real: $\lambda_i \in \mathbb{R}$.
Skew-Hermitian matrices: $A = -A^H$, which implies all eigenvalues are purely imaginary: $\lambda_i \in i\mathbb{R}$.
Unitary matrices: $A^HA = I$, which implies all eigenvalues have unit modulus: $|\lambda_i| = 1$.

2.2.4 Hermitian Matrices

Definition 2.5: Hermitian (Self-adjoint)

A matrix $A \in \mathbb{C}^{n\times n}$ is *Hermitian*(or *self-adjoint*) if

$$A = A^H,$$

that is, A equals its conjugate transpose. Over the reals, this condition reduces to symmetry: $A = A^\top$.

Remark 2. Intuition
Hermitian matrices are the complex analogue of real symmetric matrices. They have a built-in geometric symmetry: being equal to their own conjugate transpose makes them "balanced" across the main diagonal. This structure guarantees real eigenvalues and an orthonormal eigenbasis, which makes Hermitian matrices highly predictable and stable compared to general matrices.

Theorem 2.6: Spectral Theorem for Hermitian Matrices

If $A \in \mathbb{C}^{n\times n}$ is Hermitian, then:
1. All eigenvalues are real: $\lambda_i \in \mathbb{R}$.
2. There exists an orthonormal basis of eigenvectors.
3. A admits a unitary diagonalization:
$$A = UDU^H,$$
where U is unitary and D is a real diagonal matrix of eigenvalues.

Proof sketch: eigenvalues are real. Let λ be an eigenvalue of A with eigenvector $\mathbf{v} \neq 0$. Then

$$\lambda\|\mathbf{v}\|^2 = \lambda\mathbf{v}^H\mathbf{v} = \mathbf{v}^HA\mathbf{v} = \mathbf{v}^HA^H\mathbf{v} = (A\mathbf{v})^H\mathbf{v} = (\lambda\mathbf{v})^H\mathbf{v} = \bar{\lambda}\,\|\mathbf{v}\|^2.$$

Since $\|\mathbf{v}\|^2 > 0$, we conclude $\lambda = \bar{\lambda}$, hence $\lambda \in \mathbb{R}$. □

Variational characterization. For Hermitian A, the eigenvalues sorted nonincreasingly satisfy the *min-max principle*:

$$\lambda_k(A) = \min_{\dim S=k} \max_{\substack{\mathbf{x}\in S \\ \|\mathbf{x}\|=1}} \mathbf{x}^H A\mathbf{x}.$$

The Rayleigh quotient $R_A(\mathbf{x}) = \dfrac{\mathbf{x}^H A\mathbf{x}}{\mathbf{x}^H\mathbf{x}}$ obeys $\min R_A = \lambda_n$, $\max R_A = \lambda_1$.

Computational advantages. Hermitian structure halves storage, enables real arithmetic when $A \in \mathbb{R}^{n\times n}$, and admits stable, cost-effective eigenvalue algorithms (e.g., tridiagonal reduction + QR). Perturbation theory is particularly benign: eigenvalues satisfy Weyl's theorem; eigenvectors obey Davis-Kahan bounds when eigenvalue gaps are present.

> **Example 1. Quadratic Forms**
>
> For Hermitian A and vector $\mathbf{x}$, the quadratic form $\mathbf{x}^H A\mathbf{x}$ is always real. Using the spectral decomposition:
>
> $$\mathbf{x}^H A\mathbf{x} = \mathbf{x}^H UDU^H\mathbf{x} = \sum_{i=1}^{n} \lambda_i |u_i^H\mathbf{x}|^2$$
>
> This shows how the eigenvalues directly control the behavior of the quadratic form.

The special structure of normal and Hermitian matrices makes them the foundation for many numerical algorithms, from eigenvalue computation to optimization methods that rely on their predictable spectral behavior.

2.2.5 Nonnegative Matrices

Nonnegative matrices are widely used in applications involving positive quantities, such as probability distributions, population dynamics, economic models, and network analysis. Their spectral properties are described by the Perron-Frobenius theory, which governs their eigenvalue structure.

Definition 2.7: Nonnegative Matrix

A matrix $A \in \mathbb{R}^{n\times n}$ is *nonnegative* if $a_{ij} \geq 0$ for all i, j. We write $A \geq 0$.
A matrix is *positive* if $a_{ij} > 0$ for all i, j, denoted $A > 0$.

Theorem 2.8: Perron-Frobenius Theorem

Let $A \geq 0$ be a nonnegative matrix. Then:

1. The spectral radius $\rho(A) = \max_i |\lambda_i|$ is an eigenvalue of A.
2. There exists a nonnegative eigenvector $\mathbf{x} \geq 0$ such that $A\mathbf{x} = \rho(A)\mathbf{x}$.
3. If A is irreducible, then $\rho(A)$ is a simple eigenvalue, and there exists a positive eigenvector $\mathbf{x} > 0$ such that $A\mathbf{x} = \rho(A)\mathbf{x}$.
4. If A is irreducible and aperiodic, then $\rho(A)$ is the unique eigenvalue of maximum modulus, i.e., $|\lambda| < \rho(A)$ for all other eigenvalues λ.

The Perron-Frobenius theorem guarantees that nonnegative matrices have a dominant eigenvalue $\rho(A)$, which is real and positive. This eigenvalue corresponds to a nonnegative eigenvector, and in the case of irreducibility, a strictly positive eigenvector.

The dominant eigenvalue $\rho(A)$ often represents the long-term growth rate or stability of the system described by A.

2.2.6 Kantorovich Inequality

Kantorovich inequality provides bounds on the relationship between different norms induced by a symmetric positive definite matrix.

Theorem 2.9: Kantorovich Inequality

Let $B \in \mathbb{R}^{n\times n}$ be SPD with eigenvalues $0 < \lambda_1 \le \cdots \le \lambda_n$. Then for all $\mathbf{x} \in \mathbb{R}^n$,

$$\frac{\|\mathbf{x}\|_B^2 \, \|\mathbf{x}\|_{B^{-1}}^2}{\|\mathbf{x}\|_2^4} \le \frac{1}{4} \cdot \frac{(\lambda_1 + \lambda_n)^2}{\lambda_1 \lambda_n}.$$

Proof. Since B is SPD, diagonalize $B = Q^\top \Lambda Q$ with $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ and Q orthogonal. For $y = Q\mathbf{x}$ with $\|\mathbf{x}\|_2 = \|y\|_2 = 1$:

$$B^{-1} = Q^\top \Lambda^{-1} Q,$$

$$\|\mathbf{x}\|_B^2 = \mathbf{x}^\top B \mathbf{x} = \sum_{i=1}^n \lambda_i y_i^2,$$

$$\|\mathbf{x}\|_{B^{-1}}^2 = \mathbf{x}^\top B^{-1} \mathbf{x} = \sum_{i=1}^n \lambda_i^{-1} y_i^2.$$

Thus $\left(\bar{\lambda}, \bar{\lambda}^{-1}\right)$ with

$$\bar{\lambda} = \sum_{i=1}^n \lambda_i y_i^2, \qquad \bar{\lambda}^{-1} = \sum_{i=1}^n \lambda_i^{-1} y_i^2,$$

is a convex combination of points $(\lambda_i, 1/\lambda_i)$.
The curve $1/\lambda$ is convex on $(0, \infty)$, hence

$$\left(\bar{\lambda}, \bar{\lambda}^{-1}\right)$$

lies below the chord

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

Therefore

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

The maximum of $q(\bar{\lambda}) = \bar{\lambda}\, \ell(\bar{\lambda})$ occurs at $\bar{\lambda} = \frac{1}{2}(\lambda_1 + \lambda_n)$, yielding

$$\bar{\lambda}\, \bar{\lambda}^{-1} \le \max_{\bar{\lambda} \in [\lambda_1, \lambda_n]} q(\bar{\lambda}) = \frac{(\lambda_1 + \lambda_n)^2}{4 \lambda_1 \lambda_n}$$

□ □

2.2.7 M-matrices

M-matrices are matrices with nonpositive off-diagonal entries and a nonnegative inverse. They ensure stability and monotonicity, making them useful in numerical analysis, optimization, and modeling problems with positivity constraints.

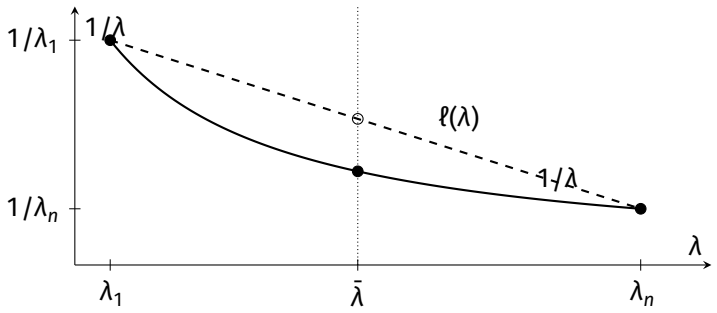


Figure 2.1: Kantorovich inequality visualized via the convexity of $1/\lambda$ and its chord between λ_1 and λ_n.

Definition 2.10: M-matrix

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

1. $a_{ij} \le 0$ for all $i \neq j$ (nonpositive off-diagonal entries)
2. A is nonsingular
3. $A^{-1} \ge 0$ (nonnegative inverse)

Corollary 1: (M-matrix characterization)**.** An M-matrix can be written as $A = sI - B$ where $s > \rho(B)$ and $B \ge 0$.

Properties of M-matrices

Let A be an M-matrix. Then:

1. All eigenvalues have positive real parts: $\mathrm{Re}(\lambda_i) > 0$ for all i.
2. All principal minors are positive: $\det(A_{ij}) > 0$ for all principal submatrices A_{ij}.
3. A is positive stable: solutions to $\mathbf{x}' = -A\mathbf{x}$ decay exponentially.
4. The linear system $A\mathbf{x} = \mathbf{b}$ with $\mathbf{b} \ge 0$ has solution $\mathbf{x} \ge 0$.

Their positive inverse property makes them particularly well-suited for iterative solution methods, as they preserve nonnegativity and ensure convergence.

Example 2. Discrete Laplacian as M-matrix

Consider the discrete 1D Laplacian on n interior points with Dirichlet boundary conditions:

$$A = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 \end{bmatrix}$$

This matrix satisfies the M-matrix conditions: the diagonal entries are positive, off-diagonal entries are nonpositive, and the matrix is positive definite (hence $A^{-1} > 0$). This structure ensures that the discrete maximum principle holds for the corresponding difference equations.

2.2.8 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:

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

Example 3. 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.

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.

2.3 Canonical Forms and Matrix Structure

Canonical forms provide standardized representations that reveal the essential structure of mathematical objects. In numerical linear algebra, they serve both theoretical and computational purposes, offering insight into matrix properties and serving as targets for numerical algorithms.

Definition 2.11: Canonical Form

A canonical form is a unique representative chosen from each equivalence class of objects under a given equivalence relation, selected according to a fixed rule or procedure.

Understanding canonical forms helps us recognize when two apparently different matrices share the same fundamental properties and provides roadmaps for developing efficient algorithms.

2.3.1 Similarity of Matrices

Two matrices $A, B \in \mathbb{C}^{n\times n}$ are *similar* if there exists an invertible matrix X such that

$$B = X^{-1}AX.$$

Similarity defines an equivalence relation on the set of square matrices, partitioning them into equivalence classes where matrices within the same class share the same eigenvalues (counting multiplicities) and many other spectral properties.

Remark 3. Intuition for Similarity

Similarity transformations correspond to changing the basis of the vector space. If we think of a matrix as representing a linear transformation with respect to a particular basis, then similar matrices represent the same transformation but expressed in different bases. This explains why similar matrices have the same eigenvalues: eigenvalues are invariant under basis changes.

Theorem 2.12: Properties Preserved by Similarity

If A and B are similar matrices, then they share the following properties:

1. eigenvalues (including algebraic multiplicities).
2. characteristic polynomial: $\det(\lambda I - A) = \det(\lambda I - B)$
3. trace: $\operatorname{tr}(A) = \operatorname{tr}(B)$
4. determinant: $\det(A) = \det(B)$
5. rank: $\operatorname{rank}(A) = \operatorname{rank}(B)$
6. minimal polynomial: $\mu_A(\lambda) = \mu_B(\lambda)$

Proof sketch. Most properties follow directly from the similarity transformation.
For eigenvalues: if $A\mathbf{v} = \lambda\mathbf{v}$, then

$$\begin{aligned} B(X^{-1}\mathbf{v}) &= X^{-1}AX(X^{-1}\mathbf{v}) \\ &= X^{-1}A\mathbf{v} \\ &= X^{-1}(\lambda\mathbf{v}) \\ &= \lambda X^{-1}\mathbf{v} \end{aligned}$$

The characteristic polynomial follows from the eigenvalues, and trace/determinant are polynomial functions of the eigenvalues. □

Canonical forms are essentially unique representatives of similarity equivalence classes, chosen according to specific rules (e.g., diagonal form for diagonalizable matrices, Jordan form for general matrices).

2.3.2 Affine Spaces and Affine Maps

Definition 2.13: Affine subspace

Let $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. An *affine subspace* of $\mathbb{K}^n$ is a translate of a linear subspace. For a point $\mathbf{p} \in \mathbb{K}^n$ and a linear subspace $V \subseteq \mathbb{K}^n$ the set

$$\mathcal{A} = \mathbf{p} + V = \{\mathbf{p} + \mathbf{v} : \mathbf{v} \in V\}$$

is an affine subspace. The subspace V is called the *direction* of $\mathcal{A}$.

A finite set of points $\{\mathbf{x}_1, \dots, \mathbf{x}_k\} \subset \mathbb{K}^n$ is *affinely independent* if the vectors

$$\mathbf{x}_2 - \mathbf{x}_1, \ \dots, \ \mathbf{x}_k - \mathbf{x}_1$$

are linearly independent. The *affine hull* of a set $S \subset \mathbb{K}^n$, denoted $\operatorname{aff}(S)$, is the smallest affine subspace containing S. Equivalently,

$$\operatorname{aff}(S) = \Big\{ \sum_i \alpha_i \mathbf{x}_i : \mathbf{x}_i \in S, \ \sum_i \alpha_i = 1 \Big\},$$

the set of all affine combinations of points in S.

Definition 2.14: Affine map

An *affine map* is a function $f : \mathbb{K}^n \to \mathbb{K}^m$ of the form

$$f(\mathbf{x}) = A\mathbf{x} + \mathbf{b},$$

where $A \in \mathbb{K}^{m \times n}$ is the linear part and $\mathbf{b} \in \mathbb{K}^m$ is a translation.

Let $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ be an affine map. Then

1. f preserves affine combinations; in particular, f maps affine subspaces to affine subspaces.
2. f is linear if and only if $\mathbf{b} = \mathbf{0}$.
3. The composition of two affine maps is affine.

2.3.3 Matrix Polynomials

A polynomial $p(t) = \sum_{k=0}^{d} c_k t^k \in \mathbb{K}[t]$ acts on a matrix $A \in \mathbb{K}^{n \times n}$ by

$$p(A) = \sum_{k=0}^{d} c_k A^k.$$

The set $\{p(A) : p \in \mathbb{K}[t]\}$ is a *commutative subalgebra* of $\mathbb{K}^{n \times n}$.

> **Remark 4. commutative subalgebra**
>
> The set
>
> $$\{p(A) : p \in \mathbb{K}[t]\} \subseteq \mathbb{K}^{n \times n}$$
>
> is a subalgebra: it contains 0 and I, is closed under addition and scalar multiplication, and satisfies $(pq)(A) = p(A)q(A)$, so it is closed under multiplication. Since all elements are polynomials in the same matrix A, they commute, and the subalgebra is commutative.

Example 4. Commutative subalgebra

If $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, then

$$\{p(A) : p \in \mathbb{R}[t]\} = \left\{ \begin{bmatrix} c_0 & c_1 \\ 0 & c_0 \end{bmatrix} : c_0, c_1 \in \mathbb{R} \right\}.$$

This is a 2-dimensional subalgebra of $\mathbb{R}^{2\times 2}$.

Minimal Polynomial The *minimal polynomial* $\mu_A(t)$ of $A \in \mathbb{K}^{n\times n}$ is the unique monic polynomial of smallest degree satisfying

$$\mu_A(A) = 0.$$

For $A \in \mathbb{K}^{n\times n}$ the minimal polynomial $\mu_A(t)$ satisfies:

1. μ_A divides every polynomial p with $p(A) = 0$.
2. The distinct roots of μ_A are precisely the eigenvalues of A.
3. For each eigenvalue λ, the multiplicity of $(t - \lambda)$ in μ_A equals the size of the largest Jordan block of A associated with λ.
4. $\deg(\mu_A) \leq n$ and μ_A divides the characteristic polynomial $\chi_A(t) = \det(tI - A)$.

Knowledge of μ_A determines the smallest polynomial algebra containing A.

Example 5. I

A is diagonalizable with distinct eigenvalues $\{\lambda_1, \dots, \lambda_r\}$, then

$$\mu_A(t) = \prod_{i=1}^{r} (t - \lambda_i).$$

2.3.4 Jordan Canonical Form

Jordan form reveals the fine structure of linear transformations, particularly the behavior of eigenspaces and generalized eigenspaces.

A Jordan block of size k with eigenvalue λ is the $k \times k$ matrix

$$J_k(\lambda) = \begin{bmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda & 1 \\ & & & & \lambda \end{bmatrix}.$$

The superdiagonal of 1's captures the action on generalized eigenvectors.

Jordan blocks represent the building blocks of linear transformations that are "as close to diagonal as possible" when diagonalization is not achievable.

Definition 2.15: Jordan Canonical Form

Every square matrix $A \in \mathbb{C}^{n\times n}$ is similar to a block diagonal matrix

$$J = \begin{bmatrix} J_{k_1}(\lambda_1) & & & \\ & J_{k_2}(\lambda_2) & & \\ & & \ddots & \\ & & & J_{k_r}(\lambda_r) \end{bmatrix},$$

where each $J_{k_i}(\lambda_i)$ is a Jordan block. The Jordan form is unique up to permutation of blocks.

The Jordan form provides complete information about the eigenvalue structure and the geometric versus algebraic multiplicities of eigenvalues.

Remark 5. Numerical Considerations for Jordan Form

Despite its theoretical importance, Jordan form is numerically unstable to compute. The structure is highly sensitive to perturbations: arbitrarily small changes in matrix entries can dramatically alter the Jordan block structure. This makes Jordan form unsuitable for practical numerical computation, though it remains valuable for theoretical analysis.

2.3.5 Schur Decomposition

The Schur decomposition provides a numerically stable similarity transformation that triangularizes a matrix while preserving its spectrum.

Theorem 2.16: Schur Decomposition

Every $A \in \mathbb{C}^{n\times n}$ admits a Schur decomposition

$$A = QTQ^H,$$

where Q is unitary and T is upper triangular with eigenvalues of A on its diagonal.

Proof by induction.
Base case: For $n = 1$, any 1×1 matrix $A = [\lambda]$ is trivially upper triangular, and we can take $Q = [1]$.
Inductive step: Assume the theorem holds for $(n-1)\times(n-1)$ matrices. Let $A \in \mathbb{C}^{n\times n}$, with eigenpair $(\lambda, \mathbf{v})$ where $\|\mathbf{v}\| = 1$.
Choose $\tilde{Q}_1 \in \mathbb{C}^{n\times(n-1)}$ s.t. $\mathbf{v}^H\tilde{Q}_1 = 0$ and $Q_1 = [\mathbf{v}, \tilde{Q}_1]$ s.t. $Q_1^HQ_1 = I$ (unitary).
Then

$$Q_1^HAQ_1 = \begin{bmatrix}\mathbf{v}^H\\ \tilde{Q}_1^H\end{bmatrix}A\begin{bmatrix}\mathbf{v} & \tilde{Q}_1\end{bmatrix} = \begin{bmatrix}\lambda & \mathbf{v}^HA\tilde{Q}_1\\ 0 & \tilde{Q}_1^HA\tilde{Q}_1\end{bmatrix} =: \begin{bmatrix}\lambda & \mathbf{w}^H\\ 0 & A_1\end{bmatrix}.$$

□

Properties of the Schur Form The Schur form T satisfies:

1. $\det(\lambda I - T) = \det(\lambda I - A)$
2. If A is normal, T can be chosen diagonal
3. Small ΔA implies small ΔT (backward stability)

Computing the Schur Decomposition via the QR Algorithm

The QR algorithm computes the Schur decomposition iteratively:

Algorithm 1 Basic QR Algorithm for Schur Decomposition

Require: $A^{(0)} = A$, $Q^{(0)} = I$
1: **for** $k = 1, 2, \ldots$ **do**
2: $A^{(k-1)} = Q_kR_k$ (QR decomposition)
3: $A^{(k)} = R_kQ_k$
4: $Q^{(k)} = Q^{(k-1)}Q_k$
Ensure: $T = A^{(\infty)}$, $Q = Q^{(\infty)}$

Visualization. A Householder reflector $P = I - 2uu^\top$ flips the component of a vector parallel to u and leaves the orthogonal component unchanged. The figure illustrates $x = \pi_u(x) + (x - \pi_u(x))$ and $Px = -\pi_u(x) + (x - \pi_u(x))$.

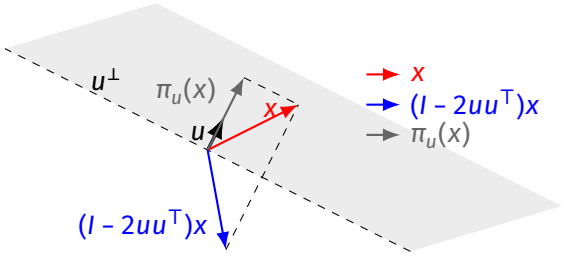


Figure 2.2: Householder reflection across the hyperplane orthogonal to u.

Applications of the Schur Form

- Eigenvalue computation: $\lambda_i = t_{ii}$
- Matrix functions: $f(A) = Qf(T)Q^H$
- Stability analysis via triangular structure
- Pseudospectral computations

Example 6

or $A = \begin{bmatrix} 1 & 2 \\ -1 & 4 \end{bmatrix}$, eigenvalues $\lambda = \frac{5 \pm \sqrt{17}}{2}$. Schur form has these on diagonal with unitary Q.

Remark 6. Comparison of Jordan and Schur Forms

Jordan form reveals block structure but is numerically unstable due to sensitivity to perturbations. The Schur form provides stable triangularization; Jordan blocks can be inferred from T but with care.

Givens Rotations

Givens rotations annihilate single entries by acting on two rows at a time. For indices $i < j$ and scalars c, s with $c^2 + s^2 = 1$, define

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

Applied from the left, G mixes rows i and j. To zero an entry b under a pivot a, choose

$$r = \sqrt{a^2 + b^2}, \qquad c = \frac{a}{r}, \; s = \frac{b}{r}, \qquad \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

Iterating over columns and eliminating entries below the diagonal yields an upper triangular R; accumulating the applied rotations gives Q so that $A = QR$.

Algorithm 2 QR via Givens Rotations (outline)

Require: $A \in \mathbb{R}^{m\times n}$, $m \geq n$
  $Q \leftarrow I_m$
  for $k = 1, \dots, n$ **do**
    for $i = m, \dots, k+1$ **do**
      $(a, b) \leftarrow (A_{i-1,k}, A_{i,k})$, compute (c, s) as above
      Apply $\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ to rows $i-1, i$ of A (left multiply)
      Apply same rotation to rows $i-1, i$ of Q
  $R \leftarrow A$, return Q, R with $A = QR$

Givens is attractive for sparse and structured problems because each rotation affects only two rows, limiting fill-in, and for streaming least-squares where rotations can be applied incrementally. It is also the tool used to update the GMRES least-squares; see Section **??**.

Remark 7. Householder vs Givens

- Dense QR: Householder is typically faster (BLAS-3 friendly), more stable, and easier to block.
- Sparse/structured: Givens can reduce fill locally and target individual entries.
- Streaming LS / online updates: Givens supports incremental QR updates with simple 2×2 rotations.
- Parallelism: Householder blocks vector–matrix operations; Givens exposes fine-grained parallelism on independent rotations.

2.4 Gershgorin's Theorem and Diagonal Dominance

Gershgorin's theorem provides a simple geometric method to localize eigenvalues using only the matrix entries, without computing them explicitly. It reveals a fundamental connection between diagonal dominance and spectral properties, with important consequences for stability and nonsingularity.

2.4.1 Gershgorin Discs

For a matrix $A = (a_{ij}) \in \mathbb{C}^{n\times n}$, define the *off-diagonal row sum*

$$R_i = \sum_{j\neq i} |a_{ij}|.$$

The *Gershgorin disc* centered at the diagonal entry a_{ii} is

$$D_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i\}.$$

Theorem 2.17: Gershgorin Circle Theorem

Every eigenvalue of A lies in at least one Gershgorin disc:

$$\sigma(A) \subseteq \bigcup_{i=1}^{n} D_i.$$

Moreover, if k discs form a connected component that is disjoint from the remaining $n-k$ discs, then exactly k eigenvalues (counting multiplicities) lie in that component.

Proof. Let λ be an eigenvalue with eigenvector $\mathbf{v}$. Choose index m where $|v_m| = \|\mathbf{v}\|_\infty$. The m-th component of $A\mathbf{v} = \lambda\mathbf{v}$ gives

$$\lambda v_m = a_{mm} v_m + \sum_{j \neq m} a_{mj} v_j.$$

Rearranging:

$$|\lambda - a_{mm}| \cdot |v_m| = \left| \sum_{j \neq m} a_{mj} v_j \right| \leq \sum_{j \neq m} |a_{mj}| \, |v_j| \leq R_m \, |v_m|,$$

hence $\lambda \in D_m$.
The separation property follows by continuity: consider the homotopy $A(t) = D + tH$ where $D = \mathrm{diag}(a_{11}, \ldots, a_{nn})$ and $H = A - D$. At $t = 0$, eigenvalues are $\{a_{ii}\}$; at $t = 1$, they reach $\sigma(A)$. Eigenvalues vary continuously with t, and isolated disc components trap eigenvalues by topological degree arguments. □

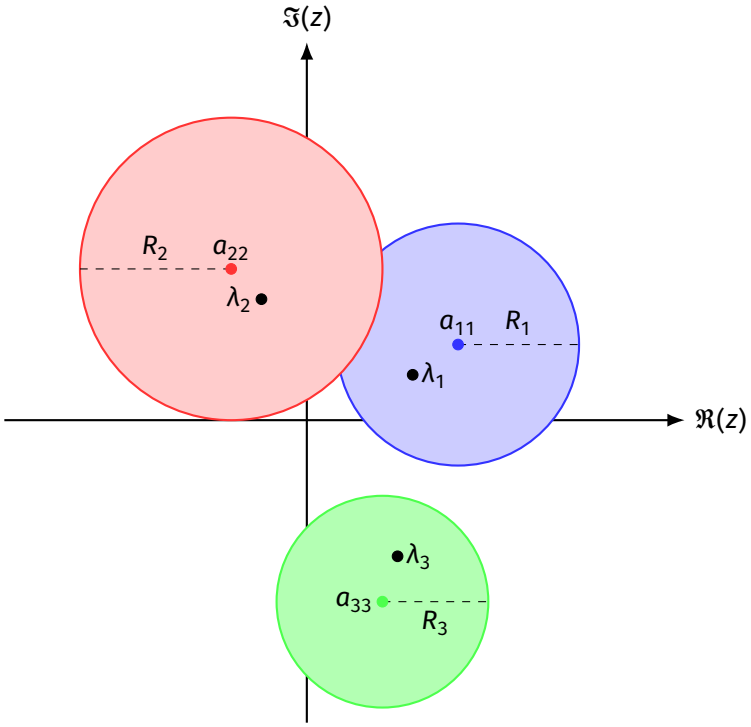


Figure 2.3: Gershgorin discs D_i in the complex plane. Each eigenvalue must lie within at least one disc. Isolated disc clusters contain a number of eigenvalues equal to the number of discs in the cluster.

Irreducible matrices. A matrix A is *irreducible* if its directed graph is strongly connected, or equivalently, no permutation reduces A to block upper-triangular form with a zero block below the diagonal. For irreducible matrices, if any eigenvalue lies on the boundary of one disc, it must lie on the boundary of *all* discs—a strong global constraint.

2.4.2 Diagonal Dominance and Nonsingularity

A matrix A is *strictly diagonally dominant by rows* if

$$|a_{ii}| > R_i \qquad \text{for all } i = 1, \ldots, n.$$

Geometrically, this means each Gershgorin disc D_i excludes the origin.

Levy–Desplanques theorem. If A is strictly diagonally dominant, then A is nonsingular.

Proof. Since $0 \notin D_i$ for any i, Gershgorin's theorem implies $0 \notin \sigma(A)$. □

Varga's criterion (irreducible case). If A is irreducible and weakly diagonally dominant ($|a_{ii}| \geq R_i$ for all i) with strict inequality for at least one row, then A is nonsingular.

Proof sketch. If 0 were an eigenvalue, irreducibility forces 0 to lie on the boundary of all discs. But at least one disc strictly excludes the origin, a contradiction. □

Perturbation insight. Consider the homotopy $A(t) = D + tH$. At $t = 0$, eigenvalues coincide with diagonal entries; as $t \to 1$, they move continuously within shrinking discs. This perspective explains why diagonal dominance stabilizes spectra under perturbations and why iterative methods (which implicitly reduce off-diagonal coupling) drive eigenvalues toward diagonal entries.

Chapter 3

Linear Systems

Consider the linear system:

$$A\mathbf{x} = \mathbf{b}, \tag{3.1}$$

where $A \in \mathbb{R}^{m\times n}$ is a given matrix, $\mathbf{x} \in \mathbb{R}^n$ is the vector of unknowns, and $\mathbf{b} \in \mathbb{R}^m$ is the right-hand side vector.

3.1 Types of Linear Systems

- **Overdetermined Systems:** When $m > n$, the system is *overdetermined*. Such systems often arise in data fitting and regression problems. In general, there may be no exact solution, and we seek a least-squares solution that minimizes the residual norm:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|_2.$$

- **Underdetermined Systems:** When $m < n$, the system is *underdetermined*. These systems have infinitely many solutions if consistent, and we often seek the minimum-norm solution:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{x}\|_2.$$

- **Square Systems:** when $m = n$, the system is *square*. If A is non-singular, there exists a unique solution given by:

$$\mathbf{x} = A^{-1}\mathbf{b}.$$

- **Homogeneous Systems:** If a linear system is *homogeneous* then:

$$A\mathbf{x} = \mathbf{0}.$$

3.2 Existence and Uniqueness of Solutions

The solvability of system (3.1) depends on the relationship between $\mathbf{b}$ and the fundamental subspaces of A:

No Solution (Inconsistent System): The system has *no solution* when $\mathbf{b} \notin \text{Im}(A)$. This means the right-hand side vector lies outside the column space of A.

Unique Solution: The system has a *unique solution* when:

- $\mathbf{b} \in \mathrm{Im}(A)$ (consistency condition), and
- $\mathrm{rank}(A) = n$ (full column rank).

When A is square ($m = n$) and invertible, the unique solution is: $\mathbf{x} = A^{-1}\mathbf{b}$.

Infinitely Many Solutions: The system has *infinitely many solutions* when:

- $\mathbf{b} \in \mathrm{Im}(A)$ (consistency condition), and
- $\mathrm{rank}(A) < n$ (rank deficient).

The general solution has the form:

$$\mathbf{x} = \mathbf{x}_p + \mathbf{x}_h,$$

where $\mathbf{x}_p$ is any particular solution satisfying $A\mathbf{x}_p = \mathbf{b}$, and $\mathbf{x}_h \in \ker(A)$ is any solution to the homogeneous system $A\mathbf{x}_h = \mathbf{0}$.

3.3 Methods for Solving Linear Systems

Various numerical methods exist for solving linear systems:

Direct Methods: Compute the exact solution (up to rounding errors) in a finite number of steps.
- *Gaussian elimination*: Reduces the system to row echelon form
- *LU decomposition*: Factorizes $A = LU$ with L lower triangular and U upper triangular
- *Cholesky decomposition*: For symmetric positive definite matrices

Iterative Methods: Generate a sequence of approximations converging to the solution.
- *Stationary methods*: Jacobi, Gauss-Seidel, SOR
- *Krylov subspace methods*: Conjugate Gradient, GMRES, BiCGSTAB

3.4 Matrix Storage

Practical computations use compressed formats instead of dense storage. Common schemes include:

- CSR (Compressed Sparse Row): arrays `values`, `col_idx`, and `row_ptr` for fast SpMV and row access.
- CSC (Compressed Sparse Column): column-oriented analogue of CSR; favors column operations.
- COO (Coordinate): triples (row, col, value); simple to assemble, converted to CSR/CSC for compute.

We denote by $N_z(A)$ the number of nonzeros; memory and SpMV cost scale with $O(N_z(A))$.

3.4.1 Model Problem: 2D Poisson and Sparsity

A standard test case is the 2D Poisson equation $-\Delta u = f$ on $\Omega = (0,1)^2$ with Dirichlet data. Using a five-point stencil on an $N \times N$ interior grid ($h = 1/(N+1)$) gives

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

After ordering the unknowns, we obtain $A\mathbf{u} = \mathbf{f}$ where $A \in \mathbb{R}^{N^2 \times N^2}$ is sparse, symmetric, and block tridiagonal with tridiagonal blocks.

Definition 3.1: Banded matrix

A matrix $A = (a_{ij})$ is *banded* with bandwidth $m_u + m_l + 1$ if $a_{ij} = 0$ whenever $|i - j| > m_u + m_l$, where m_u and m_l are the upper and lower bandwidths.

For the 2D Laplacian with natural ordering, bandwidth$(A) = 2N + 1$. Sparse direct factorizations of banded matrices preserve the band but generally introduce *fill-in*. Exploiting sparsity and structure is essential for large n.

3.4.2 Spectrum of the Discrete 2D Laplacian

For the $N \times N$ five-point Laplacian on $(0,1)^2$ with Dirichlet data, the eigenpairs are known in closed form. Enumerating interior grid indices $(i,j) = 1,\dots,N$,

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

Hence $\lambda_{\min} = 4 - 4\cos\left(\frac{\pi}{N+1}\right)$ and $\lambda_{\max} \approx 4$. The condition number grows like $O\big((N+1)^2\big)$, which implies slow convergence for basic gradient-like methods on fine grids unless preconditioned.

Example 7. Classification of Solutions

Consider the simple 2×2 diagonal systems to illustrate the three cases: $\begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}\mathbf{x} = \begin{pmatrix} \\ 8 \end{pmatrix}$ Since rank$(A) = 2 = n$ and A is invertible:

$$\mathbf{x} = \begin{pmatrix} \frac{1}{2} \\ 2 \end{pmatrix}$$

Let $\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}\mathbf{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. Here rank$(A) = 1 < n = 2$ and $\mathbf{b} \in \mathrm{Im}(A)$. The general solution is:

$$\mathbf{x}(t) = \begin{pmatrix} \frac{1}{2} \\ t \end{pmatrix}, \quad t \in \mathbb{R}$$

Finally, consider the system: $\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Since $\mathbf{b} \notin \mathrm{Im}(A)$ (the second component of $\mathbf{b}$ is nonzero while the second row of A is zero), the system is inconsistent (no solution).

3.5 Perturbation Analysis

When solving linear systems numerically, we often want to understand how sensitive the solution is to small changes in the input data.

3.5.1 Perturbation Framework

Consider the linear system $A\mathbf{x} = \mathbf{b}$ where $A \in \mathbb{R}^{n\times n}$ is non-singular. Let $\mathbf{x}$ be the exact solution. Now consider perturbations in both the coefficient matrix and right-hand side:

$$\tilde{A} = A + \Delta A,$$
$$\tilde{\mathbf{b}} = \mathbf{b} + \Delta\mathbf{b},$$

where ΔA and $\Delta\mathbf{b}$ represent small perturbations. The perturbed system becomes:

$$(A + \Delta A)\tilde{\mathbf{x}} = \mathbf{b} + \Delta\mathbf{b}. \tag{3.5}$$

Let $\tilde{\mathbf{x}} = \mathbf{x} + \Delta\mathbf{x}$ denote the solution to the perturbed system. Substituting into (3.5.1) and using $A\mathbf{x} = \mathbf{b}$:

$$A\Delta\mathbf{x} + \Delta A(\mathbf{x} + \Delta\mathbf{x}) = \Delta\mathbf{b}.$$

For sufficiently small perturbations, we neglect the second-order term $\Delta A \Delta \mathbf{x}$ to obtain the **first-order perturbation equation**:

$$A\Delta\mathbf{x} = \Delta\mathbf{b} - \Delta A\mathbf{x}.$$

Since A is non-singular, we can solve for the change in solution:

$$\Delta\mathbf{x} = A^{-1}\Delta\mathbf{b} - A^{-1}\Delta A\mathbf{x}.$$

This relationship shows how perturbations in the data propagate to the solution.

3.5.2 Condition Number

The *condition number* of an invertible matrix A with respect to a matrix norm $\|\cdot\|$ is defined as:

$$\kappa(A) = \|A\|\|A^{-1}\|.$$

The condition number quantifies the sensitivity of the linear system to perturbations and has the following key properties:

- $\kappa(A) \geq 1$ for any invertible matrix.
- $\kappa(A) = 1$ if and only if A is a scaled orthogonal matrix (see **??**).
- $\kappa(A) = +\infty$ if A is singular.
- $\kappa(\alpha A) = \kappa(A)$ for any $\alpha \neq 0$.

3.5.3 Perturbation Bounds

Let A be invertible and assume $\|\Delta A\| < \frac{1}{\|A^{-1}\|}$. Then:

- **Right-hand side perturbation only:** If $\Delta A = 0$, then

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \kappa(A)\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}.$$

- **Matrix perturbation only:** If $\Delta\mathbf{b} = 0$, then

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}}\frac{\|\Delta A\|}{\|A\|}.$$

- **General case:** For both perturbations,

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}\right).$$

Based on the condition number, we classify matrices as:

Well-conditioned: $\kappa(A)$ is small (typically $\kappa(A) \leq 10^{12}$ in double precision)
Ill-conditioned: $\kappa(A)$ is large, making the system sensitive to perturbations
Singular: $\kappa(A) = +\infty$, indicating the matrix is not invertible

Example 8. Hilbert Matrix

The $n \times n$ Hilbert matrix has entries $H_{ij} = \frac{1}{i+j-1}$. These matrices are notoriously ill-conditioned:

$$H_3 = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}, \quad \kappa_2(H_3) \approx 524.$$

For larger n, the condition number grows exponentially: $\kappa_2(H_{10}) \approx 1.6 \times 10^{13}$.

3.5.4 Residual Analysis

For an approximate solution $\tilde{\mathbf{x}} \approx \mathbf{x}$ to (3.1) we distinguish between:

- **Residual:** $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$.
- **Error:** $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$.

The condition number relates these quantities:

$$\frac{1}{\kappa(A)} \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} \leq \frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq \kappa(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}. \tag{3.6}$$

The bounds in (3.6) show that:

- *Well-conditioned* systems, small residual implies small error: $\|\mathbf{r}\| \ll \|\mathbf{b}\| \implies \|\mathbf{e}\| \ll \|\mathbf{x}\|$.
- *Ill-conditioned* systems, small residual does not guarantee small error: $\|\mathbf{r}\| \ll \|\mathbf{b}\| \not\Rightarrow \|\mathbf{e}\| \ll \|\mathbf{x}\|$.
- *Condition number* provides both upper and lower bounds on the relative error:

$$\frac{1}{\kappa(A)} \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} \leq \frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq \kappa(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

Chapter 4

Orthogonal Vectors and Subspaces

Orthogonality is one of the most important concepts in numerical linear algebra. It provides both theoretical insight and computational stability, making it essential for developing robust algorithms.

Let V be an inner product space with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$. The notion of orthogonality generalizes the familiar concept of perpendicularity from Euclidean geometry.

Definition 4.1: Orthogonal and Orthonormal Sets

A set of vectors $\{v_1, v_2, \dots, v_n\} \subset V$ is *orthogonal* if

$$\langle v_i, v_j \rangle = 0 \quad \text{for all } i \neq j.$$

The set is *orthonormal* if it is orthogonal and each vector has unit norm:

$$\langle v_i, v_j \rangle = \delta_{ij} \quad \text{for all } i, j \in \{1, 2, \dots, n\},$$

where δ_{ij} is the Kronecker delta.

Orthonormal sets are particularly valuable because they form an ideal basis: coordinates can be computed easily using inner products, and the basis transformation matrix is orthogonal (or unitary), which preserves lengths and angles.

4.1 Gram-Schmidt Process

The Gram-Schmidt process (GS) goal is to construct an orthonormal basis from a given set of linearly independent vectors. GS transforms any linearly independent set of vectors into an orthonormal set spanning the same subspace.

Theorem 4.2: Gram-Schmidt Theorem

Let $\{v_1, v_2, \dots, v_n\}$ be linearly independent vectors in an inner product space V. Then there exists a unique orthonormal set $\{q_1, q_2, \dots, q_n\}$ such that

$$\operatorname{span}\{v_1, \dots, v_k\} = \operatorname{span}\{q_1, \dots, q_k\} \quad \text{for } k = 1, 2, \dots, n.$$

Moreover, each q_k can be written as a linear combination of $v_1, \dots, v_k$.

The construction proceeds iteratively: at each step, we remove the components of the current vector that lie in the span of the previously computed orthonormal vectors, then normalize the result.

The orthonormal vectors are defined recursively:

$$
\begin{aligned}
q_1 &= \frac{v_1}{\|v_1\|}, \\
q_k &= \frac{u_k}{\|u_k\|}, \quad k = 2, 3, \dots, n, \\
u_k &= v_k - \sum_{j=1}^{k-1} \langle v_k, q_j \rangle q_j.
\end{aligned}
$$

The key insight is that u_k represents the component of v_k orthogonal to the subspace spanned by $\{q_1, \dots, q_{k-1}\}$.

Algorithm 3 Gram-Schmidt

Require: Linearly independent vectors $v_1, v_2, \dots, v_n \in \mathbb{R}^m$
Ensure: Orthonormal vectors $q_1, q_2, \dots, q_n \in \mathbb{R}^m$
  $q_1 \leftarrow \frac{v_1}{\|v_1\|}$
  for $k = 2, \dots, n$ **do**
    $u_k \leftarrow v_k$
    for $j = 1, \dots, k-1$ **do**
      $u_k \leftarrow u_k - \langle v_k, q_j \rangle q_j$ ▷ Project out q_j component
    $q_k \leftarrow \frac{u_k}{\|u_k\|}$ ▷ Normalize

Remark 8. Instability of Classical Gram-Schmidt

In the classical algorithm, rounding errors can cause significant loss of orthogonality, especially when the input vectors are nearly linearly dependent. The computed vectors may be far from orthogonal, undermining the algorithm's purpose.

This motivates the modified Gram-Schmidt algorithm, which reorders the computations to improve stability.

4.2 Modified Gram-Schmidt Process

The modified Gram-Schmidt algorithm (MGS) is significantly better numerical stability by performing orthogonalization sequentially against the already computed orthonormal vectors. This reduces the accumulation of rounding errors and better preserves orthogonality in finite precision arithmetic.

Algorithm 4 Modified Gram-Schmidt

Require: Linearly independent vectors $v_1, v_2, \dots, v_n \in \mathbb{R}^m$
Ensure: Orthonormal vectors $q_1, q_2, \dots, q_n \in \mathbb{R}^m$
  for $k = 1, \dots, n$ **do**
    $q_k \leftarrow v_k$
    for $j = 1, \dots, k-1$ **do**
      $r_{jk} \leftarrow \langle q_k, q_j \rangle$ ▷ Compute projection coefficient
      $q_k \leftarrow q_k - r_{jk} q_j$ ▷ Remove q_j component immediately
    $r_{kk} \leftarrow \|q_k\|$
    $q_k \leftarrow \frac{q_k}{r_{kk}}$ ▷ Normalize

The key difference is that each orthogonalization step is performed immediately against the current (partially orthogonalized) vector, rather than against the original input vectors.

Comparison of CGS and MGS The key difference between CGS and MGS lies in *when* the orthogonalized vectors are used:

Classical Gram-Schmidt (CGS)	**Modified Gram-Schmidt (MGS)**
Project v_k against *all* $q_1, \dots, q_{k-1}$ using the *original* v_k	Project v_k *sequentially*, updating after each projection
1: $u_k \leftarrow v_k$ 2: **for** $j = 1, \dots, k-1$ **do** 3: $\quad r_{jk} \leftarrow \langle v_k, q_j \rangle$ 4: $\quad u_k \leftarrow u_k - r_{jk} q_j$ 5: $q_k \leftarrow u_k / \|u_k\|$	1: $q_k \leftarrow v_k$ 2: **for** $j = 1, \dots, k-1$ **do** 3: $\quad r_{jk} \leftarrow \langle q_k, q_j \rangle$ 4: $\quad q_k \leftarrow q_k - r_{jk} q_j$ 5: $q_k \leftarrow q_k / \|q_k\|$
Uses original vector v_k for all inner products Accumulates rounding errors Can lose orthogonality for ill-conditioned matrices	Uses progressively orthogonalized q_k for inner products Reduces error propagation Better preserves orthogonality

Example 9. Numerical example

For $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$, both methods yield (in exact arithmetic):

$$\tilde{Q} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \end{bmatrix}, \qquad R = \begin{bmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} \\ 0 & \frac{\sqrt{6}}{2} \end{bmatrix}.$$

However, in finite precision:

- **CGS**: For nearly dependent columns, $\|Q^\top Q - I\|$ can be $O(\kappa(A)\varepsilon)$ where $\kappa(A)$ is the condition number
- **MGS**: Achieves $\|Q^\top Q - I\| = O(\varepsilon)$, maintaining orthogonality to machine precision

While CGS provides geometric intuition, MGS is strongly preferred for numerical computation. For even better stability, use Householder QR (Section 4.4).

4.3 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

$$A = QR,$$

where $Q \in \mathbb{C}^{m \times m}$ is unitary and $R \in \mathbb{C}^{m \times n}$ is upper triangular (or upper trapezoidal when $m > n$).

When A has full column rank, one commonly uses the *thin* (or *economy*) QR factorization:

$$A = \tilde{Q}R, \qquad \tilde{Q} \in \mathbb{C}^{m \times n},\ \tilde{Q}^H \tilde{Q} = I_n,\ R \in \mathbb{C}^{n \times n} \text{ upper triangular},$$

which is unique up to multiplication of $\tilde{Q}$ on the right by a diagonal unitary matrix (in the real case, by signs).

The QR decomposition has numerous applications:

- Solving least squares problems: $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2$
- Computing matrix eigenvalues (QR algorithm)
- Orthogonalizing vectors (Gram-Schmidt process)
- Numerical solution of linear systems

There are several algorithms for computing the QR decomposition. The Gram-Schmidt process provides geometric insight and is covered in Section **??**. For practical computation, Householder reflections (discussed below) are the most numerically stable and widely used method.

4.4 Householder Reflections

Householder reflections are orthogonal transformations that reflect vectors across hyperplanes. They are widely used in numerical linear algebra for QR decomposition and other orthogonalizations due to their numerical stability and efficiency.

Definition 4.3: Householder reflection

Let $\mathbf{u} \in \mathbb{R}^n$ be a unit vector. The Householder matrix is

$$H = I - 2uu^\top,$$

which represents the reflection across the hyperplane orthogonal to $\mathbf{u}$.

Geometrically, H sends $\mathbf{u} \mapsto -\mathbf{u}$ and fixes every vector orthogonal to $\mathbf{u}$.

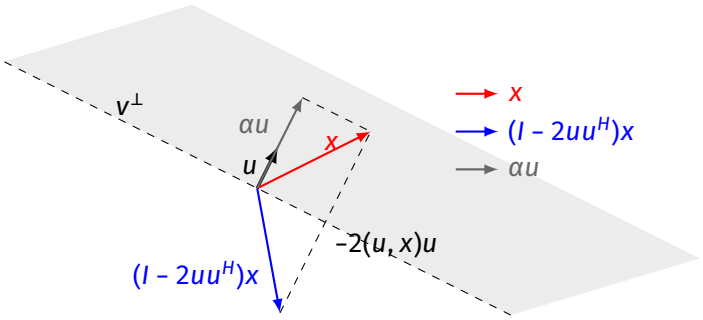


Figure 4.1: Householder reflection of $\mathbf{x}$ across the hyperplane orthogonal to $\mathbf{u}$. The projection $\pi_u(\mathbf{x})$ is in grey and the reflected vector Hx is in blue.

Proposition 1: Basic properties

Let $H = I - 2uu^\top$ with $\|\mathbf{u}\| = 1$. Then

1. $H^\top = H$ (symmetric),
2. $H^\top H = I$ (orthogonal),
3. $H^{-1} = H$ (involutory),
4. $H\mathbf{u} = -\mathbf{u}$ and $H\mathbf{v} = \mathbf{v}$ for all $\mathbf{v} \perp \mathbf{u}$,
5. $\det(H) = -1$.

Proof.

1. is immediate.
2. Expand $(I - 2uu^\top)^2 = I - 4uu^\top + 4u(\mathbf{u}^\top\mathbf{u})\mathbf{u}^\top = I$.
3. Follows from 1. and 2.
4. $H\mathbf{u} = \mathbf{u} - 2\mathbf{u}(\mathbf{u}^\top\mathbf{u}) = \mathbf{u} - 2\mathbf{u} = -\mathbf{u}$. If $\mathbf{v} \perp \mathbf{u}$, then $H\mathbf{v} = \mathbf{v} - 2\mathbf{u}(\mathbf{u}^\top\mathbf{v}) = \mathbf{v}$.
5. $\det(H) = \det(I - 2uu^\top) = \det(I)\det(I - 2u^\top u) = 1 \cdot (1 - 2) = -1$.

□

4.5 Vector Annihilation with Householder Reflections

A key application of Householder reflections is to transform a given vector into a multiple of a standard basis vector, effectively zeroing out all but one component.

Theorem 4.4: Vector annihilation via Householder reflection

For any nonzero $\mathbf{x} \in \mathbb{R}^n$, there exists a Householder matrix $H = I - 2\mathbf{u}\mathbf{u}^\top$ such that

$$H\mathbf{x} = \sigma \mathbf{e}_1, \quad \text{where } \sigma = \pm \|\mathbf{x}\|_2.$$

A numerically stable choice is $\sigma = -\operatorname{sign}(x_1)\,\|\mathbf{x}\|_2$ (with $\operatorname{sign}(0) = 1$). The unit vector $\mathbf{u}$ is constructed as

$$\mathbf{v} = \mathbf{x} - \sigma \mathbf{e}_1, \qquad \mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}.$$

Proof. Let $\mathbf{v} = \mathbf{x} - \sigma\mathbf{e}_1$ and $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|_2$ with $\sigma = \pm\|\mathbf{x}\|_2$. Then

$$H\mathbf{x} = \mathbf{x} - 2\mathbf{u}(\mathbf{u}^\top \mathbf{x}) = \mathbf{x} - \frac{2\mathbf{v}(\mathbf{v}^\top \mathbf{x})}{\|\mathbf{v}\|_2^2}.$$

Computing the inner product:

$$\mathbf{v}^\top \mathbf{x} = (\mathbf{x} - \sigma\mathbf{e}_1)^\top \mathbf{x} = \|\mathbf{x}\|_2^2 - \sigma x_1.$$

Computing the squared norm:

$$\|\mathbf{v}\|_2^2 = (\mathbf{x} - \sigma\mathbf{e}_1)^\top(\mathbf{x} - \sigma\mathbf{e}_1) = \|\mathbf{x}\|_2^2 - 2\sigma x_1 + \sigma^2 = 2(\|\mathbf{x}\|_2^2 - \sigma x_1),$$

where the last equality uses $\sigma^2 = \|\mathbf{x}\|_2^2$. Therefore

$$\frac{2\mathbf{v}(\mathbf{v}^\top \mathbf{x})}{\|\mathbf{v}\|_2^2} = \frac{2\mathbf{v}(\|\mathbf{x}\|_2^2 - \sigma x_1)}{2(\|\mathbf{x}\|_2^2 - \sigma x_1)} = \mathbf{v},$$

and thus $H\mathbf{x} = \mathbf{x} - \mathbf{v} = \sigma\mathbf{e}_1$.
The sign choice $\sigma = -\operatorname{sign}(x_1)\|\mathbf{x}\|_2$ maximizes $|\mathbf{v}|$ and avoids catastrophic cancellation when computing $\mathbf{v} = \mathbf{x} - \sigma\mathbf{e}_1$. If $x_1 > 0$, then $\sigma < 0$, making $\mathbf{v} = \mathbf{x} + |\sigma|\mathbf{e}_1$ an addition rather than a subtraction. □

Algorithm 5 Construct Householder Vector for Annihilation

Require: Nonzero vector $\mathbf{x} \in \mathbb{R}^n$
Ensure: Unit vector $\mathbf{u}$ such that $(I - 2\mathbf{u}\mathbf{u}^\top)\mathbf{x} = \sigma\mathbf{e}_1$, where $\sigma = -\operatorname{sign}(x_1)\|\mathbf{x}\|_2$

$\alpha \leftarrow \|\mathbf{x}\|_2$
$\sigma \leftarrow -\operatorname{sign}(x_1)\,\alpha$ ▷ sign(0) = 1 by convention
$\mathbf{v} \leftarrow \mathbf{x} - \sigma\mathbf{e}_1$
$\mathbf{u} \leftarrow \mathbf{v}/\|\mathbf{v}\|_2$
return $\mathbf{u}$

4.6 QR decomposition via Householder reflections

Theorem 4.5: Householder QR

Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. There exist Householder matrices $H_1, \dots, H_n$ such that

$$R \;=\; H_n \cdots H_2 H_1 A$$

is upper triangular, and with $Q = (H_n \cdots H_1)^\top$ we have $A = QR$.

Algorithmically, process columns $k = 1, \dots, n$: apply a Householder reflection to zero out entries $k+1 : m$ in column k, leaving previously formed zeros undisturbed.

Algorithm 6 QR Decomposition via Householder Reflections (full Q)

Require: $A \in \mathbb{R}^{m\times n}$ with $m \geq n$
Ensure: $Q \in \mathbb{R}^{m\times m}$ (orthogonal), $R \in \mathbb{R}^{m\times n}$ (upper-trapezoidal) s.t. $A = QR$

$Q \leftarrow I_m$
for $k = 1, 2, \dots, n$ **do**
  $\mathbf{x} \leftarrow A_{k:m,\,k}$
  if $\mathbf{x} \neq \mathbf{0}$ **then**
    $\alpha \leftarrow \|\mathbf{x}\|_2$
    $\sigma \leftarrow -\operatorname{sign}(x_1)\,\alpha$ ▷ sign(0) = 1
    $\mathbf{v} \leftarrow \mathbf{x} - \sigma\,\mathbf{e}_1$
    $\beta \leftarrow \|\mathbf{v}\|_2$
    if $\beta > 0$ **then**
      $\mathbf{u} \leftarrow \mathbf{v}/\beta$
      $H_k \leftarrow I_{m-k+1} - 2\,\mathbf{u}\mathbf{u}^\top$ ▷ Reflector
      $A_{k:m,\,k:n} \leftarrow H_k\, A_{k:m,\,k:n}$
      $Q_{k:m,\,:} \leftarrow H_k\, Q_{k:m,\,:}$
$R \leftarrow A$ ▷ $A \implies$ upper-trapezoidal
$Q \leftarrow Q^\top$ ▷ $Q \leftarrow H_n \cdots H_1 \implies Q = H_1 \cdots H_n$

Remark 9. Stability and cost

Householder QR is backward stable; the computed $\hat{Q}$ is orthogonal to machine precision, and $\hat{R}$ is the exact R of a nearby $A + \Delta A$ with small relative $\|\Delta A\|$. The flop count is

$$2mn^2 - \frac{2}{3}n^3 \quad (m \geq n),$$

and blocked implementations use cache-efficient matrix–matrix updates.

Example 10. A 3×2 example

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

Step 1.

$$a_1 = (1, 1, 0)^\top, \quad \|a_1\| = \sqrt{2}, \quad \sigma_1 = -\sqrt{2}$$
$$v_1 = a_1 - \sigma_1 e_1 = (1 + \sqrt{2},\ 1,\ 0)^\top$$
$$u_1 = \frac{v_1}{\|v_1\|}$$

Then

$$H_1 = I - 2u_1u_1^{\mathsf{T}} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad A^{(1)} = H_1A = \begin{bmatrix} -\sqrt{2} & -\frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 \end{bmatrix}.$$

Step 2. Take the subvector $\mathbf{x} = \left(-\frac{1}{\sqrt{2}}, 1\right)^{\mathsf{T}}$, $\|\mathbf{x}\| = \sqrt{\frac{3}{2}} = \frac{\sqrt{6}}{2}$, $\sigma_2 = \frac{\sqrt{6}}{2}$. Build a 2×2 reflector and embed:

$$H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} \\ 0 & \frac{\sqrt{2}}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix}.$$

Then

$$R = H_2A^{(1)} = \begin{bmatrix} -\sqrt{2} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{\sqrt{6}}{2} \\ 0 & 0 \end{bmatrix}, \qquad Q = H_1H_2.$$

The thin factor is

$$\tilde{Q} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \end{bmatrix}, \quad \text{so } A = \tilde{Q}R.$$

Part II

Projection Methods for Linear Systems

Chapter 5

Projections

Projection operators provide a mathematical framework for decomposing vectors into components along different subspaces, which is essential for understanding least squares problems, orthogonal decompositions, and many iterative methods.

5.1 Notation and Setup

Linear system. Solve $A\mathbf{x} = \mathbf{b}$ with $A \in \mathbb{R}^{n\times n}$, $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. The exact solution is $\mathbf{x}^* = A^{-1}\mathbf{b}$ (when A is nonsingular).

Initial guess. $\mathbf{x}_0 \in \mathbb{R}^n$.

Residuals. Given $\mathbf{x}_0$, the current residual is $\mathbf{r} = \mathbf{b} - A\mathbf{x}$, and the initial residual is $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$.

Subspaces.
- *Search space* $\mathcal{K} \subset \mathbb{R}^n$ of dimension m.
- *Constraint (left) space* $\mathcal{L} \subset \mathbb{R}^n$ of dimension m.

Bases / matrices.
- $V = [\mathbf{v}_1, \dots, \mathbf{v}_m] \in \mathbb{R}^{n\times m}$ with $\operatorname{range}(V) = \mathcal{K}$.
- $W = [\mathbf{w}_1, \dots, \mathbf{w}_m] \in \mathbb{R}^{n\times m}$ with $\operatorname{range}(W) = \mathcal{L}$.
- Unless stated otherwise, V and W have full column rank.

Trial solution. $\mathbf{x} = \mathbf{x}_0 + V\mathbf{y}$ with coefficients $y \in \mathbb{R}^m$.

Petrov–Galerkin condition. The residual is orthogonal to $\mathcal{L}$:

$$\mathbf{r} \perp \mathcal{L} \quad \Longleftrightarrow \quad W^\top(\mathbf{b} - A(\mathbf{x}_0 + V\mathbf{y})) = 0.$$

This yields the reduced $(m \times m)$ system

$$H\mathbf{y} = \mathbf{g}, \qquad H := W^\top AV, \quad \mathbf{g} := W^\top \mathbf{r}_0,$$

and the update $\mathbf{x} = \mathbf{x}_0 + V\mathbf{y}$. The matrix H must be nonsingular for y to be uniquely defined.

Orthogonal projections. $\mathcal{L} = \mathcal{K} \Rightarrow$ condition $V^T\mathbf{r} = 0$.

Oblique projections. $\mathcal{L} \neq \mathcal{K}$ (e.g. $\mathcal{L} = A\mathcal{K}$) $\Rightarrow$ condition $W^\top\mathbf{r} = 0$ with a different left basis.

Projectors (Euclidean). For a full-rank basis $B \in \mathbb{R}^{n\times m}$,

$$P_{\operatorname{range}(B)} := B(B^TB)^{-1}B^\top,$$

and if B has orthonormal columns, $P_{\operatorname{range}(B)} = BB^\top$.

***A*-inner product.** If A is symmetric positive definite (SPD), define the A-inner product $(\mathbf{u}, \mathbf{v})_A := \mathbf{u}^\top A\mathbf{v}$ and the energy norm $\|\mathbf{u}\|_A := \sqrt{\mathbf{u}^\top A\mathbf{u}}$.

Special choices.
- $\mathcal{L} = \mathcal{K}$ (Galerkin, SPD A): best approximation in $\|\cdot\|_A$.
- $\mathcal{L} = A\mathcal{K}$ (residual minimization): best approximation in residual 2-norm.

5.2 Projection Operator

A projection finds the *closest point/shadow* of a vector onto a subspace. This operation decomposes any vector into two parts: one lying within the target subspace and another orthogonal to it.

Definition 5.1: Projection Operator

A linear operator $P : V \to V$ on an inner product space V is called a *projection* if it is idempotent:

$$P^2 = P$$

Corollary 2: (Orthogonal Projection)**.** The operator P is called an *orthogonal projection* if it is additionally *self-adjoint/Hermitian*:

$$P^H = P$$

The idempotent property captures the essential characteristic of projections: applying the projection twice gives the same result as applying it once. Geometrically, once a vector is projected onto a subspace, further projections leave it unchanged.

5.2.1 Properties of Projections

Let P be a projection operator on an inner product space V. Then:

1. $\text{Range}(P) = \{v \in V : P\mathbf{v} = v\}$ (the range consists of fixed points)
2. $V = \text{Range}(P) \oplus \text{Null}(P)$ (direct sum decomposition)
3. If P is an orthogonal projection, then $\text{Range}(P) \perp \text{Null}(P)$
4. The eigenvalues of any projection are 0 and 1

Proof.

1. If $\mathbf{v} \in \text{Range}(P)$, then $\mathbf{v} = Pu$ for some u, so $P\mathbf{v} = P^2u = Pu = v$. Conversely, if $P\mathbf{v} = v$, then v is clearly in the range of P.
2. For any $\mathbf{v} \in V$, write $\mathbf{v} = P\mathbf{v} + (\mathbf{v} - P\mathbf{v})$. Since $P\mathbf{v} \in \text{Range}(P)$ and $P(\mathbf{v} - P\mathbf{v}) = P\mathbf{v} - P^2\mathbf{v} = P\mathbf{v} - P\mathbf{v} = 0$, we have $\mathbf{v} - P\mathbf{v} \in \text{Null}(P)$.
3. For orthogonal projections, if $u \in \text{Range}(P)$ and $v \in \text{Null}(P)$, then $\mathbf{u} = P\mathbf{w}$ for some w, and $\langle u, \mathbf{v}\rangle = \langle P\mathbf{w}, \mathbf{v}\rangle = \langle \mathbf{w}, P^*\mathbf{v}\rangle = \langle \mathbf{w}, P\mathbf{v}\rangle = \langle \mathbf{w}, 0\rangle = 0$.
4. The eigenvalues of any projection are 0 and 1, since $P^2 = P$ implies the minimal polynomial divides $\mathbf{x}(\mathbf{x} - 1)$, so the only possible eigenvalues are 0 and 1.

□

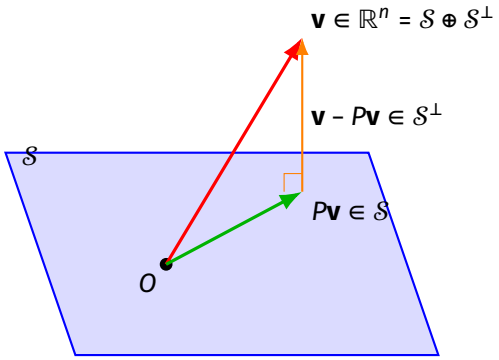
5.3 Reduced Systems

In iterative methods for solving linear systems, we often work with reduced systems H, that capture the essential features of the original problem (3.1) within a lower-dimensional subspace.

Proposition 2: Galerkin with SPD A

If $A = A^\top > 0$ and $\mathcal{L} = \mathcal{K}$, then for any full-rank bases V, W with $\text{range}(V) = \text{range}(W) = \mathcal{K}$, the matrix $H = W^\top AV$ is symmetric positive definite and hence invertible.

Proof sketch. Write $W = VG$ with $G \in \mathbb{R}^{m\times m}$ invertible. Then $H = G^\top V^\top AV$. Since $A = C^\top C$ and V has full column rank, $V^\top AV = (CV)^\top (CV) > 0$. □



Orthogonal projection: $\mathbf{v} = P\mathbf{v} + (\mathbf{v} - P\mathbf{v})$
where $P\mathbf{v} \in \mathcal{S}$ and $(\mathbf{v} - P\mathbf{v}) \perp \mathcal{S}$

Figure 5.1: Geometric interpretation of orthogonal projection. The vector v is decomposed into its projection $P\mathbf{v}$ onto the subspace S and the orthogonal component $\mathbf{v} - P\mathbf{v}$.

Proposition 3: Petrov–Galerkin with $\mathcal{L} = A\mathcal{K}$

Suppose A is invertible and $\mathcal{L} = A\mathcal{K}$. For full-rank V with range$(V) = \mathcal{K}$, choose $W = AVG$ with G invertible. Then $H = W^\top AV = G^\top (AV)^\top (AV) > 0$ and is invertible.

These conditions underpin the optimality results: Galerkin ($\mathcal{L} = \mathcal{K}$) yields best approximation in the A-norm when $A > 0$, while $\mathcal{L} = A\mathcal{K}$ yields residual 2-norm minimization.

5.4 Matrix Representation of Projections

In finite-dimensional spaces, projections can be represented as matrices with specific structural properties.

Theorem 5.2: Matrix Characterization of Orthogonal Projections

A matrix $P \in \mathbb{R}^{n \times n}$ represents an orthogonal projection if and only if:

1. $P^2 = P$ (idempotent)
2. $P^\top = P$ (symmetric)

In this case, P projects onto Col(P) along Null(P).

The geometric interpretation is crucial: P maps every vector to its closest point in the column space of P, measured in the Euclidean norm.

Example 11. Simple Projection Examples

Projection onto a line: Let $u \in \mathbb{R}^n$ be a unit vector. The projection onto the line spanned by u is:

$$P_u = uu^\top$$

Projection onto coordinate subspace: The projection onto the first k coordinates is:

$$P_k = \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix}$$

where I_k is the $k \times k$ identity matrix.

5.5 Oblique Projections

Not all useful projections are orthogonal. An *oblique* projection maps onto a target subspace along a (non-orthogonal) complementary subspace.

Definition 5.3: Oblique Projection

Let $\mathcal{S}, \mathcal{T} \subset \mathbb{R}^n$ be subspaces with a direct sum decomposition

$$\mathbb{R}^n = \mathcal{S} \oplus \mathcal{T}$$
$$\mathcal{S} \cap \mathcal{T} = \{0\}$$

The *oblique projection* P onto $\mathcal{S}$ along $\mathcal{T}$ is the unique linear map satisfying $\text{range}(P) = \mathcal{S}$ and $\text{null}(P) = \mathcal{T}$; equivalently,

$$P\mathbf{v} \in \mathcal{S}, \qquad \mathbf{v} - P\mathbf{v} \in \mathcal{T}, \qquad \forall\, \mathbf{v} \in \mathbb{R}^n$$

Matrix realizations: Let $S \in \mathbb{R}^{n\times k}$ have full column rank with $\text{range}(S) = \mathcal{S}$.

- If $W \in \mathbb{R}^{n\times k}$ has full column rank with $\ker(W^\top) = \mathcal{T}$ (i.e., $\text{range}(W) = \mathcal{T}^\perp$) and $W^\top S$ is nonsingular, then
$$P = S(W^\top S)^{-1} W^\top$$
This realizes the projector *onto* $\mathcal{S}$ *along* $\mathcal{T}$. (Note: using $T^\top$ in place of $W^\top$ would project along $\mathcal{T}^\perp$, not along $\mathcal{T}$.)
- If $T \in \mathbb{R}^{n\times(n-k)}$ spans $\mathcal{T}$ and the block $[S\ T]$ is invertible, then
$$P = [ST] \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} [ST]^{-1}.$$

In general P is not symmetric ($P^\top \neq P$) but is always idempotent ($P^2 = P$).

Example 12. Oblique projection in iterative methods

In iterative solvers for $A\mathbf{x} = \mathbf{b}$, choose a *search space* $\mathcal{K} = \text{span}(V)$ and a *test space* $\text{span}(W)$; the Petrov–Galerkin condition $W^\top \mathbf{r} = 0$ produces

$$P = V(W^\top V)^{-1} W^\top,$$

which is the oblique projector onto $\mathcal{K}$ along $\ker(W^\top)$.
With W chosen so that $\text{span}(W) = A\mathcal{K}$ (as in GMRES), the residual is enforced orthogonal to $A\mathcal{K}$, i.e., the projection is along $(A\mathcal{K})^\perp$.

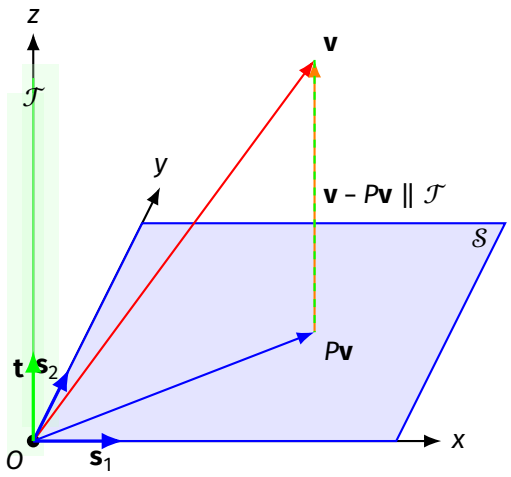


Figure 5.2: Oblique projection: $P\mathbf{v} \in \mathcal{S}$, $\mathbf{v} - P\mathbf{v} \in \mathcal{T}$ (not necessarily orthogonal). Basis vectors $\mathbf{s} \in \mathcal{S}$ and $\mathbf{t} \in \mathcal{T}$ are shown.

Chapter 6

Krylov Subspaces

Krylov subspaces capture the span of repeated applications of a matrix to a vector and are the foundation of projection methods driven by matrix–vector products.

Definition 6.1: Krylov subspace

For $A \in \mathbb{R}^{n\times n}$ and a nonzero $\mathbf{v} \in \mathbb{R}^n$, the m-th Krylov subspace is

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

Always $\dim \mathcal{K}_m \le \min\{m, n\}$.

6.1 Properties of Krylov Subspaces

6.1.1 Minimal polynomial and grade

The *grade* μ of $\mathbf{v}$ with respect to A is the degree of the monic polynomial p of least degree such that $p(A)\mathbf{v} = 0$. Then

$$A\mathcal{K}_\mu(A, \mathbf{v}) = \mathcal{K}_\mu(A, \mathbf{v}), \qquad \mathcal{K}_m(A, \mathbf{v}) = \mathcal{K}_\mu(A, \mathbf{v}) \text{ for all } m \ge \mu.$$

Thus Krylov spaces stabilize once an A-invariant subspace is reached (`happy breakdown`).

Cayley–Hamilton For any $\mathbf{x} \in \mathcal{K}_m(A, \mathbf{v})$ with $m \ge \mu$, there exists a polynomial $q_{\mu-1}$ of degree at most $\mu - 1$ such that

$$\mathbf{x} = q_{\mu-1}(A)\,\mathbf{v}.$$

Indeed, dividing any polynomial representative by the minimal polynomial gives $q = q_1\,p + q_{\mu-1}$ and $q(A)\mathbf{v} = q_{\mu-1}(A)\mathbf{v}$.

Dimension and nesting. The dimensions satisfy

$$\dim \mathcal{K}_m(A, \mathbf{v}) \le m, \qquad \mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \cdots \subseteq \mathcal{K}_\mu = \cdots.$$

Projection viewpoint. Given an initial guess $\mathbf{x}_0$ for $A\mathbf{x} = \mathbf{b}$ and residual $\mathbf{r}_0$, Krylov methods search in $\mathbf{x}_0 + \mathcal{K}_m(A, \mathbf{r}_0)$ while enforcing a Petrov–Galerkin condition on the residual. Choices of the test space $\mathcal{L}$ produce FOM ($\mathcal{L} = \mathcal{K}_m$) and GMRES ($\mathcal{L} = A\mathcal{K}_m$); see Chapters **??** and **??**.

6.2 Arnoldi Iteration

The *Arnoldi iteration* is a *Krylov subspace iterative method* that reduces A to an **upper Hessenberg matrix** H_m. We can then use this simple representation of A to approximate some **eigenvalues** of A.

6.2.1 Derivation of Arnoldi Iteration

Let $A \in \mathbb{C}^{n\times n}$. We want to compute $A = VHV^*$ where H is upper Hessenberg and V is unitary ($V^*V = I$).

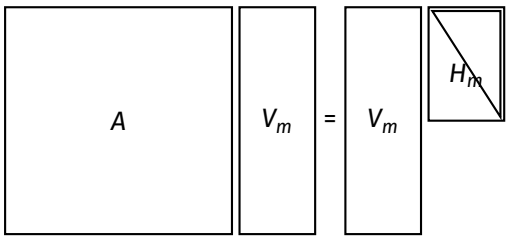
But why do we want to compute $A = VHV^*$? The reason is that such a *unitary similarity* preserves the eigenvalues of A, while making the matrix H much simpler in structure (upper Hessenberg). Indeed, since $H = V^*AV$, if $H\mathbf{x} = \lambda\mathbf{x}$, then

$$A(V\mathbf{x}) = VH\mathbf{x} = \lambda(V\mathbf{x}),$$

so A and H share the same eigenvalues.

But what if A is very large? For $n \gg 1$, computing the full factorization is too expensive and unnecessary. Instead, we work with a smaller subspace of dimension $m \ll n$.

$$AV_m \approx V_m H_m,$$



Consider the sequence of vectors generated by repeatedly applying a matrix A to an initial vector $\mathbf{r}_0$:

$$\mathbf{r}_0, \quad A\mathbf{r}_0, \quad A^2\mathbf{r}_0, \quad A^3\mathbf{r}_0, \quad \dots$$

The *Krylov subspace* of dimension $m + 1$ is the span of the first $m + 1$ such vectors:

$$\mathcal{K}_{m+1}(A, \mathbf{r}_0) := \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^m\mathbf{r}_0\}$$

While this sequence captures how A acts on $\mathbf{r}_0$, these vectors typically become increasingly aligned and numerically dependent. The Arnoldi process transforms this unwieldy sequence into an orthonormal basis that preserves all the essential information about A's action on the Krylov subspace.

6.2.2 The Arnoldi Algorithm

The key insight of Arnoldi iteration is to apply the Gram-Schmidt process *incrementally* as we build up the Krylov subspace. Starting with $\mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|_2$, we:

Algorithm 7 Arnoldi Iteration (Modified Gram-Schmidt)

Require: $A \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{x}_0$ (init. guess), m (num. steps)
Ensure: $V_{m+1} = [\mathbf{v}_1, \dots, \mathbf{v}_{m+1}]$ (Orth. basis of $\mathcal{K}_{m+1}$, $n \times m$), $\overline{H}_m$ (Upper Hessenberg, $(m+1)\times m$)
  $\mathbf{r}_0 \leftarrow \mathbf{b} - A\mathbf{x}_0$, $\beta \leftarrow \|\mathbf{r}_0\|_2$, $\mathbf{v}_1 \leftarrow \mathbf{r}_0/\beta$
  for $j = 1, 2, \dots, m$ **do**
    $\mathbf{w}_j \leftarrow A\mathbf{v}_j$
    for $i = 1, 2, \dots, j$ **do**
      $h_{i,j} \leftarrow \langle \mathbf{v}_i, \mathbf{w}_j \rangle$
      $\mathbf{w}_j \leftarrow \mathbf{w}_j - h_{i,j}\mathbf{v}_i$
    $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|_2$
    if $h_{j+1,j} = 0$ **then**
      Break
    $\mathbf{v}_{j+1} \leftarrow \mathbf{w}_j / h_{j+1,j}$

After m steps, we have constructed:

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

The matrix $\overline{H}_m$ is *upper Hessenberg* (zero below the first subdiagonal) and encodes all the orthogonalization coefficients from the Gram-Schmidt process.

6.2.3 Arnoldi Relation

The fundamental relationship is:

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

This compact relation captures a profound fact: *the action of the large matrix A on the Krylov subspace is completely characterized by the small Hessenberg matrix* $\overline{H}_m$. Column by column, this says:

$$A\mathbf{v}_j = h_{1,j}\mathbf{v}_1 + h_{2,j}\mathbf{v}_2 + \cdots + h_{j,j}\mathbf{v}_j + h_{j+1,j}\mathbf{v}_{j+1}$$

In other words, $A\mathbf{v}_j$ lies in $\text{span}\{\mathbf{v}_1, \dots, \mathbf{v}_{j+1}\}$, which is exactly what we'd expect since $A\mathbf{v}_j$ is the $(j+1)$-th Krylov vector (before orthogonalization).

6.2.4 Breakdown Conditions

If $h_{j+1,j} = 0$ for some $j < m$, then $\mathbf{w} = A\mathbf{v}_j$ lies entirely in $\text{span}\{\mathbf{v}_1, \dots, \mathbf{v}_j\}$. This means:

$$A(\mathcal{K}_j(A, \mathbf{r}_0)) \subseteq \mathcal{K}_j(A, \mathbf{r}_0)$$

The Krylov subspace is *A-invariant*, and we have found an exact invariant subspace. This is called "happy breakdown" because:

- We can solve linear systems exactly within this subspace
- We have found exact eigenvalue/eigenvector information
- No further iteration is needed

Near-breakdown. In finite precision arithmetic, $h_{j+1,j}$ may be very small but nonzero, leading to numerical instability. This requires careful handling through techniques like deflation or restarting.

6.2.5 Numerical Stability and Reorthogonalization

The Modified Gram-Schmidt process used in Algorithm 7 is more numerically stable than Classical Gram-Schmidt, but orthogonality can still be lost due to:

- Round-off errors accumulating over many iterations
- Nearly linearly dependent Krylov vectors
- Ill-conditioned matrices A

6.2.6 Computational Complexity

Per iteration cost:

- One matrix-vector product: $A\mathbf{v}_j$ costs $O(\text{nnz}(A))$ or $O(n^2)$ flops
- Orthogonalization: j inner products and j vector updates cost $O(jn)$ flops

Total cost for m iterations:

$$\text{Flops} = O\left(m \cdot \text{cost}(A\mathbf{v}) + \sum_{j=1}^{m} jn\right) = O(m \cdot \text{cost}(A\mathbf{v}) + m^2 n)$$

Storage requirements:

- Basis vectors V_{m+1}: $O(nm)$ memory
- Hessenberg matrix $\overline{H}_m$: $O(m^2)$ memory
- **Total**: $O(nm + m^2)$ memory

The storage requirement $O(nm)$ can become prohibitive for large m, motivating restarted variants.

6.2.7 Applications: The Foundation for Krylov Solvers

The Arnoldi relation (**??**) enables two major classes of iterative solvers:

Full Orthogonalization Method (FOM): Uses Galerkin projection: find $\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m(A, \mathbf{r}_0)$ such that the residual is orthogonal to the Krylov subspace.

Generalized Minimal Residual (GMRES): Uses minimal residual projection: find $\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m(A, \mathbf{r}_0)$ that minimizes $\|A\mathbf{x}_m - \mathbf{b}\|_2$.

Both methods reduce the original $n \times n$ problem to an $m \times m$ problem involving H_m or $\overline{H}_m$.

Summary 1: The Power of Arnoldi

The Arnoldi iteration transforms the numerically unstable sequence $\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots\}$ into:

- A numerically stable orthonormal basis V_{m+1} of $\mathcal{K}_{m+1}(A, \mathbf{r}_0)$
- A compact representation $\overline{H}_m$ of how A acts on the Krylov subspace

- The fundamental relation $AV_m = V_{m+1}\overline{H}_m$ that enables efficient projections
- A foundation for optimal Krylov subspace methods like GMRES
- Natural stopping criteria through happy breakdown detection

This combination of numerical stability, dimensional reduction, and preserved spectral information makes Arnoldi iteration one of the most important algorithms in numerical linear algebra.

The specific applications of this framework to solve linear systems and eigenvalue problems are developed in Chapter **??**, where we'll see how the Arnoldi relation enables both exact solutions (via FOM) and optimal approximations (via GMRES).

6.3 Lanczos Iteration

When $A = A^\top$ is symmetric, the Arnoldi process simplifies to the *Lanczos iteration*, which produces a tridiagonal matrix T_m instead of a Hessenberg matrix $\overline{H}_m$.

6.3.1 Derivation of Lanczos Iteration

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

Then we have the Arnoldi relation

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

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:

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

We solve the tridiagonal system:

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

using **LU-factorization**:

$$T_m = L_m U_m$$

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

Algorithm 8 Lanczos Iteration (Arnoldi for symmetric $A = A^\top$)

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

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.

Part III

Iterative Solvers for Linear Systems

Chapter 7

Steepest Descent

7.1 Notation

We wish to solve the linear system

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

7.1.1 Quadratic form

The quadratic form is just a scalar, quadratic function $f : \mathbb{R}^n \to \mathbb{R}$ that maps a vector $\mathbf{x}$ to a scalar:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top\mathbf{x} + c \tag{7.1}$$

where c is just a constant.

Gradient Now we want to observe how the function f changes as we change $\mathbf{x}$, i.e. we want to compute the gradient $\nabla f(\mathbf{x})$.

$$\begin{aligned}\nabla f(\mathbf{x}) &= \begin{bmatrix}\frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_n} f(\mathbf{x})\end{bmatrix} \\ &= \begin{bmatrix}\frac{1}{2}(2a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n) - b_1 \\ \vdots \\ \frac{1}{2}(a_{n1}x_1 + a_{n2}x_2 + \ldots + 2a_{nn}x_n) - b_n\end{bmatrix} \\ &= \frac{1}{2}(A + A^\top)\mathbf{x} - \mathbf{b}\end{aligned} \tag{7.2}$$

Now, if we assume that A is symmetric ($A = A^\top$), then (7.2) simplifies to

$$\nabla f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$$

Now, under the assumption that A also is positive definite ($A > 0$), the quadratic form (7.1) is strictly convex and has a unique minimizer $\mathbf{x}_\star$.

$$\nabla f(\mathbf{x}_\star) = A\mathbf{x}_\star - \mathbf{b} = 0 \implies A\mathbf{x}_\star = \mathbf{b}$$

Case 1: Suppose A is *SPD*, then the quadratic form (7.1) is strictly convex and has a unique minimizer $\mathbf{x}_\star$ which is equivalent to the solution of the linear system (3.1).

Case 2: Suppose A is *symmetric* (be it positive definite or not). Let $\mathbf{x}$ be a point that satisfies (3.1) and minimizes the quadratic form (7.1). Now we introduce the error term $\mathbf{e}$ s.t.

$$\begin{aligned} f(\mathbf{x}+\mathbf{e}) &= \frac{1}{2}(\mathbf{x}+\mathbf{e})^\top A(\mathbf{x}+\mathbf{e}) - \mathbf{b}^\top(\mathbf{x}+\mathbf{e}) + c \\ &= \overbrace{\frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top\mathbf{x} + c}^{f(\mathbf{x})} + \frac{1}{2}\mathbf{e}^\top A\mathbf{e} + \mathbf{e}^\top \overbrace{A\mathbf{x}}^{\mathbf{b}} - \mathbf{b}^\top\mathbf{e} \\ &= f(\mathbf{x}) + \frac{1}{2}\mathbf{e}^\top A\mathbf{e} + \mathbf{e}^\top\mathbf{b} - \mathbf{b}^\top\mathbf{e} \\ &= f(\mathbf{x}) + \frac{1}{2}\mathbf{e}^\top A\mathbf{e} \end{aligned}$$

If A is positive definite, then $\mathbf{e}^\top A\mathbf{e} > 0$ for all $\mathbf{e} \neq 0$, and therefore $\mathbf{x}$ is the unique minimizer of f.

Now if A is not SPD, then the equation (7.2) hints that we try to find the solution of the system $\frac{1}{2}(A + A^\top)\mathbf{x} = \mathbf{b}$ instead, which we will discuss later.

Throughout the first section, we will use the following 2×2 SPD example to illustrate and explain the concepts and methods we introduce.

Example 13. SPD problem

$$A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 2 \\ -8 \end{bmatrix}, \quad c = 0.$$

The corresponding quadratic form is

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top\mathbf{x} = \frac{1}{2}(3x_1^2 + 2x_1x_2 + 6x_2^2) - (2x_1 - 8x_2).$$

The solution/minimizer is

$$\mathbf{x}_\star = \begin{bmatrix} 2 \\ -2 \end{bmatrix}.$$

If we plot the corresponding quadratic form of this problem, we get the following surface plot:

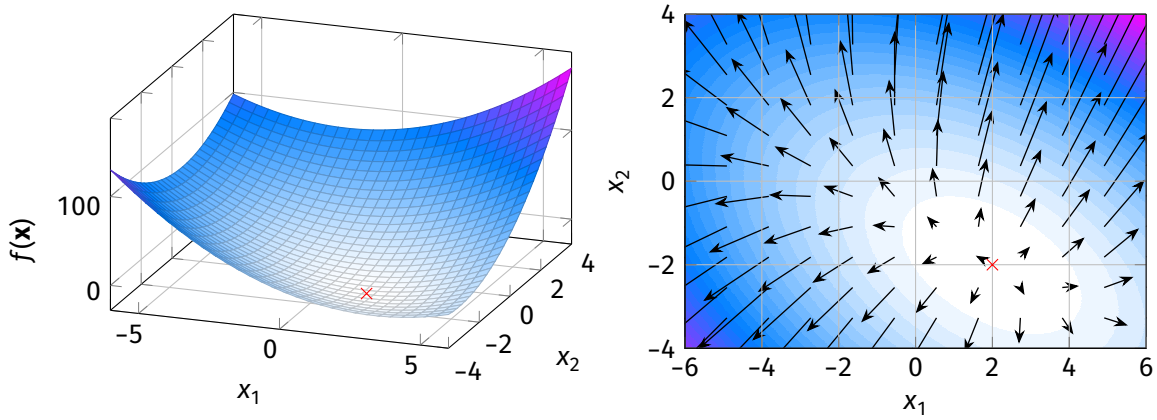


Figure 7.1: **(a)** We see the surface plot of the quadratic form f curving upwards, indicating that it is convex and therefore has a minimum. **(b)** Same plot seen from above, with gradient vectors added. The gradient vectors point in the direction of steepest ascent, and we see that they point away from the minimizer (red cross).

Let A be SPD, with a given initial guess $\mathbf{x}_0$, and residual $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$.

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

Let $B \in \mathbb{R}^{n \times n}$ be SPD, and using Kantorovich's inequality 2.9 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 $B = Q^T \Lambda Q$. Choose $\|\mathbf{x}\|_2 = 1$ where $\|Q\mathbf{x}\|_2 = \|\mathbf{x}\|_2 = 1$. Then:

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

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

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

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

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

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

$$\frac{\|\mathbf{r}_k\|_2^4}{\|\mathbf{r}_k\|_A^2 \|\mathbf{r}_k\|_{A^{-1}}^2} \geq \frac{4\lambda_1\lambda_n}{(\lambda_1 + \lambda_n)^2}$$

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

Example 14. Discrete Laplacian in 2D

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

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

Example 15

If A is SPD, with $\mathcal{L} = \mathcal{K} = \text{span}\{\mathbf{r}_k\}$, then:

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{r}_k, \quad \alpha_k \in \mathbb{R} \\
\mathbf{r}_{k+1} &= \mathbf{b} - A\mathbf{x}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{r}_k \\
\mathbf{r}_{k+1} \perp \mathbf{r}_k &\Rightarrow \mathbf{r}_k^\top(\mathbf{r}_k - \alpha_k A\mathbf{r}_k) = 0 \quad \Rightarrow \alpha_k = \frac{\mathbf{r}_k^\top \mathbf{r}_k}{\mathbf{r}_k^\top A\mathbf{r}_k} \\
\mathbf{d}_k &= \mathbf{x}_\star - \mathbf{x}_k \\
\mathbf{r}_k &= \mathbf{b} - A\mathbf{x}_k = A\mathbf{x}_\star - A\mathbf{x}_k = A\mathbf{d}_k
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{r}_{k+1} &= \mathbf{b} - A\mathbf{x}_{k+1} = A(\mathbf{x}_\star - \mathbf{x}_{k+1}) = A\mathbf{d}_{k+1} = A\mathbf{d}_k - \alpha_k A\mathbf{r}_k \\
\mathbf{d}_{k+1} &= \mathbf{d}_{k+1}^\top A\mathbf{d}_{k+1} = \mathbf{d}_{k+1}^\top \mathbf{r}_{k+1} \\
&= (\mathbf{d}_k - \alpha_k \mathbf{r}_k)^\top \mathbf{r}_{k+1} = \mathbf{d}_k^\top \mathbf{r}_{k+1} \\
&= \mathbf{d}_k^\top(\mathbf{r}_k - \alpha_k A\mathbf{r}_k) = \mathbf{d}_k^\top \mathbf{r}_k - \alpha_k \mathbf{d}_k^\top A\mathbf{r}_k \\
&= \mathbf{d}_k^\top A\mathbf{d}_k - \alpha_k \mathbf{r}_k^\top \mathbf{r}_k \\
&= \|\mathbf{d}_k\|_A^2 - \alpha_k \|\mathbf{r}_k\|^2 \\
&= \|\mathbf{d}_k\|_A^2 - \frac{\|\mathbf{r}_k\|^4}{\|\mathbf{r}_k\|_A^2} \\
\|\mathbf{d}_{k+1}\|_A^2 &= \|\mathbf{d}_k\|_A^2 \left(1 - \frac{\|\mathbf{r}_k\|^4}{\|\mathbf{r}_k\|_A^2 \|\mathbf{r}_k\|_{A^{-1}}^2}\right)
\end{aligned}$$

Chapter 8

Conjugate Gradient

Let $A \in \mathbb{R}^{n \times n}$ be symmetric positive definite (SPD). We want to solve

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

We use the Galerkin condition with

$$\mathcal{K}_m = \mathcal{L}_m = \mathcal{K}_m(A, \mathbf{r}_0), \quad \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0.$$

Then

$$\mathbf{x}_m = \mathbf{x}_0 + V_m \left(V_m^\top A V_m\right)^{-1} V_m^\top \mathbf{r}_0,$$

Proposition 4

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

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

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

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 9 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 $\mathbf{x}_m$

Convergence of CG

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:

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

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 $p(0) = 1$.

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.

$$\langle f, g\rangle = \int_a^b w(x) f(x) g(x)\, dx, \quad w(x) > 0 \text{ (weight function)}$$

$$p_0(x) = 1$$

$$p_1(x) = x$$

$$p_n(x) = (x - a_n) p_{n-1}(x) - b_n p_{n-2}(x), \quad n \geq 2$$

$$a_n = \frac{\langle x p_{n-1}, p_{n-1}\rangle}{\langle p_{n-1}, p_{n-1}\rangle}$$

$$b_n = \frac{\langle x p_{n-2}, p_{n-2}\rangle}{\langle p_{n-2}, p_{n-2}\rangle}$$

Chapter 9

Full Orthogonalization Method (FOM)

9.1 Overview and intuition

FOM (Full Orthogonalization Method) is a Krylov-subspace method for solving the linear system

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

where we build successive approximations $\mathbf{x}_m \in \mathbf{x}_0 + \mathcal{K}_m(A, \mathbf{r}_0)$ with the initial residual $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$. The characteristic feature of FOM is the Galerkin condition with equal trial and test spaces $\mathcal{L}_m = \mathcal{K}_m$: the residual $\mathbf{r}_m$ is orthogonal to the entire Krylov subspace $\mathcal{K}_m$. Intuitively, we insist that the update to the approximate solution has removed all components of the residual that lie in the directions in which we seek corrections.

While GMRES enforces a residual-minimization property (and can be viewed as a Petrov–Galerkin method with test space $A\mathcal{K}_m$), FOM enforces the Galerkin property directly on $\mathcal{K}_m$. This difference has important consequences for both theory and practice, which we discuss in later sections.

9.2 Recap of the Arnoldi decomposition

To work with Krylov subspaces numerically we use the Arnoldi algorithm to generate an orthonormal basis $V_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$ of $\mathcal{K}_m(A, \mathbf{r}_0)$ and the corresponding upper Hessenberg matrix $\overline{H}_m \in \mathbb{R}^{(m+1)\times m}$ that satisfies

$$AV_m = V_{m+1}\overline{H}_m = V_m H_m + h_{m+1,m}\,\mathbf{v}_{m+1}\mathbf{e}_m^\top, \tag{9.1}$$

where $H_m = \overline{H}_m(1:m, 1:m) = V_m^\top A V_m$ is the projection of A onto the subspace spanned by the columns of V_m.

9.3 Galerkin formulation and the small projected system

Express the FOM approximation in the form

$$\mathbf{x}_m = \mathbf{x}_0 + V_m\mathbf{y}_m,$$

and impose the Galerkin condition $V_m^\top\mathbf{r}_m = 0$, where $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$. Using the Arnoldi relation (9.1) we have

$$V_m^\top\mathbf{r}_m = V_m^\top\mathbf{r}_0 - V_m^\top A V_m\mathbf{y}_m = V_m^\top\mathbf{r}_0 - H_m\mathbf{y}_m = 0.$$

Since $V_m^\top \mathbf{r}_0 = \beta \mathbf{e}_1$, $\beta = \|\mathbf{r}_0\|_2$, this leads to the small projected linear system

$$H_m \mathbf{y}_m = \beta \mathbf{e}_1. \tag{9.2}$$

The FOM iterate follows from $\mathbf{x}_m = \mathbf{x}_0 + V_m \mathbf{y}_m$.

9.4 Residual formula and cost of evaluation

The Arnoldi relation gives an inexpensive expression for the residual:

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

because $H_m \mathbf{y}_m = \beta \mathbf{e}_1$ by (9.2). Thus the residual is a scalar multiple of the last Arnoldi vector $\mathbf{v}_{m+1}$, and

$$\|\mathbf{r}_m\|_2 = |h_{m+1,m}|\, |\mathbf{e}_m^\top \mathbf{y}_m|. \tag{9.3}$$

We can therefore monitor convergence cheaply, by tracking the (scalar) quantities $h_{m+1,m}$ and the last component of $\mathbf{y}_m$.

9.5 FOM algorithm

The practical method performs Arnoldi incrementally and solves the small system $H_j \mathbf{y}_j = \beta \mathbf{e}_1$ at each step to form the current approximation. The algorithm below summarizes the approach.

Algorithm 10 Full Orthogonalization Method (FOM)

Require: $A \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{x}_0$, $m_{\max}$, tol > 0
  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$, $\mathbf{v}_1 = \mathbf{r}_0/\beta$.
Ensure: Approximations $\mathbf{x}_j$.
  for $j = 1, 2, \ldots, m_{\max}$ **do**
    Arnoldi step: compute $h_{1:j+1,j}$ and $\mathbf{v}_{j+1}$ (see Alg. 12).
    Assemble $H_j = \overline{H}_j(1:j, 1:j)$ and $V_j = [\mathbf{v}_1, \ldots, \mathbf{v}_j]$.
    Solve $H_j \mathbf{y}_j = \beta \mathbf{e}_1$ and set $\mathbf{x}_j = \mathbf{x}_0 + V_j \mathbf{y}_j$.
    Compute $\|\mathbf{r}_j\|_2 = |h_{j+1,j}|\, |\mathbf{e}_j^\top \mathbf{y}_j|$ using (9.3).
    if $\|\mathbf{r}_j\|_2 \le$ tol **then**
      Return $\mathbf{x}_j$.
    if $h_{j+1,j} = 0$ **then**
      Break: a invariant subspace has been found and the exact solution is reached.

9.6 Practical notes and comparisons

- **Per-iteration cost.** Each iteration requires one matrix-vector product $A\mathbf{v}_j$ and orthogonalization of that vector against the preceding j Arnoldi vectors. The cost of orthogonalization grows with j (roughly $\mathcal{O}(jn)$) while the mat–vec depends on the structure (sparsity) of A.
- **Solving the projected system.** At iteration j we must solve the dense but small linear system $H_j \mathbf{y}_j = \beta \mathbf{e}_1$. Direct solution is usually cheap for moderate j ($\mathcal{O}(j^3)$ per solve); however algorithms can reuse or update factorizations to reduce cost.
- **Breakdown and exactness.** If $h_{j+1,j} = 0$ then the Arnoldi process has produced an invariant subspace of A, and the exact solution is contained in $\mathcal{K}_j$; FOM then finds the exact solution. In finite precision, loss of orthogonality is a concern and reorthogonalization may be necessary.

- **Relation to GMRES.** GMRES (Generalized Minimal Residual) uses the test space $A\mathcal{K}_m$ which yields a residual-minimizing condition. In contrast, the Galerkin requirement of FOM is simpler to state and leads to a square projected system. For non-normal matrices GMRES often performs better in practice because it explicitly minimizes the residual norm.
- **Preconditioning.** Standard right-preconditioning preserves the Krylov structure and can be applied with FOM; left-preconditioning changes the residuals and must be used with care (or requires a different formulation).

9.7 Summary

FOM is an instructive Krylov method: it combines the Arnoldi decomposition with a Galerkin projection on the Krylov space to reduce a large problem to a small dense one. The algorithm provides a cheap residual estimate and clear stopping criteria, while trade-offs exist with other methods (notably GMRES) regarding robustness, cost and memory use.

Chapter 10

Generalized Minimum Residual Method (GMRES)

Let $\mathcal{K} = \mathcal{K}_m(A, \mathbf{r}_0)$ and $\mathcal{L}_m = A\mathcal{K}_m$. This chapter describes the GMRES algorithm (Generalized Minimum Residual), an important Krylov-subspace method for solving nonsymmetric linear systems. We present the least-squares formulation, the practical QR/Givens implementation used in practice, a short worked example, and a visualisation of residual decay.

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

We therefore solve the overdetermined system ($m < n$)

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

via a least squares solve, efficiently obtained by QR factorization of the upper Hessenberg matrix $\overline{H}_m$ using Givens rotations.

10.1 QR factorization approach

Since $\overline{H}_m \in \mathbb{R}^{(m+1)\times m}$ is upper Hessenberg, we compute a thin QR

$$\overline{H}_m = Q_{m+1}\,\tilde{R}_m,$$

where $Q_{m+1} \in \mathbb{R}^{(m+1)\times(m+1)}$ is orthogonal and

$$\tilde{R}_m = \begin{bmatrix} R_m \\ \mathbf{0}^\top \end{bmatrix}, \qquad R_m \in \mathbb{R}^{m\times m} \text{ upper triangular.}$$

Set

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

Then the least squares problem becomes

$$\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y} = Q_{m+1}\big(\bar{\mathbf{g}}_m - \tilde{R}_m\mathbf{y}\big) \quad \Rightarrow \quad \|\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y}\|_2 = \|\bar{\mathbf{g}}_m - \tilde{R}_m\mathbf{y}\|_2$$
$$= \left\| \begin{bmatrix} \mathbf{g}_{1:m} - R_m\mathbf{y} \\ g_{m+1} \end{bmatrix} \right\|_2 = \|\mathbf{g}_{1:m} - R_m\mathbf{y}\|_2^2 + |g_{m+1}|^2.$$

Hence the minimizer is

$$\mathbf{y}_m = R_m^{-1}\mathbf{g}_{1:m}, \qquad \|\mathbf{r}_m\|_2 = |\gamma_{m+1}|.$$

We obtain the QR factorization incrementally using Givens rotations that annihilate the subdiagonal entries of each new column of $\overline{H}_m$.

Let

$$h = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}, \quad G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad c^2 + s^2 = 1,$$

then one chooses c, s so that

$$G\,h = \begin{bmatrix} \|h\| \\ 0 \end{bmatrix}, \qquad r = \sqrt{h_1^2 + h_2^2}, \quad c = \frac{h_1}{r}, \; s = \frac{h_2}{r}.$$

In the Arnoldi process, as each column is produced we apply one new Givens rotation (and update previous ones) so that after m steps we have

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

If after applying the kth rotation we have

$$\begin{bmatrix} \gamma_k \\ 0 \end{bmatrix} \xrightarrow{G_k} \begin{bmatrix} c_k\gamma_k \\ -s_k\gamma_k \end{bmatrix}, \qquad \|r_k\|_2 = |-s_k\gamma_k| = |s_k|\,\|r_{k-1}\|_2.$$

Thus $|s_k| \le 1$, and if $|s_k| < 1$ the residual norm strictly decreases. If $|s_k| = 1$ then $c_k = 0$, which typically indicates $h_{k,k} = 0$ (possible breakdown) or singular behavior of A in the Krylov subspace.

Explicitly,

$$c_k = \frac{h_{k,k}}{\sqrt{h_{k,k}^2 + h_{k+1,k}^2}}, \qquad s_k = \frac{h_{k+1,k}}{\sqrt{h_{k,k}^2 + h_{k+1,k}^2}}.$$

10.1.1 Practical remarks

In practice one performs the Arnoldi process and applies the Givens rotation for the new column immediately, keeping only the vectors of V_m, the (small) Hessenberg matrix entries and the rotation coefficients c_k, s_k. This yields an $\mathcal{O}(m^2)$ storage and $\mathcal{O}(m^2 n)$ cost for m steps. For large problems GMRES is typically restarted ("GMRES(m)") to limit storage: compute m steps, update the solution, then restart with the new residual as initial vector.

GMRES algorithm

Algorithm 11 GMRES (Arnoldi + Givens)

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
$\beta = \|\mathbf{r}_0\|_2$
$\mathbf{v}_1 = \mathbf{r}_0 / \beta$
for $j = 1, 2, \dots, m$ **do**
  $\mathbf{w}_j = A\mathbf{v}_j$
  for $i = 1, 2, \dots, j$ **do**
    $h_{ij} = \langle \mathbf{w}_j, \mathbf{v}_i \rangle$
    $\mathbf{w}_j \leftarrow \mathbf{w}_j - h_{ij}\mathbf{v}_i$
  $h_{j+1,j} = \|\mathbf{w}_j\|_2$
  if $h_{j+1,j} = 0$ **then**
    break
  $\mathbf{v}_{j+1} = \mathbf{w}_j / h_{j+1,j}$
  Update Givens rotations to introduce $\tilde{h}_{*,j}$
Form R_m and $\mathbf{g}_{1:m}$ from accumulated rotations
Solve $R_m \mathbf{y}_m = \mathbf{g}_{1:m}$
$\mathbf{x}_m = \mathbf{x}_0 + V_m \mathbf{y}_m$

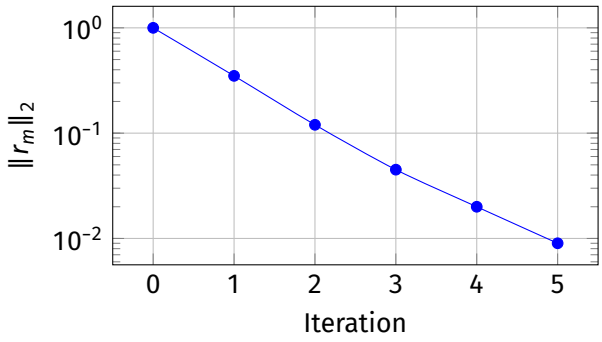


Figure 10.1: Residual norm vs iteration for a typical GMRES run (see the runnable demo in `examples/gmres_demo.py`).

Example 16. GMRES example

Consider the system

$$A = \begin{bmatrix} 4 & 1 & 2 \\ 0 & 3 & -1 \\ 1 & -1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The initial residual is $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = \mathbf{b}$ with norm $\beta = \|\mathbf{r}_0\|_2 = \sqrt{5}$. The first Arnoldi vector is $\mathbf{v}_1 = \mathbf{r}_0 / \beta$. We perform the Arnoldi process to build the Krylov basis and Hessenberg matrix, applying Givens rotations to maintain the QR factorization. After m steps, we solve the least-squares problem to find $\mathbf{y}_m$ and update the solution $\mathbf{x}_m = \mathbf{x}_0 + V_m \mathbf{y}_m$.

10.2 Convergence of GMRES

We give a standard polynomial-based bound. Let $\mathbf{x}_\star$ be the exact solution of $A\mathbf{x} = \mathbf{b}$ and let $\mathbf{x}_m$ be the iterate after m steps of a Krylov method. Then there exists a polynomial $p_m \in \mathbb{P}_m$ with $p_m(0) = 1$ such

that

$$\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m = p_m(A)\mathbf{r}_0.$$

If A is diagonalizable, $A = X\Lambda X^{-1}$, then

$$\|\mathbf{r}_m\|_2 \le \|X\|_2\|X^{-1}\|_2 \max_{1\le i\le n} |p_m(\lambda_i)|\ \|\mathbf{r}_0\|_2 = \kappa_2(X) \max_{\lambda\in\sigma(A)} |p_m(\lambda)|\ \|\mathbf{r}_0\|_2,$$

where $\kappa_2(X) = \|X\|_2\|X^{-1}\|_2$.

Thus to bound GMRES one seeks

$$\min_{\substack{p\in\mathbb{P}_m\\ p(0)=1}} \max_{\lambda\in E} |p(\lambda)|,$$

where E is a compact region (often an ellipse) that contains the spectrum of A.

Chebyshev polynomials (complex extension)

For $z \in \mathbb{C}$ with $|z| > 1$ one may use the analytic continuation of Chebyshev polynomials. With $\rho = \operatorname{arccosh}(z)$ and $w = e^{\rho}$,

$$C_m(z) = \cosh(m\rho) = \frac{1}{2}(w^m + w^{-m}), \quad z = \frac{1}{2}(w + w^{-1}),$$

and the three-term recurrence $C_{m+1}(z) = 2zC_m(z) - C_{m-1}(z)$ holds.

Lemma 1: Zarantello Let $\gamma \in \mathbb{C}$ with $|\gamma| > \rho$. Then

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

and the minimizer is $p(z) = \left(\frac{z}{\gamma}\right)^m$ (maximum attained at $|z| = \rho$).

Joukowsky map and ellipse bounds

The Joukowsky map

$$J(w) = \frac{1}{2}(w + w^{-1}), \quad w \in \mathbb{C}\setminus\{0\},$$

maps the circle $D_\rho = \{w : |w| = \rho\}$ to an ellipse

$$J(D_\rho) = \mathrm{E}\big(0, 1, \tfrac{1}{2}(\rho + \rho^{-1})\big).$$

Theorem 10.1: Elman

Let $J(D_\rho) = \mathrm{E}_\rho$ and choose $\gamma \notin \mathrm{E}_\rho$. Let w_γ be the preimage of γ under J with maximal modulus. Then

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

The near-optimal polynomial is

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

For a general ellipse $\mathrm{E}(c,d,a)$ containing the spectrum, one can scale and shift the Chebyshev polynomial:

$$\hat{C}_m(z) = \frac{C_m\!\left(\frac{z-c}{d}\right)}{C_m\!\left(-\frac{c}{d}\right)}, \qquad \hat{C}_m(0) = 1,$$

and obtain

$$\max_{z\in\mathrm{E}(c,d,a)} |\hat{C}_m(z)| = \frac{C_m(\frac{a}{d})}{\left|C_m(-\frac{c}{d})\right|}.$$

Therefore a practical bound for GMRES is

$$\|\mathbf{r}_m\|_2 \le \kappa_2(X)\,\varepsilon^m\,\|\mathbf{r}_0\|_2, \qquad \varepsilon^m = \frac{C_m(\frac{a}{d})}{\left|C_m(-\frac{c}{d})\right|} \approx \left(\frac{a+\sqrt{a^2-d^2}}{c+\sqrt{c^2-d^2}}\right)^m$$

for large m.

Note: the ellipse enclosing the eigenvalues must not include 0 (because $p(0)=1$ is required). If the ellipse is separated from the origin (e.g. $a < c$ in a standard parametrization) one obtains guaranteed geometric decay of the bound above.

Lectures

.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:

$$A\mathbf{U} = \mathbf{f},$$

$$A = \begin{bmatrix} B & -I_N & 0 & \cdots & 0 \\ -I_N & B & -I_N & \cdots & 0 \\ 0 & -I_N & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -I_N \\ 0 & 0 & 0 & -I_N & B \end{bmatrix}, \qquad B = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \cdots & 0 \\ 0 & -1 & 4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & -1 & 4 \end{bmatrix},$$

$$\mathbf{U} = \begin{bmatrix} U_{11} \\ U_{12} \\ \vdots \\ U_{1N} \\ U_{21} \\ U_{22} \\ \vdots \\ U_{NN} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_{11} \\ f_{12} \\ \vdots \\ f_{1N} \\ f_{21} \\ f_{22} \\ \vdots \\ f_{NN} \end{bmatrix}.$$

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 .2: 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 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 $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ 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 $0 < \omega < 2$

.1.2 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.

.1.3 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.

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

.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:

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

.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 $Q = [q_1, q_2, \ldots, q_n]$. 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.

.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 $k+1, k+2, \dots, m$ 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:

$$\sigma = \begin{cases} -1 & \text{if } \Re(x_1) > 0 \\ 1 & \text{if } \Re(x_1) \le 0 \end{cases}$$

$$u = x - \sigma\|x\|_2 e_1$$

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

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 $k = 1, 2, \dots, n$:

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: $R = P_2 P_1 A$ 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

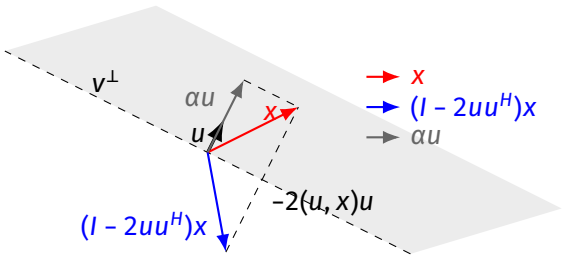
The goal of this figure is to make the algebraic action of a Householder reflector visually transparent. Let u be a unit vector (the reflector normal) and set

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

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.

.3 Lecture 3: 26.08.2025

.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 10
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.

.3.2 Matrix Properties and Non-singularity

Definition .3: 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 .4: Non-singularity of Strictly Diagonally Dominant Matrices
Every strictly diagonally dominant matrix is non-singular.

Definition .5: 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

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

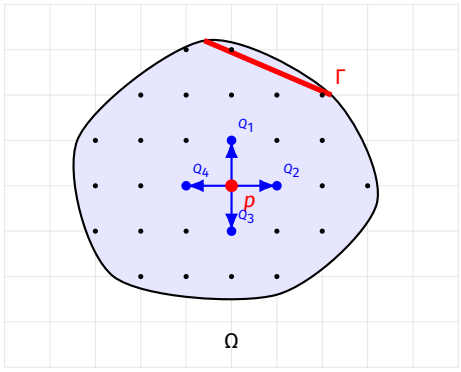
where A_{11} and A_{22} are square matrices.

Theorem .6: 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 17. 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



.3.3 Gershgorin Circle Theorem

Theorem .7: Gershgorin Circle Theorem

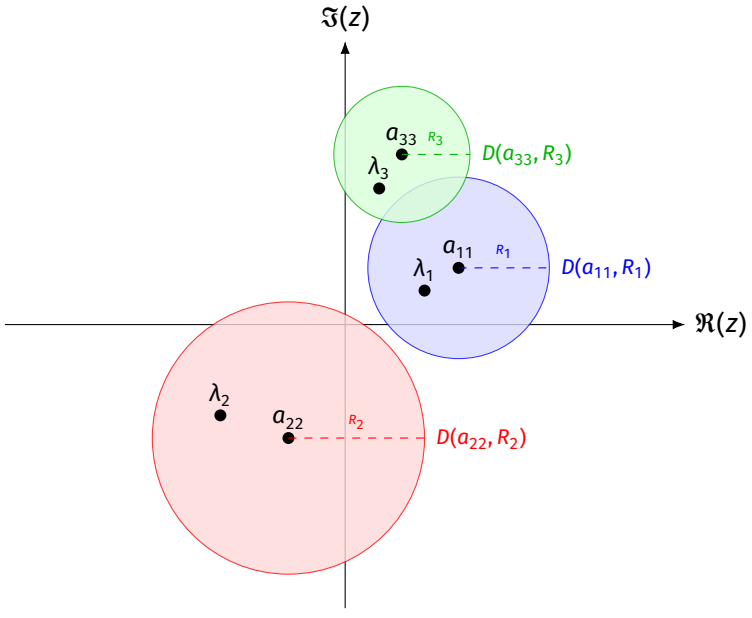
Let $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ and define the **row radii**:

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

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 .8: 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 .9: 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. □

.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.

.4 Lecture 4: 27.08.2025

.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

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

.4.2 Schur decomposition

Theorem .10: 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

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

.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.

.4.4 QR factorization

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

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

.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 $\kappa(\lambda) \to \infty$.

.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:

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

.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:

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

.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 $\|d_{k+1}\|_A \le C\|d_k\|_A$ for some $C < 1$.

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

.6 Lecture 6: 03.09.2025

Steepest Descent (SD)

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

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

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

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

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 $i = 1, \ldots, n$.

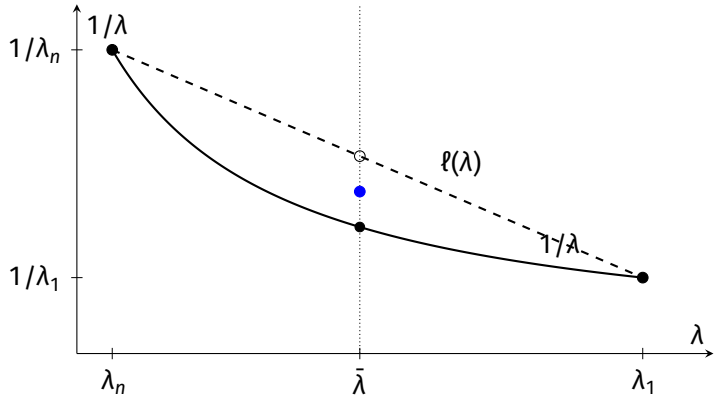
$$\ell(\lambda) = \frac{1}{\lambda_1} + \frac{1}{\lambda_n} - \frac{\lambda}{\lambda_1 \lambda_n}, \qquad \ell(\lambda_1) = \frac{1}{\lambda_1}, \qquad \ell(\lambda_n) = \frac{1}{\lambda_n}$$

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:

$$\frac{\|\mathbf{r}_k\|_2^4}{\|\mathbf{r}_k\|_A^2 \|\mathbf{r}_k\|_{A^{-1}}^2} \ge \frac{4\lambda_1\lambda_n}{(\lambda_1 + \lambda_n)^2}$$

$$\|\mathbf{d}_{k+1}\|_A^2 \le \|\mathbf{d}_k\|_A^2 \left(1 - 4\frac{\lambda_1\lambda_n}{(\lambda_1 + \lambda_n)^2}\right)$$

$$= \|\mathbf{d}_k\|_A^2 \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2$$

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:

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

$$\lambda_{\max} = 4 \text{ if } N \text{ odd}$$

$$\lambda_{\min} = 4 - 4\cos\left(\frac{\pi}{N+1}\right)$$

$$\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{4\cos\left(\frac{\pi}{N+1}\right)}{8 - 4\cos\left(\frac{\pi}{N+1}\right)} \approx 1 - \frac{1}{2}\left(\frac{\pi}{N+1}\right)^2 + \ldots$$

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 $\mathbf{x}_0$:

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

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*.

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

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.

$$\mathbf{v} = A^\top \mathbf{r}_0$$
$$\mathbf{w} = A\mathbf{r}_0$$

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}, $i, j = 1, \ldots, p$, 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:

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

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.

.7 Lecture 9: 09.09.2025

.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)

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

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

We now define the **Krylov subspace**:

Definition .11: Krylov Subspace

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

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

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

.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 .12: 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 18

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

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

.7.3 Cayley-Hamilton Theorem

Theorem .13: 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 $m \geq \mu$ = 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 μ = grade($\mathbf{v}$), then

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

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 2: 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:

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

.7.4 Practical implementation of Krylov Subspace Methods

Let

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

FOM: Full Orthogonalization Method

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

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

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

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.

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

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

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

Algorithm 12 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:

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

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 13 Full Orthogonalization Method (FOM)

Require:
$A \in \mathbb{R}^{n\times n}, \mathbf{b} \in \mathbb{R}^n, \mathbf{x}_0 \in \mathbb{R}^n, m_{\max} \in \mathbb{N}, \text{tol} > 0$
$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0, \quad \beta = \|\mathbf{r}_0\|_2$
$\mathbf{v}_1 = \mathbf{r}_0/\beta$
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. 12)
  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 $\|\mathbf{r}_j\|_2 \le \text{tol}$ **then**
    Return $\mathbf{x}_j$
  if $h_{j+1,j} = 0$ **then**
    Breakdown: exact solution in $\mathcal{K}_j$ (stop)

.8 Lecture 10: 10.09.2025

.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 14 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:

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

With the relations:

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

.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

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

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.

.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

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

The least squares problem becomes:

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

Then:

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

Then we do QR factorization by Givens rotations:

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

In the Arnoldi process for $k = 1$:

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

For $k = 2$:

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

after m iterations we have:

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

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

$|s_k| \leq 1$
If $|s_k| < 1$, then $\|r_k\| < \|r_{k-1}\|$
If $|s_k| = 1$, then stagnation, but then $c_k = 0$ which means $h_{k,k} = 0$ or A 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 15 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)

.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., $A = A^T > 0$.

.9.1 Recap: Arnoldi iteration

Algorithm 16 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

$$AV_m = V_{m+1}\bar{H}_m$$
$$V_m^T A V_m = H_m$$

Where we solve the reduced linear system:

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

How can this be simplified if $A = A^T$?

In this case $H_m = V_m^T A V_m = H_m^T$ 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:

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

.9.2 Lanczos iteration

Algorithm 17 Lanczos: Arnoldi for symmetric $A = A^\top$

Require: $A, \mathbf{b}, \mathbf{x}_0, m$
  $\beta_1 = 0$
  $\mathbf{v}_0 = 0$
  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
  $\beta = \|\mathbf{r}_0\|_2$
  $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\beta}$
  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.

.9.3 Conjugate gradient (CG) method

Proposition 5

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

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.

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

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

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

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

- So: A lower triangular symmetric matrix is diagonal.

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

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

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 18 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.

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

.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:

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

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.

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

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:

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

.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:

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

Thus we have the final bound for CG:

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

.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 $N_{\text{pot}} \in (0,1)$.

.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:

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

.11 Lecture 13: 23.09.2025

.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 $p(0) = 1$.

.11.2 GMRES

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

$$\begin{aligned}
\|\mathbf{r}_m\|_2 &= \min_{\mathbf{x}\in\mathbf{x}_0+\mathcal{K}_m} \|\mathbf{b} - A\mathbf{x}\|_2 \\
\|\mathbf{r}_m\|_2 &\le \|\mathbf{r}_{m-1}\|_2 \le \dots \le \|\mathbf{r}_0\|_2
\end{aligned}$$

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

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

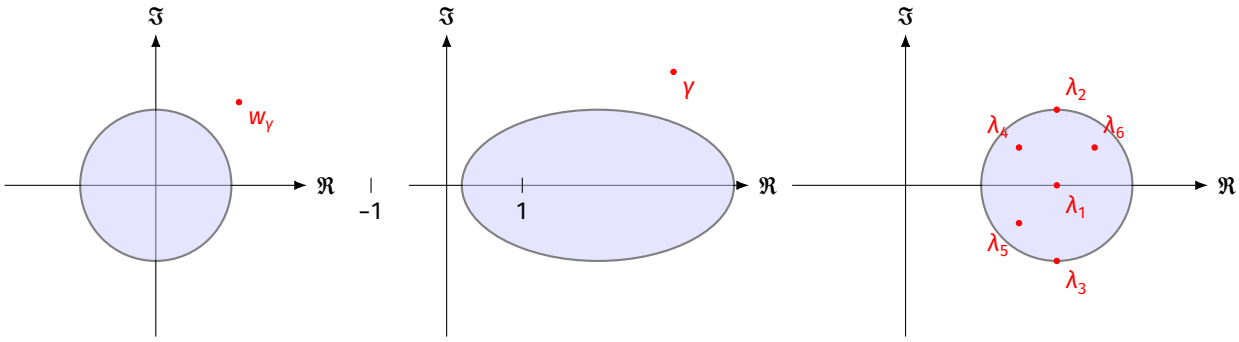
A may not be diagonalizable.

Now assume A is diagonalizable:

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

but X is not orthogonal anymore.

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



Let $\lambda_i \in \mathrm{E}$ for $i = 1, \ldots, n$, 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 3: 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 .14: 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.

.12 Lecture 14: 24.09.2025

.12.1 Convergence

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

- **CG:**

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

- **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:

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

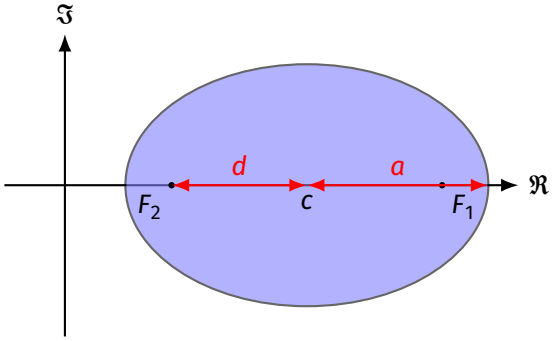
  For an ellipse one can use Chebyshev-type estimates to get an explicit geometric rate. Defining

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

  one convenient bound is

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

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



.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 19 Arnoldi process with RPC

$\beta = \|\mathbf{r}_0\|_2$, $\mathbf{v}_1 = \mathbf{r}_0/\beta$
$\mathbf{w}_j = AM^{-1}\mathbf{v}_j$
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
$\mathbf{v}_{j+1} = \mathbf{w}_j / h_{j+1,j}$ **return** $\overline{H}_m, V_m$

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:

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

.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 20 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 $\|\tilde{\mathbf{r}}_{j+1}\|_2 <$ 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 21 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 $j = 0, 1, 2, \ldots$ **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 $\|\mathbf{r}_{j+1}\|_2 <$ 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*

.13 Lecture 15: 25/09/2025

.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 22 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.

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

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

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

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. $M^{-1}A \approx I_n$.

.13.2 Preconditioning the CG method

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

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

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 23 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**
$\alpha_j = \frac{\tilde{\mathbf{r}}_j^T \tilde{\mathbf{r}}_j}{\tilde{\mathbf{p}}_j^T \tilde{A} \tilde{\mathbf{p}}_j} = \frac{\|\tilde{\mathbf{r}}_j\|_2^2}{\|\tilde{p}_j\|_A^2}$
$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \tilde{\mathbf{p}}_j$
$\tilde{\mathbf{r}}_{j+1} = \tilde{\mathbf{r}}_j - \alpha_j \tilde{A} \tilde{\mathbf{p}}_j$
$\beta_j = \frac{\tilde{\mathbf{r}}_{j+1}^T \tilde{\mathbf{r}}_{j+1}}{\tilde{\mathbf{r}}_j^T \tilde{\mathbf{r}}_j} = \frac{\|\tilde{\mathbf{r}}_{j+1}\|_2^2}{\|\tilde{\mathbf{r}}_j\|_2^2}$
$\tilde{\mathbf{p}}_{j+1} = \tilde{\mathbf{r}}_{j+1} + \beta_j \tilde{\mathbf{p}}_j$

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 24 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 $\mathbf{z}_j$.

.13.3 Choosing a preconditioner

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