TMA4205

Numerical Linear Algebra

Compendium in Numerical Linear Algebra

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

Semester
Autumn 2025
Last updated: October 4, 2025

Norwegian University of Science and Technology
Department of Mathematical Sciences

Contents

I Foundations of Numerical Linear Algebra 1

1 Introduction 3

1.1 Large Sparse Problems 3

2 Preliminaries 5

2.1 Notation and Conventions 5

2.2 Matrices 6

2.2.1 Eigenvalues and Eigenvectors 6

2.2.2 Image (Range) and Kernel (Nullspace) 6

2.2.3 Normal Matrices 6

2.2.4 Hermitian Matrices 7

2.2.5 Nonnegative Matrices 8

2.2.6 Kantorovich Inequality 9

2.2.7 M-matrices 9

2.2.8 Unitary Matrices 10

2.3 Orthogonal Vectors and Subspaces 12

2.3.1 QR Decomposition 12

2.3.2 Gram-Schmidt Process 12

2.3.3 Modified Gram-Schmidt Process 14

2.3.4 QR Decomposition 16

2.3.5 Householder Reflections 17

2.4 Canonical Forms and Matrix Structure 20

2.4.1 Similarity of Matrices 20

2.4.2 Affine Spaces and Affine Maps 21

2.4.3 Matrix Polynomials 22

2.4.4 Jordan Canonical Form 23

2.4.5 Schur Decomposition 23

2.5 Diagonal Dominance and Gershgorin Discs 25

2.5.1 Diagonal Dominance and Nonsingularity 25

2.5.2 Gershgorin Circle Theorem 26

3 Linear Systems 29

3.1 Types of Linear Systems 29

3.1.1 Overdetermined Systems 29

3.1.2 Underdetermined Systems 29

3.1.3 Square Systems 29

3.1.4 Homogeneous Systems 29

3.2 Existence and Uniqueness of Solutions 30

3.3 Methods for Solving Linear Systems 30

3.4 Matrix Storage . . . 30
3.4.1 Model Problem: 2D Poisson and Sparsity . . . 31
3.4.2 Spectrum of the Discrete 2D Laplacian . . . 31
3.5 Perturbation Analysis . . . 31
3.5.1 Perturbation Framework . . . 32
3.5.2 Condition Number . . . 32
3.5.3 Perturbation Bounds . . . 32
3.5.4 Residual Analysis . . . 33

4 Preconditioning 35
4.1 Forms of Preconditioning . . . 35
4.2 Spectral Effects . . . 35
4.3 Basic Preconditioners . . . 35
4.4 Restarts and Flexibility . . . 36

II Projection Methods for Linear Systems 37

5 Projections 39
5.1 Notation and Setup . . . 39
5.2 Projection Operator . . . 40
5.2.1 Properties of Projections . . . 40
5.3 Reduced Systems . . . 41
5.4 Matrix Representation of Projections . . . 41
5.5 Oblique Projections . . . 42

6 Krylov Subspaces 45
6.1 Properties of Krylov Subspaces . . . 45
6.1.1 Minimal polynomial and grade . . . 45
6.2 Arnoldi Iteration . . . 46
6.2.1 Derivation of Arnoldi Iteration . . . 46
6.2.2 The Arnoldi Algorithm . . . 46
6.2.3 Arnoldi Relation . . . 47
6.2.4 Breakdown Conditions . . . 47
6.2.5 Numerical Stability and Reorthogonalization . . . 48
6.2.6 Computational Complexity . . . 48
6.2.7 Applications: The Foundation for Krylov Solvers . . . 48
6.3 Lanczos Iteration . . . 49
6.3.1 Derivation of Lanczos Iteration . . . 49

III Iterative Solvers for Linear Systems 53
6.4 Steepest Descent (SD) . . . 55
6.4.1 Conjugate gradient (CG) method . . . 57
6.5 Full Orthogonalization Method (FOM) . . . 60
6.6 Generalized Minimum Residual Method (GMRES) . . . 61
6.6.1 QR Factorization Approach . . . 62
6.6.2 Convergence of GMRES . . . 64

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 $\mathbf{e} = (1, \dots, 1)^{\mathrm{T}}$,
- $\operatorname{tr}(A)$ 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 2.2.1. 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 2.2.2. 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 2.2.3. 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 $AA^T = A^TA$.

Normal matrices are special because they *commute* with their conjugate transpose; meaning $AA^H = A^HA$. 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 2.2.4. 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 $D = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ 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 2.2.5. 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 2.2.6. 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

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

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 2.2.7. 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 2.2.8. 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 2.2.9. 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 $\Lambda = \mathrm{diag}(\lambda_1,\dots,\lambda_n)$ 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

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

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.

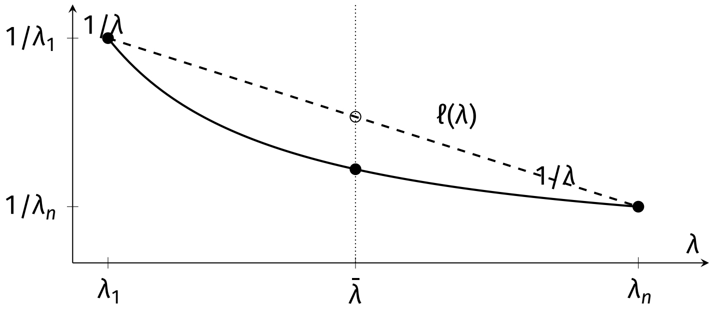


Figure 2.1: Kantorovich inequality visualized via the convexity of $1/\lambda$ and its chord between λ_1 and λ_n.

Definition 2.2.10. M-matrix

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

1. $a_{ij} \le 0$ for all $i \ne j$ (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 $Q = [q_1, q_2, \ldots, q_n]$. Then the orthonormality condition is:

$$(q_i, q_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{2.1}$$

Examples of Unitary Matrices

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

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{2.2}$$

   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:

$$G(i, j, \theta) = \begin{bmatrix} I_{i-1} & & & \\ & c & -s & \\ & s & c & \\ & & & I_{n-j} \end{bmatrix} \tag{2.3}$$

   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 $\|v\|_2 = 1$:

$$P = I_n - 2vv^H \tag{2.4}$$

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

   Verification of unitarity:

$$\begin{aligned} P^H P &= (I_n - 2vv^H)^2 && (2.5) \\ &= I_n - 4vv^H + 4v(v^H v)v^H && (2.6) \\ &= I_n - 4vv^H + 4vv^H = I_n && (2.7) \end{aligned}$$

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

$$P\mathbf{x} = \mathbf{x} - 2(v^H \mathbf{x})v = \mathbf{x} - 2(\mathbf{x}, v)v \tag{2.8}$$

   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 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 2.3.1. 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.

2.3.1 QR Decomposition

The QR decomposition is a fundamental matrix factorization that uses orthogonal (or unitary) matrices to decompose a given matrix into a product of an orthogonal matrix and an upper triangular 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. When A has full column rank one commonly uses the thin factorization (meaning $m \geq n$, i.e. A might not be square):

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

which is unique up to multiplication of Q_{thin} on the right by a diagonal unitary (in the real case, by signs). The QR factorization is widely used in:

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

2.3.2 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 2.3.2. 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:

$$q_1 = \frac{v_1}{\|v_1\|}, \tag{2.9}$$

$$q_k = \frac{u_k}{\|u_k\|}, \quad k = 2, 3, \dots, n, \tag{2.10}$$

$$u_k = v_k - \sum_{j=1}^{k-1} \langle v_k, q_j \rangle q_j. \tag{2.11}$$

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 1 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
```

QR Decomposition via Gram-Schmidt

Let's illustrate the classical Gram-Schmidt process by computing a (thin) QR decomposition of

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad a_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

We seek orthonormal columns q_1, q_2 and an upper triangular matrix $R \in \mathbb{R}^{2\times2}$ such that $A = \tilde{Q}R$.

Step-by-step computation:

1. Normalize the first column:

$$r_{11} = \|a_1\| = \sqrt{1^2 + 1^2 + 0^2} = \sqrt{2},$$

$$q_1 = \frac{a_1}{r_{11}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

2. Orthogonalize and normalize the second column:

$$
\begin{aligned}
r_{12} &= q_1^\top a_2 = \frac{1}{\sqrt{2}}(1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1) = \frac{1}{\sqrt{2}}, \\
u_2 &= a_2 - r_{12} q_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 1 \end{bmatrix}, \\
r_{22} &= \|u_2\| = \sqrt{\frac{1}{4} + \frac{1}{4} + 1} = \frac{\sqrt{6}}{2}, \\
q_2 &= \frac{u_2}{r_{22}} = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}.
\end{aligned}
$$

Resulting QR factors:

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

It is straightforward to verify that $\tilde{Q}^\top \tilde{Q} = I_2$ and $A = \tilde{Q}R$.

> **Remark 3. 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.

2.3.3 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 2 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.

QR Decomposition via Modified Gram-Schmidt

Let's work through the modified Gram-Schmidt (MGS) process for the same example as in Section 2.3.2:

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad a_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

We seek orthonormal columns q_1, q_2 and an upper triangular matrix $R \in \mathbb{R}^{2\times 2}$ such that $A = \tilde{Q}R$.

Step-by-step computation:

1. **Initialize:** Set $v_1 = a_1$, $v_2 = a_2$.
2. **First vector:**

$$r_{11} = \|v_1\| = \sqrt{1^2 + 1^2 + 0^2} = \sqrt{2},$$
$$q_1 = \frac{v_1}{r_{11}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

3. **Orthogonalize v_2 against q_1:**

$$r_{12} = q_1^\top v_2 = \frac{1}{\sqrt{2}}(1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1) = \frac{1}{\sqrt{2}},$$
$$v_2' = v_2 - r_{12} q_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ 1 \end{bmatrix}.$$

4. **Normalize v_2' to get q_2:**

$$r_{22} = \|v_2'\| = \sqrt{\left(\frac{1}{2}\right)^2 + \left(-\frac{1}{2}\right)^2 + 1^2} = \frac{\sqrt{6}}{2},$$
$$q_2 = \frac{v_2'}{r_{22}} = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}.$$

Resulting QR factors:

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

As with classical Gram-Schmidt, $\tilde{Q}^\top \tilde{Q} = I_2$ and $A = \tilde{Q}R$. However, the MGS process is more robust in finite precision arithmetic, better preserving orthogonality.

> **Remark 4. Comparison with Classical Gram-Schmidt**
> The numerical results of classical and modified Gram-Schmidt are identical in exact arithmetic, but MGS is much more stable in finite precision. In practice, MGS better preserves orthogonality, especially for nearly linearly dependent columns.

While Gram-Schmidt methods provide intuitive insight into orthogonalization, Householder reflections offer a more robust approach for practical computations. These transformations are based on geometric reflections and provide better numerical stability.

2.3.4 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:

$$P_n P_{n-1} \cdots P_2 P_1 A = R \tag{2.12}$$

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

$$A = \underbrace{P_1^H P_2^H \cdots P_n^H}_{Q} R \tag{2.13}$$

Algorithm

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

Construction of Householder vector:

$$\sigma = \begin{cases} -1 & \text{if } \Re(x_1) > 0 \\ 1 & \text{if } \Re(x_1) \leq 0 \end{cases} \tag{2.14}$$

$$\mathbf{u} = \mathbf{x} - \sigma\|\mathbf{x}\|_2 e_1 \tag{2.15}$$

$$v = \frac{\mathbf{u}}{\|\mathbf{u}\|_2} \tag{2.16}$$

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

Result: $P\mathbf{x} = (I - 2vv^H)\mathbf{x} = -\sigma\|\mathbf{x}\|_2 e_1$

Full QR Algorithm

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

1. Extract subcolumn: $\mathbf{x} = A_{k:m,k}$
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:

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

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

2.3.5 Householder Reflections

Householder reflections are a powerful tool for orthogonal transformations in numerical linear algebra. They are particularly useful for QR decomposition due to their numerical stability and efficiency.

Definition 2.3.3. 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 $\mathbf{u}$.

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. $\det(H) = -1$.

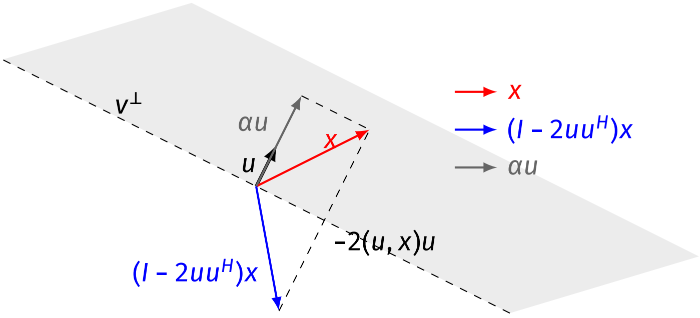


Figure 2.2: Householder reflection of $\mathbf{x}$ across the hyperplane orthogonal to $\mathbf{u}$. The projection $\pi_u(\mathbf{x})$ is in grey and the reflected vector Hx is in blue.

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. $\det(H) = \det(I - 2uu^\top) = \det(I)\det(I - 2u^\top u) = 1 \cdot (1 - 2) = -1$.

Constructing a Householder reflector

Theorem 2.3.4. Vector annihilation

For any nonzero $\mathbf{x} \in \mathbb{R}^n$, there is a Householder matrix H such that

$$Hx = \sigma e_1, \qquad \sigma = \pm\|\mathbf{x}\|.$$

A numerically stable choice is $\sigma = -\operatorname{sign}(x_1)\,\|\mathbf{x}\|$ (with $\operatorname{sign}(0) = 1$). Define $\mathbf{v} = \mathbf{x} - \sigma e_1$, $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$, and set $H = I - 2uu^\top$.

Proof. Let $\mathbf{v} = \mathbf{x} - \sigma e_1$ and $\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$.
Then $Hx = \mathbf{x} - 2u(\mathbf{u}^\top\mathbf{x}) = \mathbf{x} - \frac{2v(\mathbf{v}^\top\mathbf{x})}{\|\mathbf{v}\|^2}$.
Now $\mathbf{v}^\top\mathbf{x} = \mathbf{x}^\top\mathbf{x} - \sigma x_1 = \|\mathbf{x}\|^2 - \sigma x_1$ and $\|\mathbf{v}\|^2 = (\mathbf{x} - \sigma e_1)^\top(\mathbf{x} - \sigma e_1) = 2(\|\mathbf{x}\|^2 - \sigma x_1)$.
Hence $\dfrac{2v(\mathbf{v}^\top\mathbf{x})}{\|\mathbf{v}\|^2} = \mathbf{v}$ and $Hx = \mathbf{x} - \mathbf{v} = \sigma e_1$.
The sign choice $\sigma = -\operatorname{sign}(x_1)\|\mathbf{x}\|$ maximizes $\|\mathbf{v}\|$ and avoids cancellation when $\mathbf{x} \approx \sigma e_1$.

Algorithm 3 Householder vector (stable sign)

```
Require: $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} \neq 0$
Ensure: $\mathbf{u}$ with $\|\mathbf{u}\| = 1$ such that $(I - 2uu^\top)\mathbf{x} = \sigma e_1$, where $\sigma = -\operatorname{sign}(x_1)\|\mathbf{x}\|$
  $\alpha \leftarrow \|\mathbf{x}\|$
  $\sigma \leftarrow -\operatorname{sign}(x_1)\,\alpha$                ▷ take sign(0) = 1
  $\mathbf{v} \leftarrow \mathbf{x} - \sigma e_1$
  $\mathbf{u} \leftarrow \mathbf{v}/\|\mathbf{v}\|$
  return $\mathbf{u}$
```

QR decomposition via Householder reflections

Theorem 2.3.5. 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 $Q = (H_n \cdots H_1)^{\mathsf{T}}$ 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 4 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
            $\mathbf{u} \leftarrow \mathbf{v}/\beta$
            $H_k \leftarrow I_{m-k+1} - 2\,\mathbf{u}\mathbf{u}^{\mathsf{T}}$                ▷ 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
$Q \leftarrow Q^{\mathsf{T}}$                ▷ $Q \leftarrow H_n \cdots H_1 \implies Q = H_1 \cdots H_n$
```

Remark 5. 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

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

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

Example 3. A 3×2 example

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

Step 1.

$$\begin{aligned} a_1 &= (1, 1, 0)^{\mathsf{T}}, \quad \|a_1\| = \sqrt{2}, \quad \sigma_1 = -\sqrt{2} \\ v_1 &= a_1 - \sigma_1 e_1 = (1 + \sqrt{2},\ 1,\ 0)^{\mathsf{T}} \\ u_1 &= \frac{v_1}{\|v_1\|} \end{aligned}$$

Then

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

Step 2. Take the subvector $\mathbf{x} = \left(-\frac{1}{\sqrt{2}}, 1\right)^T$, $\|\mathbf{x}\| = \sqrt{\frac{3}{2}} = \frac{\sqrt{6}}{2}$, $\sigma_2 = \frac{\sqrt{6}}{2}$. 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

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

2.4 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 2.4.1. 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.

2.4.1 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 6. 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 2.4.2. 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).

2.4.2 Affine Spaces and Affine Maps

Definition 2.4.3. 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,

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

the set of all affine combinations of points in S.

Definition 2.4.4. 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.

2.4.3 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 7. 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. $\deg(\mu_A) \le n$ 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

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

2.4.4 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 2.4.5. 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 8. 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.

2.4.5 Schur Decomposition

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

Theorem 2.4.6. 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. $Q_1^H Q_1 = I$ (unitary).

Then

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

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)

QR Algorithm

The QR algorithm computes the Schur decomposition iteratively:

Algorithm 5 Basic QR Algorithm for Schur Decomposition

```
Require: $A^{(0)} = A$, $Q^{(0)} = I$
for $k = 1, 2, \ldots$ do
    $A^{(k-1)} = Q_k R_k$ (QR decomposition)
    $A^{(k)} = R_k Q_k$
    $Q^{(k)} = Q^{(k-1)} Q_k$
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 $Px = -\pi_u(x) + (x - \pi_u(x))$.

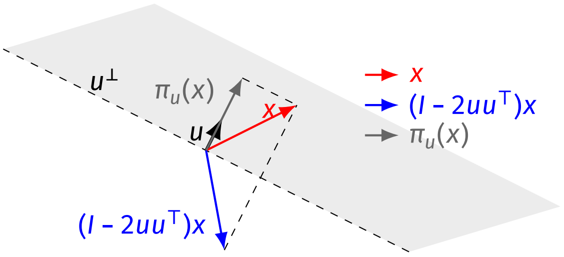


Figure 2.3: Householder reflection across the hyperplane orthogonal to u.

QR via 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 6 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 9. 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.

2.5 Diagonal Dominance and Gershgorin Discs

2.5.1 Diagonal Dominance and Nonsingularity

Definition 2.5.1. Strict Diagonal Dominance

A matrix $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ is *strictly diagonally dominant by rows* if

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \qquad i = 1, \dots, n.$$

Theorem 2.5.2. Levy–Desplanques

If A is strictly diagonally dominant by rows, then A is nonsingular.

Definition 2.5.3. Irreducible Matrix

$A \in \mathbb{C}^{n\times n}$ is *irreducible* if there is no permutation P such that PAP^T is block upper triangular with a zero block below the diagonal. Equivalently, the directed graph of A is strongly connected.

Theorem 2.5.4. Varga's criterion

If A is irreducible and diagonally dominant with at least one strict inequality, then A is nonsingular.

These criteria are useful for discretizations of elliptic PDEs (e.g., Poisson), where stencil couplings yield (often irreducible) diagonally dominant matrices.

2.5.2 Gershgorin Circle Theorem

Definition 2.5.5. Gershgorin row discs

For $A = (a_{ij}) \in \mathbb{C}^{n\times n}$, define radii $R_i = \sum_{j\neq i} |a_{ij}|$. The *row discs* are

$$D(a_{ii}, R_i) = \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i\}.$$

Theorem 2.5.6. Gershgorin

The spectrum satisfies

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

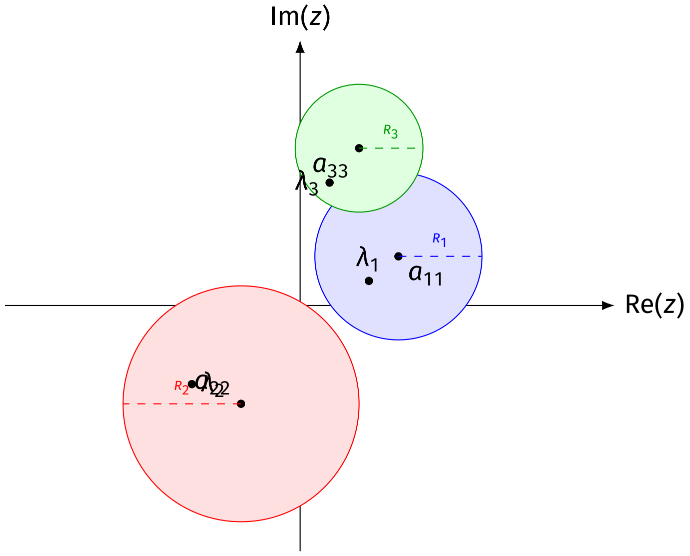


Figure 2.4: Gershgorin row discs $D(a_{ii}, R_i)$ enclosing the spectrum $\sigma(A)$ in the complex plane.

Proof sketch. Let $A\mathbf{v} = \lambda\mathbf{v}$, choose m with $|v_m| = \|\mathbf{v}\|_\infty$. From the mth equation,

$$(\lambda - a_{mm})v_m = \sum_{j\neq m} a_{mj}v_j, \quad \Rightarrow \quad |\lambda - a_{mm}| \leq \sum_{j\neq m} |a_{mj}| \frac{|v_j|}{|v_m|} \leq R_m.$$

Theorem 2.5.7. Separation

If the union of discs can be partitioned into two disjoint groups S_1, S_2, then A has as many eigenvalues (counting multiplicities) in S_k as the number of discs in S_k.

Theorem 2.5.8. Boundary for irreducible matrices

If A is irreducible and an eigenvalue lies on the boundary of one disc, then it lies on the boundary of all discs.

Remark 10. Homotopy and continuity

Consider $A(t) = D + tH$ with $D = \mathrm{diag}(a_{11}, \dots, a_{nn})$ and $H = A - D$. Eigenvalues vary continuously with t, and Gershgorin discs track their locations from $\{a_{ii}\}$ at $t = 0$ to $\sigma(A)$ at $t = 1$; this provides intuition for perturbations.

Convergence: subdiagonal entries → 0, eigenvalues shift to diagonal.

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 11. 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.

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

3.1.1 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:

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

3.1.2 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:

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

3.1.3 Square Systems

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

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

3.1.4 Homogeneous Systems

If a linear system is **homogeneous** then:

$$A\mathbf{x} = \mathbf{0}. \tag{3.2}$$

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 $\mathbf{b} \notin \mathrm{Im}(A)$. 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 3.4.1. 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 $\lambda_{\min} = 4 - 4\cos(\frac{\pi}{N+1})$ 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: $\begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}\mathbf{x} = \begin{pmatrix} 1 \\ 8 \end{pmatrix}$ Since rank$(A) = 2 = n$ and A is invertible:

$$\mathbf{x} = \begin{pmatrix} \frac{1}{2} \\ 2 \end{pmatrix} \qquad \text{(unique solution)}$$

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:

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

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:

$$\tilde{A} = A + \Delta A, \tag{3.3}$$

$$\tilde{\mathbf{b}} = \mathbf{b} + \Delta\mathbf{b}, \tag{3.4}$$

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) and using $A\mathbf{x} = \mathbf{b}$:

$$A\Delta\mathbf{x} + \Delta A(\mathbf{x} + \Delta\mathbf{x}) = \Delta\mathbf{b}. \tag{3.6}$$

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

$$A\Delta\mathbf{x} = \Delta\mathbf{b} - \Delta A\mathbf{x}. \tag{3.7}$$

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

$$\Delta\mathbf{x} = A^{-1}\Delta\mathbf{b} - A^{-1}\Delta A\mathbf{x}. \tag{3.8}$$

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:

$$\kappa(A) = \|A\|\,\|A^{-1}\|. \tag{3.9}$$

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 section 2.3).
- $\kappa(A) = +\infty$ 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 $\tilde{\mathbf{x}}$ to $A\mathbf{x} = \mathbf{b}$, we distinguish between:

- **Residual:** $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$ (computable)
- **Error:** $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$ (typically unknown)

The condition number relates these quantities:

$$\frac{1}{\kappa(A)}\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} \leq \frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq \kappa(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}. \tag{3.10}$$

Remark 12. Residual and Error Bounds

The bounds in (3.10) show that:

- For well-conditioned systems, small residual implies small error:

$$\|\mathbf{r}\| \ll \|\mathbf{b}\| \implies \|\mathbf{e}\| \ll \|\mathbf{x}\|.$$

- For ill-conditioned systems, small residual does *not* guarantee small error:

$$\|\mathbf{r}\| \ll \|\mathbf{b}\| \text{ does not imply } \|\mathbf{e}\| \ll \|\mathbf{x}\|.$$

- The 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

Preconditioning

Preconditioning transforms a linear system into an equivalent one that is easier to solve iteratively. It improves spectral properties, reduces iteration counts, and can enable restarts to be effective.

- distinguish left, right, and split preconditioning,
- understand spectral effects and conditioning improvements,
- recognize basic preconditioners (Jacobi/SSOR/ILU/IC) and when to use them.

4.1 Forms of Preconditioning

Let $M \approx A$ be nonsingular.

- Left: solve $M^{-1}A\,\mathbf{x} = M^{-1}\mathbf{b}$; Krylov space is $\mathcal{K}_m(M^{-1}A, M^{-1}\mathbf{r}_0)$.
- Right: solve $AM^{-1}\,\mathbf{y} = \mathbf{b}$ and set $\mathbf{x} = M^{-1}\mathbf{y}$; residuals live in the original space.
- Split/symmetric: e.g., $M = LL^T$ with SPD M for CG.

4.2 Spectral Effects

Ideal preconditioning clusters eigenvalues and reduces κ. For CG, use SPD M and apply CG to $M^{-1}A$ in the M-inner product. For GMRES, right preconditioning preserves residual minimization in the Euclidean norm.

4.3 Basic Preconditioners

- Jacobi (diagonal): $M = \mathrm{diag}(A)$; cheap, modest improvement.
- SSOR: uses symmetric Gauss–Seidel splitting; SPD if A is SPD.
- Incomplete LU/Cholesky (ILU/IC): drop small/far fill to approximate LU/Cholesky; strong in practice on sparse PDE matrices.
- Algebraic/Geometric multigrid (AMG/GMG): powerful for elliptic PDEs; often used as preconditioners for CG/GMRES.

4.4 Restarts and Flexibility

Restarted GMRES(m) limits memory; with good preconditioning, restarts remain effective. Flexible GMRES allows iteration-varying preconditioners (e.g., inner solves), at the cost of additional storage.

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 range$(V) = \mathcal{K}$.
- $W = [\mathbf{w}_1, \dots, \mathbf{w}_m] \in \mathbb{R}^{n\times m}$ with 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 $B \in \mathbb{R}^{n\times m}$,

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

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

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 5.2.1. 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.

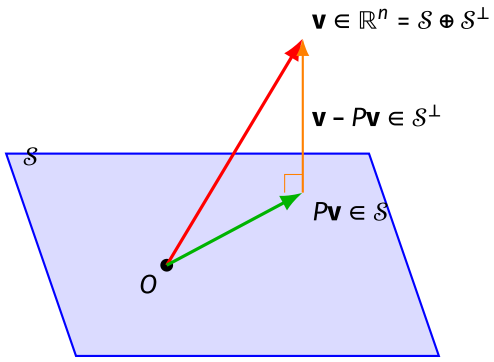
5.2.1 Properties of Projections

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 $P\mathbf{v} = P^2 u = Pu = v$. 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.


$\mathbf{v} \in \mathbb{R}^n = \mathcal{S} \oplus \mathcal{S}^\perp$
$\mathbf{v} - P\mathbf{v} \in \mathcal{S}^\perp$
$\mathcal{S}$
$P\mathbf{v} \in \mathcal{S}$
O

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 $\mathbf{v} - P\mathbf{v}$.

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 $\mathrm{range}(V) = \mathrm{range}(W) = \mathcal{K}$, 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, $V^\top AV = (CV)^\top(CV) > 0$.

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 $\mathrm{range}(V) = \mathcal{K}$, 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 5.4.1. 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 $\mathrm{Col}(P)$ along $\mathrm{Null}(P)$.

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 9. 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 5.5.1. 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 $\mathrm{range}(P) = \mathcal{S}$ and $\mathrm{null}(P) = \mathcal{T}$; 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 $\mathrm{range}(S) = \mathcal{S}$.

- If $W \in \mathbb{R}^{n\times k}$ has full column rank with $\ker(W^\top) = \mathcal{T}$ (i.e., $\mathrm{range}(W) = \mathcal{T}^\perp$) 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 10. Oblique projection in iterative methods

In iterative solvers for $A\mathbf{x} = \mathbf{b}$, choose a *search space* $\mathcal{K} = \mathrm{span}(V)$ and a *test space* $\mathrm{span}(W)$; 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 $\mathrm{span}(W) = A\mathcal{K}$ (as in GMRES), the residual is enforced orthogonal to $A\mathcal{K}$, i.e., the projection is along $(A\mathcal{K})^\perp$.

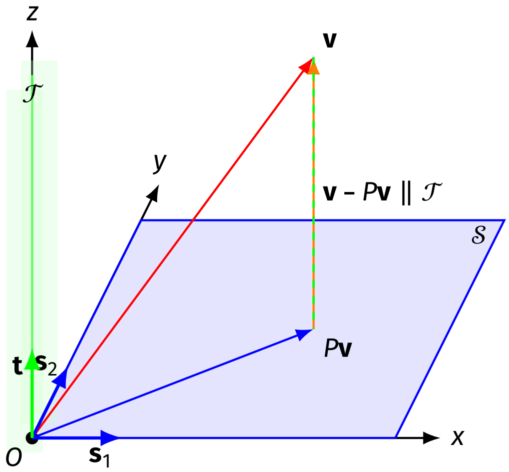


Figure 5.2: Oblique projection: $P\mathbf{v} \in \mathcal{S}$, $\mathbf{v} - P\mathbf{v} \in \mathcal{T}$ (not necessarily orthogonal). Basis vectors $\mathbf{s} \in \mathcal{S}$ and $\mathbf{t} \in \mathcal{T}$ are shown.

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 6.0.1. 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 ($V^*V = I$).

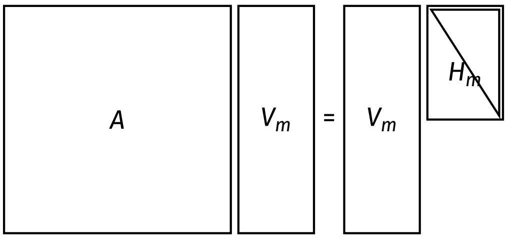
But why do we want to compute $A = VHV^*$? The reason is that such a *unitary similarity* preserves the eigenvalues of A, while making the matrix H much simpler in structure (upper Hessenberg). Indeed, since $H = V^*AV$, if $H\mathbf{x} = \lambda\mathbf{x}$, then

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

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:

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

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:

$$V_{m+1} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{m+1}] \in \mathbb{R}^{n\times(m+1)} \quad \text{(orthonormal basis)} \tag{6.1}$$

$$V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m] \in \mathbb{R}^{n\times m} \quad \text{(first } m \text{ columns)} \tag{6.2}$$

$$\overline{H}_m = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,m} \\ 0 & h_{3,2} & h_{3,3} & \cdots & h_{3,m} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{m,m} & h_{m,m} \\ 0 & \cdots & 0 & 0 & h_{m+1,m} \end{bmatrix} \in \mathbb{R}^{(m+1)\times m} \tag{6.3}$$

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:

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

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 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 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:

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

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

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

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.

Part III

Iterative Solvers for Linear Systems

6.4 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:

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

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

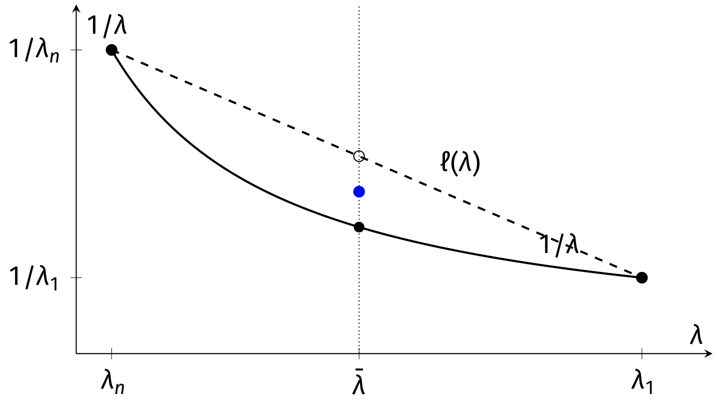
$$\ell(\lambda) = \frac{1}{\lambda_1} + \frac{1}{\lambda_n} - \frac{\lambda}{\lambda_1 \lambda_n}, \qquad \ell(\lambda_1) = \frac{1}{\lambda_1}, \qquad \ell(\lambda_n) = \frac{1}{\lambda_n}$$

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 $0 < \lambda_1 \leq \ldots \leq \lambda_n$, then:

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

Example 11. 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 $\mathcal{K} = \operatorname{span}\{\mathbf{v}\}$, $\mathcal{L} = \operatorname{span}\{\mathbf{w}\}$. 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 12

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

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

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

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

6.4.1 Conjugate gradient (CG) method

Proposition 4

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

Then:

(a) $\langle \mathbf{r}_i, \mathbf{r}_j \rangle = 0$ for $i \neq j$ (residuals are orthogonal)

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

For a) The residual:

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

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

For b) We have

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

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

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

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

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

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

- So: A lower triangular symmetric matrix is diagonal.

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

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

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

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

We know that:

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

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
    $\alpha_j = \frac{\|\mathbf{r}_j\|_2^2}{\langle \mathbf{p}_j, A\mathbf{p}_j \rangle}$
    $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j$
    $\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j A\mathbf{p}_j$
    $\beta_{j+1} = \frac{\|\mathbf{r}_{j+1}\|_2^2}{\|\mathbf{r}_j\|_2^2}$
    $\mathbf{p}_{j+1} = \mathbf{r}_{j+1} + \beta_j \mathbf{p}_j$
    if $\|\mathbf{r}_{j+1}\|_2 <$ 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:

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

We solve the min-max problem:

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

Using Chebyshev polynomials, we get the bound $[-1, 1] \rightarrow [\lambda_{\min}, \lambda_{\max}]$ 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.

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

6.5 Full Orthogonalization Method (FOM)

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

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

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

1. Arnoldi algorithm We can use the Arnoldi algorithm to compute an orthonormal basis for $\mathcal{K}_m(A, \mathbf{r}_0)$. But 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.

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

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

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

so

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

The residual can be computed cheaply from the Arnoldi relation:

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

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

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

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

Algorithm 10 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. ??)
    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)
```

6.6 Generalized Minimum Residual Method (GMRES)

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 ($m < n$):

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

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

6.6.1 QR Factorization Approach

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

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

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

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

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

Let

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

The least squares problem becomes:

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

Then:

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

Then we do QR factorization by Givens rotations:

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

In the Arnoldi process for $k = 1$:

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

For $k = 2$:

$$H_2 = \begin{bmatrix} \tilde{h}_{1,1} & h_{1,2} \\ 0 & h_{2,2} \\ 0 & h_{3,2} \end{bmatrix} \xrightarrow{\Omega_2} \begin{bmatrix} \tilde{h}_{1,1} & \tilde{h}_{1,2} \\ 0 & \tilde{h}_{2,2} \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} \xrightarrow{\Omega_2} \begin{bmatrix} \gamma_1 \\ \tilde{\gamma}_2 \\ \tilde{\gamma}_3 \end{bmatrix}$$

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 11 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
    $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}}$
$V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m] \in \mathbb{R}^{n \times m}$
$V_m^\top V_m = I$
$H_m \in \mathbb{R}^{m \times m}$
$\overline{H}_j \in \mathbb{R}^{(m+1) \times m}$ (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)
```

6.6.2 Convergence of GMRES

We want to estimate the convergence of GMRES. We have:

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

$$\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m \qquad \text{(residual error)}$$

$$\mathbf{x}_\star - \mathbf{x}_m = p_m(A)(\mathbf{x}_\star - \mathbf{x}_0) \quad \begin{cases} p_m \in \mathbb{P}_m \\ p_m(0) = 1 \end{cases}$$

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

$$\mathbf{r}_m = p_m(A)\mathbf{r}_0$$

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

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

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

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

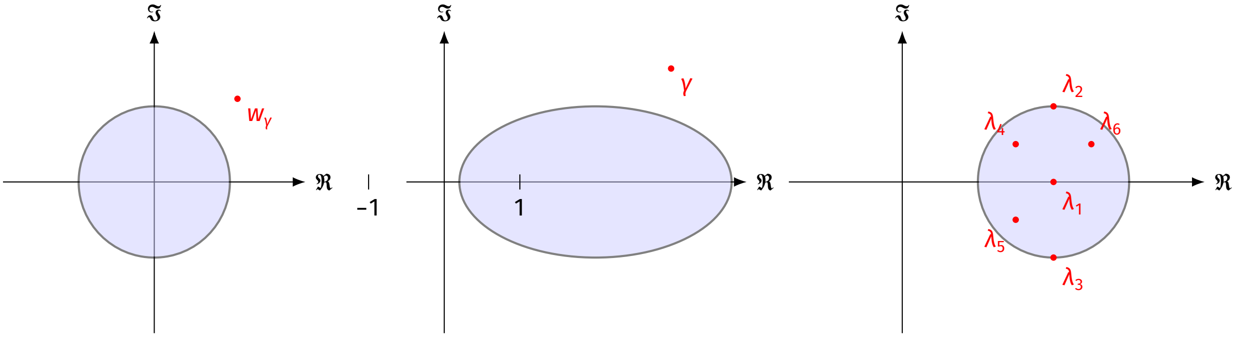
$$\|\mathbf{r}_m\|_2 = \|\mathbf{r}_{m-1}\|_2 = \dots = \|\mathbf{r}_0\|_2$$

Here A may not be diagonalizable, but we assume it is, s.t.

$$A = X\Lambda X^{-1}$$

but where X is not orthogonal anymore.

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



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

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

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

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

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 6.6.1. Elman

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

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

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.