Advanced Linear Algebra

Foundations to Frontiers

Key Concepts and Formulas

Student Name: Tam Tran

Course notes based on the ALAFF textbook by
Robert van de Geijn & Margaret Myers
The University of Texas at Austin

About these notes. This document distills the major theorems, definitions, formulas, and algorithms from the graduate course *Advanced Linear Algebra: Foundations to Frontiers*. The material is organized into the same three parts as the textbook: (I) Orthogonality, (II) Solving Linear Systems, and (III) The Algebraic Eigenvalue Problem.

Notation

$\mathbb{C}, \mathbb{R}$	The complex / real numbers
$\mathbb{C}^m, \mathbb{R}^m$	m-dimensional complex / real column vectors
$\mathbb{C}^{m\times n}$	Complex $m \times n$ matrices
x^T, x^H	Transpose and Hermitian (conjugate) transpose of x
$\bar{\alpha}$	Complex conjugate of scalar α
e_j	The j-th standard basis vector (1 in entry j, 0 elsewhere)
I_n	The $n \times n$ identity matrix
$\mathcal{C}(A), \mathcal{N}(A), \mathcal{R}(A)$	Column space, null space, and row space of A
$\text{rank}(A)$	The rank of A
$\Lambda(A)$	The spectrum (set of eigenvalues) of A
$\rho(A)$	Spectral radius of A
$\kappa(A)$	Condition number of A
$\sigma_i(A), \lambda_i(A)$	Singular values, eigenvalues of A
$\varepsilon_{\text{mach}}$	Machine epsilon (unit roundoff)
$[\cdot]$ or $\text{fl}(\cdot)$	Result computed in floating-point arithmetic

Part I — Orthogonality

1 Norms

1.1 Inner Product and Conjugates

For $x, y \in \mathbb{C}^m$ with $x = (\chi_0, \ldots, \chi_{m-1})^T$ and $y = (\psi_0, \ldots, \psi_{m-1})^T$:

- **Conjugate of vector:** $\bar{x} = (\bar{\chi}_0, \ldots, \bar{\chi}_{m-1})^T$
- **Hermitian transpose:** $x^H = \bar{x}^T$
- **Dot product (inner product):**

$$x^H y = \bar{x}^T y = \overline{x^T} y = \sum_{i=0}^{m-1} \bar{\chi}_i \psi_i$$

1.2 Vector Norms

Definition: Vector Norm

A function $\|\cdot\| : \mathbb{C}^m \to \mathbb{R}$ is a **vector norm** if for all $x, y \in \mathbb{C}^m$ and $\alpha \in \mathbb{C}$:

1. **Positive definite:** $x \neq 0 \Rightarrow \|x\| > 0$
2. **Homogeneous:** $\|\alpha x\| = |\alpha| \, \|x\|$
3. **Triangle inequality:** $\|x + y\| \leq \|x\| + \|y\|$

Key Formula

Common vector norms (for $x \in \mathbb{C}^m$):

$$\|x\|_2 = \sqrt{x^H x} = \sqrt{\textstyle\sum_{i=0}^{m-1} |\chi_i|^2} \qquad \text{(Euclidean / 2-norm)}$$
$$\|x\|_1 = \textstyle\sum_{i=0}^{m-1} |\chi_i| \qquad \text{(1-norm / taxicab)}$$
$$\|x\|_p = \left(\textstyle\sum_{i=0}^{m-1} |\chi_i|^p\right)^{1/p} \qquad (p\text{-norm}, 1 \le p < \infty)$$
$$\|x\|_\infty = \max_{0 \le i < m} |\chi_i| \qquad (\infty\text{-norm})$$
$$\|x\|_\infty = \lim_{p \to \infty} \|x\|_p$$

Theorem: Equivalence of Vector Norms

For any vector norms $\|\cdot\|, |||\cdot||| : \mathbb{C}^m \to \mathbb{R}$ there exist positive constants σ, τ with

$$\sigma \|x\| \le |||x||| \le \tau \|x\| \qquad \forall x \in \mathbb{C}^m.$$

Concrete bounds (with $x \in \mathbb{C}^m$):

$$\|x\|_2 \le \|x\|_1 \le \sqrt{m} \|x\|_2, \qquad \|x\|_\infty \le \|x\|_2 \le \sqrt{m} \|x\|_\infty, \qquad \|x\|_\infty \le \|x\|_1 \le m \|x\|_\infty.$$

1.3 Matrix Norms

Definition: Matrix Norm

A function $\|\cdot\| : \mathbb{C}^{m \times n} \to \mathbb{R}$ is a **matrix norm** if it is positive definite, homogeneous, and satisfies the triangle inequality on $\mathbb{C}^{m \times n}$.

Key Formula

Frobenius norm:

$$\|A\|_F = \sqrt{\textstyle\sum_{i,j} |\alpha_{ij}|^2} = \sqrt{\textstyle\sum_j \|a_j\|_2^2} = \sqrt{\operatorname{tr}(A^H A)} = \sqrt{\textstyle\sum_i \sigma_i^2}.$$

Induced (p-)norm:

$$\|A\|_p = \max_{x \ne 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p = 1} \|Ax\|_p.$$

Special induced norms:

$$\|A\|_1 = \max_{0 \le j < n} \|a_j\|_1 = \left\|A^H\right\|_\infty \qquad \text{(max absolute column sum)}$$
$$\|A\|_\infty = \max_{0 \le i < m} \|\tilde{a}_i\|_1 = \left\|A^H\right\|_1 \qquad \text{(max absolute row sum)}$$
$$\|A\|_2 = \sigma_0(A) = \left\|A^H\right\|_2 \qquad \text{(largest singular value)}$$

Useful bounds: $\|A\|_2 \le \|A\|_F \le \sqrt{r} \|A\|_2$ where $r = \operatorname{rank}(A)$. Also $\|A\|_2 \le \sqrt{\|A\|_1 \|A\|_\infty}$.

Subordinate & Submultiplicative Norms

Definition: Subordinate Matrix Norm

A matrix norm $\|\cdot\|$ is **subordinate** to vector norms $\|\cdot\|_\mu, \|\cdot\|_\nu$ if

$$\|Ax\|_\mu \leq \|A\| \, \|x\|_\nu \qquad \forall x.$$

Induced norms are automatically subordinate.

Definition: Submultiplicative

A consistent norm is **submultiplicative** if $\|AB\| \leq \|A\| \, \|B\|$. The Frobenius norm and all induced norms are submultiplicative.

1.4 Condition Number of a Matrix

Definition: Condition Number

For nonsingular A with subordinate matrix norm,

$$\boxed{\kappa(A) = \|A\| \, \left\|A^{-1}\right\|}$$

The 2-norm condition number satisfies

$$\kappa_2(A) = \frac{\sigma_0(A)}{\sigma_{n-1}(A)}.$$

Key Formula

Sensitivity of $Ax = b$: If $A(x + \delta x) = b + \delta b$, then

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \, \frac{\|\delta b\|}{\|b\|}.$$

Loss of approximately $\log_{10} \kappa(A)$ decimal digits of accuracy.

Practical 2-norm computation (avoiding overflow): With $\mu = \|x\|_\infty$,

$$\|x\|_2 = \mu \sqrt{\sum_{i=0}^{m-1} \left(|\chi_i| / \mu\right)^2}.$$

2 The Singular Value Decomposition (SVD)

2.1 Orthogonality and Unitary Matrices

Definition: Orthogonal Vectors

Vectors $x, y \in \mathbb{C}^m$ are **orthogonal** iff $x^H y = 0$. They are **orthonormal** if additionally $\|x\|_2 = \|y\|_2 = 1$.

For $a, b \in \mathbb{C}^m$ with $a \neq 0$:

- Component of b in the direction of a: $\widehat{b} = \frac{a^H b}{a^H a} a = \frac{a a^H}{a^H a} b$.
- Projection matrix onto Span(a): $P_a = \frac{a a^H}{a^H a}$.
- Projection onto the orthogonal complement: $I - \frac{a a^H}{a^H a}$.
- Component of b orthogonal to a: $b^\perp = b - \widehat{b}$.

Definition: Orthonormal / Unitary Matrix

$Q \in \mathbb{C}^{m \times n}$ $(n \le m)$ is **orthonormal** if $Q^H Q = I_n$. If additionally $m = n$, Q is called **unitary** (or **orthogonal** when real-valued).

Properties of unitary $U \in \mathbb{C}^{m \times m}$:

$$U^H U = U U^H = I, \quad U^{-1} = U^H, \quad \|U\|_2 = 1, \quad \kappa_2(U) = 1.$$

For all $x \in \mathbb{C}^m$ and $A \in \mathbb{C}^{m \times n}$, unitary U, V:

$$\|Ux\|_2 = \|x\|_2 \quad \text{(length preserved)}$$
$$\left\|U^H A V\right\|_2 = \|A\|_2, \qquad \left\|U^H A V\right\|_F = \|A\|_F.$$

Examples of unitary matrices:

- 2D rotation: $\begin{pmatrix} c & -s \\ s & c \end{pmatrix}$ with $c = \cos\theta$, $s = \sin\theta$.
- Reflection (Householder): $H = I - 2uu^H$ where $\|u\|_2 = 1$.
- Permutation matrices.

Change of orthonormal basis. If $U = [u_0 \,|\, \cdots \,|\, u_{m-1}]$ is unitary, then

$$x = \sum_{i=0}^{m-1} (u_i^H x)\, u_i = U U^H x.$$

2.2 The SVD Theorem

Theorem: Singular Value Decomposition

For every $A \in \mathbb{C}^{m \times n}$ there exist unitary $U \in \mathbb{C}^{m \times m}$, unitary $V \in \mathbb{C}^{n \times n}$, and a diagonal $\Sigma \in \mathbb{R}^{m \times n}$ such that

$$\boxed{A = U \Sigma V^H}$$

where $\Sigma = \begin{pmatrix} \Sigma_{TL} & 0 \\ 0 & 0 \end{pmatrix}$ and $\Sigma_{TL} = \operatorname{diag}(\sigma_0, \ldots, \sigma_{r-1})$ with

$$\sigma_0 \ge \sigma_1 \ge \cdots \ge \sigma_{r-1} > 0, \quad r = \operatorname{rank}(A).$$

The values σ_i are the **singular values**; the columns of U and V are the **left and right singular vectors**.

Reduced SVD. If we partition $U = [U_L \,|\, U_R]$, $V = [V_L \,|\, V_R]$ with $U_L \in \mathbb{C}^{m\times r}$, $V_L \in \mathbb{C}^{n\times r}$, then

$$A = U_L \Sigma_{TL} V_L^H = \sum_{i=0}^{r-1} \sigma_i u_i v_i^H \quad \text{(outer-product expansion)}.$$

Key Formula

SVD identities. For $A = U\Sigma V^H$:

- $\|A\|_2 = \sigma_0$ (largest singular value)
- $\|A\|_F^2 = \sum_i \sigma_i^2$
- $\text{rank}(A) = r$ = number of nonzero singular values
- $\mathcal{C}(A) = \mathcal{C}(U_L)$, $\mathcal{N}(A) = \mathcal{C}(V_R)$
- $\mathcal{R}(A) = \mathcal{C}(V_L)$, $\mathcal{N}(A^H) = \mathcal{C}(U_R)$ (left null space)
- $Av_j = \sigma_j u_j$ for $0 \le j < n$ (when $n \le m$)
- If A is square nonsingular: $A^{-1} = V\Sigma^{-1}U^H$, $\kappa_2(A) = \sigma_0/\sigma_{m-1}$
- Left pseudo-inverse (columns linearly independent): $A^\dagger = (A^H A)^{-1}A^H = V\Sigma_{TL}^{-1}U_L^H$

Geometric interpretation. v_0 is the direction of *maximal* magnification; v_{n-1} is the direction of *minimal* magnification. The unit sphere maps to an ellipsoid with semi-axes $\sigma_i u_i$.

2.3 Best Low-Rank Approximation

Theorem: Eckart-Young / Best Rank-k Approximation

Let $A = U\Sigma V^H$ be the SVD of $A \in \mathbb{C}^{m\times n}$. For $0 \le k \le \min(m,n)$, partition $U = [U_L|U_R]$, $V = [V_L|V_R]$ with $U_L \in \mathbb{C}^{m\times k}$, $V_L \in \mathbb{C}^{n\times k}$, and let Σ_{TL} be the leading $k \times k$ block of Σ. Then

$$B = U_L \Sigma_{TL} V_L^H = \sum_{i=0}^{k-1} \sigma_i u_i v_i^H$$

is the closest matrix of rank $\le k$ to A in the 2-norm:

$$\|A - B\|_2 = \min_{\text{rank}(C)\le k} \|A - C\|_2 = \begin{cases} \sigma_k & k < \min(m,n) \\ 0 & \text{otherwise} \end{cases}$$

This underlies low-rank image compression and Principal Component Analysis (PCA).

3 The QR Decomposition

3.1 Gram–Schmidt Orthogonalization

Given linearly independent $\{a_0, \ldots, a_{n-1}\}$ in $\mathbb{C}^m$, Gram–Schmidt produces an orthonormal set $\{q_0, \ldots, q_{n-1}\}$ with $\text{Span}(a_0, \ldots, a_k) = \text{Span}(q_0, \ldots, q_k)$.

Algorithm: Classical Gram–Schmidt (CGS)

For $k = 0, 1, \ldots, n-1$:

$$\begin{aligned} r_{0:k-1,k} &= Q_{0:k-1}^H a_k && \text{(projections)} \\ a_k^\perp &= a_k - Q_{0:k-1}\, r_{0:k-1,k} && \text{(remove projections at once)} \\ \rho_{kk} &= \left\| a_k^\perp \right\|_2 \\ q_k &= a_k^\perp / \rho_{kk} \end{aligned}$$

Algorithm: Modified Gram–Schmidt (MGS)

For $k = 0, 1, \ldots, n-1$:

- For $i = 0, \ldots, k-1$: $\rho_{ik} = q_i^H a_k$; $a_k := a_k - \rho_{ik} q_i$.
- $\rho_{kk} = \|a_k\|_2$, $q_k = a_k / \rho_{kk}$.

MGS is numerically more accurate than CGS.

Theorem: QR Decomposition

Let $A \in \mathbb{C}^{m \times n}$ have linearly independent columns. Then there exists an orthonormal matrix $Q \in \mathbb{C}^{m \times n}$ and an upper triangular matrix $R \in \mathbb{C}^{n \times n}$ such that

$$\boxed{A = QR.}$$

If the diagonal of R is taken to be real and positive, the decomposition is unique.

Cost of Gram–Schmidt: approximately $2mn^2$ flops.

3.2 Householder QR Factorization

Definition: Householder Reflector

Let $u \in \mathbb{C}^n$ with $\|u\|_2 = 1$. Then

$$H = I - 2uu^H$$

is a **Householder transformation** (reflector). It is unitary, Hermitian, and involutory ($H^2 = I$, so $H^{-1} = H$).

Computing u, τ so that $I - uu^H/\tau$ reflects x to $\pm \|x\|_2 e_0$:

$$\begin{aligned} &v = x \mp \|x\|_2 e_0 && \text{(choose sign of } \chi_0 \text{ to avoid cancellation: } v = x + \operatorname{sign}(\chi_1)\, \|x\|_2 e_0) \\ &u = v / \|v\|_2, && \tau = (1 + u_2^H u_2)/2 \text{ (with normalized } u_1 = 1). \end{aligned}$$

Algorithm: Householder QR (sketch)

Repeatedly partition $A \rightarrow \begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix}$ and, for each iteration:

1. Compute Householder vector u_{21} so that $H \begin{pmatrix} \alpha_{11} \\ a_{21} \end{pmatrix} = \begin{pmatrix} \rho_{11} \\ 0 \end{pmatrix}$.

2. Apply H to $\begin{pmatrix} a_{12}^T \\ A_{22} \end{pmatrix}$ via a rank-1 update:

$$w_{12}^T := (a_{12}^T + a_{21}^H A_{22})/\tau_1; \qquad \begin{pmatrix} a_{12}^T \\ A_{22} \end{pmatrix} := \begin{pmatrix} a_{12}^T - w_{12}^T \\ A_{22} - a_{21} w_{12}^T \end{pmatrix}.$$

Cost: approximately $2mn^2 - \frac{2}{3}n^3$ flops.

Forming Q explicitly costs approximately $2mn^2 - \frac{2}{3}n^3$ flops. **Applying Q^H to a vector** costs approximately $4mn - n^2$ flops. The columns of Q produced by Householder are more accurately orthonormal than those from Gram–Schmidt.

4 Linear Least Squares (LLS)

Problem. Given $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$, find $\widehat{x} \in \mathbb{C}^n$ minimizing

$$\|b - A\widehat{x}\|_2 = \min_{x\in\mathbb{C}^n} \|b - Ax\|_2 .$$

4.1 The Four Fundamental Subspaces

- **Column space:** $\mathcal{C}(A) = \{Ax : x \in \mathbb{C}^n\}$, $\dim = r$.
- **Null space:** $\mathcal{N}(A) = \{x : Ax = 0\}$, $\dim = n - r$.
- **Row space:** $\mathcal{R}(A) = \mathcal{C}(A^H) = \{y : y^H = x^H A\}$, $\dim = r$.
- **Left null space:** $\mathcal{N}(A^H) = \{x : x^H A = 0\}$, $\dim = m - r$.

Any $x \in \mathbb{C}^n$ can be uniquely decomposed as $x = x_r + x_n$ with $x_r \in \mathcal{R}(A)$, $x_n \in \mathcal{N}(A)$, and $Ax = Ax_r$. The row space is orthogonal to the null space; the column space is orthogonal to the left null space.

4.2 Method of Normal Equations

The orthogonal projection $\widehat{b} = A\widehat{x}$ of b onto $\mathcal{C}(A)$ satisfies $A^H(b - A\widehat{x}) = 0$:

Key Formula

Normal Equations:

$$\boxed{A^H A\,\widehat{x} = A^H b.}$$

When A has linearly independent columns ($\operatorname{rank}(A) = n$):

$$\widehat{x} = (A^H A)^{-1} A^H b = A^\dagger b, \qquad A^\dagger := (A^H A)^{-1} A^H.$$

Solving normal equations via Cholesky: $B = A^H A$ is Hermitian positive definite (HPD), so $B = LL^H$, then solve $Lz = A^H b$ and $L^H \widehat{x} = z$.

Condition number of LLS. If A has linearly independent columns,

$$\kappa_2(A) = \|A\|_2 \left\|A^\dagger\right\|_2 = \sigma_0/\sigma_{n-1}.$$

The sensitivity to changes in b is:

$$\frac{\|\delta\widehat{x}\|_2}{\|\widehat{x}\|_2} \leq \frac{1}{\cos\theta}\,\kappa_2(A)\,\frac{\|\delta b\|_2}{\|b\|_2},$$

where $\cos\theta = \left\|\widehat{b}\right\|_2 / \|b\|_2$.

Warning: Using normal equations *squares* the condition number, since $\kappa_2(A^H A) = \kappa_2(A)^2$. For ill-conditioned problems, prefer QR or SVD-based solvers.

4.3 Solution via the SVD

If $A = U_L \Sigma_{TL} V_L^H$ is the reduced SVD with $\text{rank}(A) = r$, then the general LLS solution is

$$\widehat{x} = V_L \Sigma_{TL}^{-1} U_L^H b + V_R z_b, \qquad z_b \in \mathbb{C}^{n-r} \text{ arbitrary.}$$

The minimum-norm solution corresponds to $z_b = 0$. The (Moore–Penrose) pseudoinverse is

$$A^\dagger = V_L\,\Sigma_{TL}^{-1}\,U_L^H.$$

4.4 Solution via QR Factorization

If $A = QR$ with A having linearly independent columns, then $\widehat{x}$ solves

$$R\,\widehat{x} = Q^H b \qquad \text{(triangular solve).}$$

For Householder QR factorization of $A \in \mathbb{C}^{m\times n}$:

- Factorization cost: $\approx 2mn^2 - \frac{2}{3}n^3$.
- Compute $y_T = Q^H b$: $\approx 4mn - 2n^2$.
- Triangular solve $R\widehat{x} = y_T$: $\approx n^2$.

Part II — Solving Linear Systems

5 LU and Cholesky Factorizations

5.1 LU Factorization

Definition: LU Factorization

Given $A \in \mathbb{C}^{m\times n}$ with $m \geq n$, an **LU factorization** is $A = LU$ where $L \in \mathbb{C}^{m\times n}$ is unit lower trapezoidal (1s on diagonal) and $U \in \mathbb{C}^{n\times n}$ is upper triangular with no zeros on its diagonal.

Definition: Principal Leading Submatrix

For $k \leq n$, the $k \times k$ **principal leading submatrix** of A is the upper-left $k \times k$ block.

Theorem: Existence of LU Factorization

A square matrix A with linearly independent columns has a (unique) LU factorization if and only if all its principal leading submatrices are nonsingular.

Algorithm: Right-looking LU (FLAME notation)

Repeatedly partition $A \to \begin{pmatrix} A_{TL} & A_{TR} \\ A_{BL} & A_{BR} \end{pmatrix} \to \begin{pmatrix} A_{00} & a_{01} & A_{02} \\ a_{10}^T & \alpha_{11} & a_{12}^T \\ A_{20} & a_{21} & A_{22} \end{pmatrix}$:

$$a_{21} := a_{21}/\alpha_{11}, \qquad A_{22} := A_{22} - a_{21}a_{12}^T.$$

Cost: $\approx mn^2 - \frac{1}{3}n^3$ flops (i.e. $\frac{2}{3}n^3$ when $m = n$).

Solving $Ax = b$ via LU:

$$A = LU \quad \Rightarrow \quad Lz = b \text{ (forward subs.)}, \quad Ux = z \text{ (back subs.)}.$$

Each triangular solve costs $\approx n^2$ flops.

5.2 LU with Pivoting

Definition: Permutation Matrix

Given $p = (\pi_0, \ldots, \pi_{n-1})$ a permutation of $\{0, \ldots, n-1\}$, the permutation matrix is $P(p) = (e_{\pi_0} \mid \cdots \mid e_{\pi_{n-1}})^T$. Then $P^{-1} = P^T$.

Partial pivoting (row exchanges): at step k, choose pivot $\pi_1 = \arg\max_i |\alpha_{i,k}|$ and swap rows. Gives factorization

$$PA = LU.$$

Complete pivoting: choose pivot from both rows and columns; yields $P_1AP_2 = LU$ but is rarely worth the extra cost.

Solving via LU with pivoting: Compute $PA = LU$, then $y = Pb$, solve $Lz = y$, then $Ux = z$.

Gauss transforms.

$$L_k = \begin{pmatrix} I_k & 0 & 0 \\ 0 & 1 & 0 \\ 0 & l_{21} & I \end{pmatrix}, \qquad L_k^{-1} = \begin{pmatrix} I_k & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -l_{21} & I \end{pmatrix}.$$

Iterative refinement. If $\widehat{x}$ approximates $Ax = b$, compute residual $r = b - A\widehat{x}$, solve $A\delta x = r$, update $\widehat{x} := \widehat{x} + \delta x$ to improve accuracy.

5.3 Cholesky Factorization

Definition: Hermitian Positive Definite (HPD)

$A \in \mathbb{C}^{n\times n}$ is **HPD** iff $A^H = A$ and $x^H Ax > 0$ for all nonzero $x \in \mathbb{C}^n$. (When real, it is symmetric positive definite or **SPD**.)

HPD facts.

- B has linearly independent columns $\iff$ $B^H B$ is HPD.
- Diagonal entries of an HPD matrix are real and positive.
- If A is HPD with diagonal blocks A_{TL}, A_{BR}, both blocks are HPD.
- A symmetric matrix is SPD $\iff$ all eigenvalues are positive.

Theorem: Cholesky Factorization

For every HPD matrix A there exists a lower triangular L with positive diagonal such that

$$\boxed{A = LL^H.}$$

The factor L is unique.

Algorithm: Right-looking Cholesky

Partition A as above and iterate:

$$\alpha_{11} := \sqrt{\alpha_{11}}, \quad a_{21} := a_{21}/\alpha_{11}, \quad A_{22} := A_{22} - a_{21}a_{21}^H.$$

Cost: $\approx \frac{1}{3}n^3$ flops (half of LU).

LLS via Cholesky: $B = A^H A$ (cost $\approx mn^2$); factor $B = LL^H$ (cost $\approx n^3/3$); solve two triangular systems (cost $\approx 2n^2$). *Total* $\approx mn^2 + n^3/3$.

6 Numerical Stability

6.1 Floating Point Arithmetic

A floating point number system F consists of values $\chi = \mu\beta^e$ where:

- $\beta = 2$ (base), $\mu = \pm 0.\delta_0\delta_1 \cdots \delta_{t-1}$ (t binary digits), normalized so $\delta_0 = 0$ iff $\mu = 0$.
- Exponent range $-L \le e \le U$.
- **Overflow / underflow:** numbers outside the range become NaN or 0.

Definition: Machine Epsilon

The **machine epsilon** (unit roundoff) $\varepsilon_{\text{mach}}$ is the smallest positive floating-point number χ with $\text{fl}(1 + \chi) > 1$.
Single precision: $\varepsilon_{\text{mach}} \approx 10^{-8}$; double precision: $\varepsilon_{\text{mach}} \approx 10^{-16}$.

For any $\chi \in \mathbb{R}$ stored as floating-point $\check{\chi}$ via rounding:

$$|\delta\chi| \leq \varepsilon_{\text{mach}} |\chi|, \qquad \check{\chi} = \chi(1+\varepsilon), |\varepsilon| \leq \varepsilon_{\text{mach}}.$$

Computation Models

Key Formula

Standard Computational Model (SCM): For $\chi, \psi \in F$ and op $\in \{+, -, *, /\}$:

$$\text{fl}(\chi \text{ op } \psi) = (\chi \text{ op } \psi)(1+\varepsilon), \qquad |\varepsilon| \leq \varepsilon_{\text{mach}}.$$

Alternative Computational Model (ACM):

$$\text{fl}(\chi \text{ op } \psi) = \frac{\chi \text{ op } \psi}{1+\varepsilon}, \qquad |\varepsilon| \leq \varepsilon_{\text{mach}}.$$

6.2 Stability of an Algorithm

Definition: Backward Stable

A computer implementation $\check{f}$ of $f : D \to R$ is **backward stable** if for all $x \in D$ there exists $\check{x}$ near x with $\check{f}(x) = f(\check{x})$.

Conditioning vs. Stability.

- *Conditioning* is a property of the **problem**: how sensitive the output is to input perturbation. Measured by κ.
- *Stability* is a property of the **algorithm**: whether the computed result equals the exact result for a slightly perturbed input.

A stable algorithm on a well-conditioned problem gives an accurate answer. A stable algorithm on an ill-conditioned problem can give a poor answer.

Useful Bound: the γ Factor

Definition: γ_n

For all $n \geq 1$ with $n\varepsilon_{\text{mach}} < 1$:

$$\gamma_n := \frac{n\varepsilon_{\text{mach}}}{1 - n\varepsilon_{\text{mach}}}.$$

Theorem: Lemma

If $|\varepsilon_i| \leq \varepsilon_{\text{mach}}$ for $i = 0, \ldots, n-1$, then $\prod_i (1+\varepsilon_i)^{\pm 1} = 1 + \theta_n$ with $|\theta_n| \leq \gamma_n$.

6.3 Backward Error of Basic Operations

Dot product computed left-to-right $\check{\kappa} = \text{fl}(x^T y)$ satisfies

$$\check{\kappa} = (x + \delta x)^T y \text{ with } |\delta x| \leq \gamma_n |x|.$$

Matrix-vector multiply $y := Ax$ gives $\check{y} = (A + \Delta A)x$ with $|\Delta A| \leq \gamma_n |A|$.

Matrix-matrix multiply $C := AB$ ($A \in \mathbb{R}^{m\times k}$): $\check{C} = AB + \Delta C$ with $|\Delta C| \leq \gamma_k |A| |B|$.

6.4 Linear Solve Backward Error

Theorem: LU + Triangular Solves

Let $A \in \mathbb{R}^{n\times n}$ and $\widetilde{x}$ be computed via LU + triangular solves. Then

$$(A + \Delta A)\,\widetilde{x} = y, \qquad |\Delta A| \leq (3\gamma_n + \gamma_n^2) \left|\check{L}\right| \left|\check{U}\right|.$$

Theorem: Effect on the Solution

For nonsingular A with $(A + \Delta A)(x + \delta x) = y$:

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

Important caution. LU with partial pivoting can have *element growth*; the worst-case matrix is

$$A = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ -1 & 1 & 0 & \cdots & 0 & 1 \\ -1 & -1 & 1 & \cdots & 0 & 1 \\ \vdots & & & \ddots & & \vdots \\ -1 & -1 & \cdots & & 1 & 1 \\ -1 & -1 & \cdots & & -1 & 1 \end{pmatrix},$$

which exhibits exponential growth, but is extremely rare in practice. LU with partial pivoting is *practically* stable.

7 Solving Sparse Linear Systems

A sparse matrix has enough zero entries that exploiting them pays off (Wilkinson).

7.1 Banded Matrices

Definition: Bandwidth / Half-Bandwidth

A symmetric matrix has **half-bandwidth** b if $\alpha_{ij} = 0$ for $|i - j| > b$. Diagonal: $b = 0$. Tridiagonal: $b = 1$.

For SPD matrix of bandwidth b, the Cholesky factor L has the same bandwidth, and the factorization takes $\approx nb^2$ flops.

Cholesky of a tridiagonal SPD matrix (linear in n):

$$\alpha_{i,i} := \sqrt{\alpha_{i,i}}, \quad \alpha_{i+1,i} := \alpha_{i+1,i}/\alpha_{i,i}, \quad \alpha_{i+1,i+1} := \alpha_{i+1,i+1} - \alpha_{i+1,i}^2.$$

7.2 Nested Dissection

For matrices arising from discretized PDEs (e.g., Poisson on a 2D mesh), reordering rows/columns by recursively identifying *separators* reduces **fill-in** during factorization. The smaller the separator (graph cut), the less fill-in.

7.3 Iterative (Splitting) Methods

Write $A = M - N$ and iterate

$$Mx^{(k+1)} = Nx^{(k)} + b, \qquad x^{(k+1)} = M^{-1}(Nx^{(k)} + b).$$

Key Formula

The iteration converges to the solution of $Ax = b$ iff $\rho(M^{-1}N) < 1$ (equivalently $\|M^{-1}N\| < 1$ in some norm).

Writing $A = D - L - U$ (diagonal, strictly lower, strictly upper), common splittings are:

- **Jacobi:** $M = D$, $N = L + U$.

$$x_i^{(k+1)} = \frac{1}{\alpha_{ii}}\left(\beta_i - \sum_{j\neq i} \alpha_{ij}\, x_j^{(k)}\right)$$

- **Gauss–Seidel:** $M = D - L$, $N = U$ (uses already-updated values).
- **SOR:** $M = \frac{1}{\omega}D - L$, $N = \frac{1-\omega}{\omega}D + U$, with relaxation parameter $\omega \in (0, 2)$.
- **SSOR:** Symmetric variant alternating forward and backward sweeps.

8 Descent Methods and the Conjugate Gradient Method

Setting. A is SPD. Solving $Ax = b$ is equivalent to minimizing

$$f(x) = \tfrac{1}{2}x^T Ax - x^T b, \qquad \nabla f(x) = Ax - b.$$

8.1 Descent Algorithm Framework

Given $x^{(0)}$, $r^{(0)} = b - Ax^{(0)}$. For $k = 0, 1, \ldots$:

1. Choose search direction $p^{(k)}$.
2. Compute step length: $\alpha_k = \dfrac{(p^{(k)})^T r^{(k)}}{(p^{(k)})^T Ap^{(k)}}$.
3. Update: $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$.
4. Update residual: $r^{(k+1)} = r^{(k)} - \alpha_k Ap^{(k)}$.

8.2 Steepest Descent

Take $p^{(k)} = r^{(k)} = -\nabla f(x^{(k)})$. *Convergence is slow when* $\kappa_2(A)$ *is large.* A **preconditioner** $M \approx A$ (cheap to invert) transforms to $\widetilde{A} = L_M^{-1} A L_M^{-T}$ with hopefully $\kappa(\widetilde{A}) \ll \kappa(A)$.

8.3 Conjugate Gradient Method

Definition: A-conjugate directions

For SPD A, vectors $p^{(0)}, \ldots, p^{(k-1)} \in \mathbb{R}^n$ are **A-conjugate** if

$$(p^{(j)})^T A\, p^{(i)} = 0 \quad \text{for} \quad i \neq j.$$

If A-conjugate directions are used as descent directions starting from $x^{(0)} = 0$, the method finds the exact solution in at most n steps (in exact arithmetic).

Algorithm: Practical CG (with $x^{(0)} = 0$)

$$
\begin{aligned}
& r := b,\ p := r,\ k := 0 \\
& \textbf{while } r \neq 0 : \\
& \quad q := Ap \\
& \quad \alpha := (r^T r)/(p^T q) \\
& \quad x := x + \alpha p \\
& \quad r_{\text{new}} := r - \alpha q \\
& \quad \gamma := (r_{\text{new}}^T r_{\text{new}})/(r^T r) \\
& \quad p := r_{\text{new}} + \gamma p \\
& \quad r := r_{\text{new}} \\
& \quad k := k + 1
\end{aligned}
$$

Stop when $\|r\|_2 \leq \varepsilon_{\text{mach}} \|b\|_2$ or max iterations exceeded.

Theorem: Properties of CG

With $P^{(k-1)} = [p^{(0)} | \cdots | p^{(k-1)}]$:

- $(P^{(k-1)})^T r^{(k)} = 0$ (residual orthogonal to all prior search directions).
- $\text{Span}(p^{(0)}, \ldots, p^{(k-1)}) = \text{Span}(r^{(0)}, \ldots, r^{(k-1)}) = \text{Span}(b, Ab, \ldots, A^{k-1}b)$.
- Successive residuals $r^{(k)}$ are mutually orthogonal.
- For $k \geq 1$: $p^{(k)} = r^{(k)} - \gamma_k p^{(k-1)}$.

Definition: Krylov Subspace

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

Part III — The Algebraic Eigenvalue Problem

9 Eigenvalues and Eigenvectors

9.1 Basic Definitions

Definition: Eigenvalue, Eigenvector, Eigenpair

For $A \in \mathbb{C}^{m\times m}$, $\lambda \in \mathbb{C}$ and nonzero $x \in \mathbb{C}^m$ are an **eigenpair** of A if

$$Ax = \lambda x.$$

Definition: Spectrum, Spectral Radius

$\Lambda(A) = \{\lambda \in \mathbb{C} : \lambda \text{ is an eigenvalue of } A\}$.
Spectral radius: $\rho(A) = \max_{\lambda\in\Lambda(A)} |\lambda|$.

For $A \in \mathbb{C}^{m\times m}$, the following are equivalent:

- $\lambda \in \Lambda(A)$;
- $\lambda I - A$ is singular;
- $\det(\lambda I - A) = 0$;
- $\mathcal{N}(\lambda I - A)$ is nontrivial.

9.2 Characteristic Polynomial

Definition: Characteristic Polynomial

$p_A(\lambda) = \det(\lambda I - A)$ is a polynomial of degree m whose roots are exactly the eigenvalues of A (counting algebraic multiplicity).

Theorem: Gershgorin Disk Theorem

For $A \in \mathbb{C}^{m\times m}$ define

$$\rho_i(A) = \sum_{j\neq i} |\alpha_{ij}|, \qquad R_i(A) = \{z \in \mathbb{C} : |z - \alpha_{ii}| \le \rho_i(A)\}.$$

Then $\Lambda(A) \subset \bigcup_i R_i(A)$. If a union of k disks is disjoint from the others, it contains exactly k eigenvalues (counting multiplicity).

Useful facts.

- $0 \in \Lambda(A)$ iff A is singular.
- Hermitian matrices: all eigenvalues are real.
- HPD matrices: all eigenvalues are positive (and conversely).

- Eigenvectors of distinct eigenvalues are linearly independent; for Hermitian A, they are orthogonal.
- (λ, x) eigenpair of $A \Leftrightarrow (1/\lambda, x)$ eigenpair of A^{-1} (when A nonsingular).
- $(\lambda - \rho, x)$ is an eigenpair of $A - \rho I$.

9.3 Similarity, Schur, and Spectral Decompositions

Definition: Similarity Transformation

Given nonsingular Y, the transformation $Y^{-1}AY$ is a **similarity transformation**. Two matrices A, B are **similar** if $B = Y^{-1}AY$ for some nonsingular Y.

Similarity preserves eigenvalues: $\Lambda(A) = \Lambda(Y^{-1}AY)$. If $Ax = \lambda x$ then $B(Y^{-1}x) = \lambda(Y^{-1}x)$.

Definition: Unitary Similarity

If Q is unitary, $Q^H AQ$ is a **unitary similarity transformation**. These are preferred in algorithms because they preserve length and do not amplify errors.

Theorem: Schur Decomposition

For every $A \in \mathbb{C}^{m\times m}$ there exist a unitary Q and upper triangular U such that

$$\boxed{A = QUQ^H.}$$

Theorem: Spectral Decomposition

For Hermitian $A \in \mathbb{C}^{m\times m}$ there exist a unitary Q and real diagonal D such that

$$\boxed{A = QDQ^H.}$$

The columns of Q are orthonormal eigenvectors; the diagonal of D holds the (real) eigenvalues.

Diagonalization.

Definition: Diagonalizable

A is **diagonalizable** iff there exists nonsingular X and diagonal D with $X^{-1}AX = D$.

A matrix is diagonalizable iff it has m linearly independent eigenvectors. A matrix lacking m linearly independent eigenvectors is called **defective**.

Geometric multiplicity of $\lambda \in \Lambda(A)$ is $\dim(\mathcal{N}(\lambda I - A))$. Geometric multiplicity $\leq$ algebraic multiplicity, with equality iff the matrix is non-defective at that eigenvalue.

9.4 Jordan Canonical Form

Definition: Jordan Block

$$J_k(\mu) = \begin{pmatrix} \mu & 1 & 0 & \cdots & 0 \\ 0 & \mu & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu & 1 \\ 0 & 0 & \cdots & 0 & \mu \end{pmatrix} \in \mathbb{C}^{k\times k}.$$

Theorem: Jordan Canonical Form

For every $A \in \mathbb{C}^{m\times m}$ there exists nonsingular X such that

$$X^{-1}AX = \text{diag}(J_{m_0}(\lambda_0), J_{m_1}(\lambda_1), \ldots, J_{m_{k-1}}(\lambda_{k-1})).$$

9.5 Block-Triangular Deflation

Theorem: Block Deflation

If $A = \begin{pmatrix} A_{TL} & A_{TR} \\ 0 & A_{BR} \end{pmatrix}$ with A_{TL}, A_{BR} square, then

$$\Lambda(A) = \Lambda(A_{TL}) \cup \Lambda(A_{BR}).$$

9.6 The Power Method and Variants

Algorithm: Power Method

Pick $v^{(0)}$ of unit length. For $k = 0, 1, \ldots$:

$$v^{(k+1)} := Av^{(k)}; \quad v^{(k+1)} := v^{(k+1)} / \left\| v^{(k+1)} \right\|.$$

Converges linearly to the eigenvector x_0 associated with the dominant eigenvalue at rate $|\lambda_1| / |\lambda_0|$.

Algorithm: Inverse Power Method

$$v^{(k+1)} := A^{-1}v^{(k)}; \quad v^{(k+1)} := v^{(k+1)} / \left\| v^{(k+1)} \right\|.$$

Converges to the eigenvector for the smallest (in magnitude) eigenvalue, at rate $|\lambda_{m-1}| / |\lambda_{m-2}|$.

Definition: Rayleigh Quotient

For nonzero $x \in \mathbb{C}^m$:

$$\mu(x) = \frac{x^H Ax}{x^H x}.$$

If x is an eigenvector, $\mu(x)$ is the associated eigenvalue.

Algorithm: Rayleigh Quotient Iteration

Pick $v^{(0)}$ of unit length. For $k = 0, 1, \ldots$:

$$\rho_k = v^{(k)H} A v^{(k)}; \quad v^{(k+1)} := (A - \rho_k I)^{-1} v^{(k)}; \quad v^{(k+1)} := v^{(k+1)} / \left\| v^{(k+1)} \right\|.$$

Quadratic convergence in general; **cubic** for Hermitian A.

9.7 Convergence Rates

For sequence $\alpha_k \to \alpha$:

- **Linear:** $|\alpha_{k+1} - \alpha| \le C\,|\alpha_k - \alpha|$ with $C < 1$.
- **Superlinear:** $C \to 0$.
- **Quadratic:** $|\alpha_{k+1} - \alpha| \le C\,|\alpha_k - \alpha|^2$ (doubles digits per step).
- **Cubic:** $|\alpha_{k+1} - \alpha| \le C\,|\alpha_k - \alpha|^3$ (triples digits per step).

10 Practical Solution of the Hermitian Eigenvalue Problem

10.1 Subspace Iteration

Iterating the power method with multiple vectors and orthonormalizing yields convergence to a basis for the dominant invariant subspace.

Algorithm: Subspace Iteration

$$\begin{aligned}
&\widehat{V} \text{ random } m \times n; \quad (\widehat{V}^{(0)}, R) := \mathrm{QR}(\widehat{V}) \\
&\textbf{for } k = 0, 1, \cdots: \\
&\qquad (\widehat{V}^{(k+1)}, R) := \mathrm{QR}(A\widehat{V}^{(k)}) \\
&\qquad A^{(k+1)} := \widehat{V}^{(k+1)\,H} A \widehat{V}^{(k+1)}
\end{aligned}$$

The j-th column converges linearly at rate $|\lambda_j| / |\lambda_{j-1}|$ (under appropriate assumptions).

10.2 From the Power Method to the QR Algorithm

When subspace iteration is started with $\widehat{V} = I$, it is equivalent to the **QR algorithm**:

Algorithm: Simple QR Algorithm

$V := I$; for $k = 0, 1, \ldots$:

$$(Q, R) := \mathrm{QR}(A); \quad A := RQ; \quad V := VQ.$$

Key identity. After k iterations,

$$A^k = \underbrace{Q^{(0)}Q^{(1)}\cdots Q^{(k)}}_{\text{unitary } V^{(k)}} \cdot \underbrace{R^{(k)}\cdots R^{(1)}R^{(0)}}_{\text{upper triangular}}.$$

10.3 Shifted QR Algorithm

To accelerate convergence, shift A by an estimate of an eigenvalue. The diagonal entry $\alpha^{(k)}_{m-1,m-1}$ is a good shift candidate.

Algorithm: Simple Shifted QR

For $k = 0, 1, \ldots$:

$$\mu_k := \alpha^{(k)}_{m-1,m-1}; \quad (Q, R) := \mathrm{QR}(A - \mu_k I); \quad A := RQ + \mu_k I; \quad V := VQ.$$

The last column inherits cubic convergence (Rayleigh quotient iteration).

10.4 Deflation

When subdiagonal entries are small enough, deflate by treating the matrix as block diagonal. Criterion:

$$\left\| f^{(k)}_{01} \right\|_1 \leq \varepsilon_{\text{mach}} \left(\left| \alpha^{(k)}_{0,0} \right| + \cdots + \left| \alpha^{(k)}_{m-1,m-1} \right| \right).$$

10.5 Reduction to Tridiagonal Form

For a Hermitian matrix, first reduce to tridiagonal form via Householder similarity transformations (a Hermitian rank-2 update at each step):

$$A_{22} := A_{22} - u_{21} w_{21}^H - w_{21} u_{21}^H.$$

Cost: $\approx \frac{4}{3} m^3$ flops. After tridiagonalization, each QR iteration costs $O(m)$ flops (for eigenvalues only) or $O(m^2)$ (with eigenvectors).

10.6 Givens Rotations

Definition: Givens Rotation

A 2×2 unitary matrix

$$G = \begin{pmatrix} \gamma & -\sigma \\ \sigma & \gamma \end{pmatrix}, \quad \gamma^2 + \sigma^2 = 1$$

such that $G^T \begin{pmatrix} \chi_1 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} \|x\|_2 \\ 0 \end{pmatrix}$. So $\gamma = \chi_1 / \|x\|_2$, $\sigma = \chi_2 / \|x\|_2$.

Givens rotations introduce a single zero at a time, useful for tridiagonal and bidiagonal QR steps.

10.7 The Implicit Q Theorem and Francis Step

Theorem: Implicit Q Theorem

Let $A, B \in \mathbb{C}^{m\times m}$ with B upper Hessenberg with positive subdiagonal entries. If $Q^H AQ = B$ for a unitary Q, then Q and B are uniquely determined by A and the first column of Q.

Francis Implicit QR Step. Apply a sequence of Givens' rotations chasing a "bulge" along the subdiagonal of a tridiagonal (or Hessenberg) matrix, effectively performing one shifted QR step without explicitly forming $A - \mu I$.

11 Computing the SVD

11.1 Linking SVD to Spectral Decomposition

Key Formula

If $A = U\Sigma V^H$ is the SVD of $A \in \mathbb{C}^{m\times n}$, then

$$A^H A = V\Sigma^T \Sigma V^H = V(\Sigma^T \Sigma)V^H,$$

i.e., the right singular vectors are eigenvectors of $A^H A$, and the squared singular values are its eigenvalues. Similarly $AA^H = U(\Sigma\Sigma^T)U^H$.

Naive algorithm (not recommended for ill-conditioned A): form $B = A^H A$, compute its spectral decomposition, take V = eigenvectors, Σ = square roots of eigenvalues, $U_L = AV\Sigma^{-1}$.

11.2 Practical SVD Algorithm

1. **Tall-and-skinny preprocessing:** If $A \in \mathbb{C}^{m\times n}$ with $m \gg n$, first compute $A = QR$ and find the SVD of R.
2. **Reduction to bidiagonal form:** Apply Householder transformations from the left and right to obtain $A = U_A B V_A^H$ with B real bidiagonal.
3. **Implicitly shifted bidiagonal QR:** Repeatedly apply pairs of Givens' rotations that chase a bulge through B, maintaining bidiagonal form. The product $B^H B$ is tridiagonal symmetric, and the algorithm implicitly performs the implicitly shifted symmetric tridiagonal QR algorithm on $B^H B$.
4. **Result:** $B \to U_B \Sigma_B V_B^T$. Combine to get $A = (U_A U_B)\Sigma_B (V_A V_B)^H$.

11.3 Jacobi's Method

Definition: Jacobi Rotation

For symmetric 2×2 matrix A there exists a rotation J with $J^T AJ = \text{diag}(\lambda_0, \lambda_1)$.

Jacobi's method applies Jacobi rotations to systematically zero out off-diagonal entries of a symmetric matrix, converging to the diagonalization. Slower than QR-based methods in general, but *very* parallel and high-accuracy.

12 Attaining High Performance

12.1 The Memory Hierarchy and BLAS

Modern CPUs have a multi-level memory hierarchy: registers, L1, L2, L3 cache, main memory. The cost of moving data dominates floating-point arithmetic when the algorithm's arithmetic intensity is low.

BLAS (Basic Linear Algebra Subprograms).

- **Level 1:** Vector-vector operations (dot product, AXPY) — $O(n)$ flops, $O(n)$ data: poor reuse.
- **Level 2:** Matrix-vector operations (GEMV, TRSV) — $O(n^2)$ flops, $O(n^2)$ data: poor reuse.
- **Level 3:** Matrix-matrix operations (GEMM, SYRK, TRSM) — $O(n^3)$ flops, $O(n^2)$ data: **good reuse**.

High-performance dense linear algebra algorithms are cast in terms of *matrix-matrix multiplication* via **blocked algorithms**.

12.2 Blocked Factorizations

Blocked LU partitions A into $b \times b$ blocks. Each iteration computes the LU factorization of a tall panel, then updates trailing blocks via GEMM. The bulk of the work becomes Level-3 BLAS.

Blocked Cholesky / QR / Hessenberg reduction similarly cast updates as rank-b updates or matrix-matrix products. This is how LAPACK and BLIS deliver near-peak performance.

12.3 Cost of Basic Operations (per the textbook appendix)

Operation	Description	Cost (flops)
$y := \alpha x + y$	axpy	$2n$
$\kappa := x^T y$	dot product	$2n$
$y := Ax$	matrix-vector multiply	$2mn$
$A := A + xy^T$	rank-1 update	$2mn$
$C := AB$	matrix-matrix multiply	$2mnk$
LU factorization ($n \times n$)		$\frac{2}{3}n^3$
Cholesky ($n \times n$)		$\frac{1}{3}n^3$
QR ($m \times n$)	Householder	$2mn^2 - \frac{2}{3}n^3$
SVD ($m \times n$)	full	$\sim 4m^2n + 8mn^2 + 9n^3$
Hermitian eigendecomp ($m \times m$)		$\sim \frac{4}{3}m^3 + O(m^3)$

12.4 Catastrophic Cancellation

Subtracting two nearly equal quantities loses leading significant digits, dramatically increasing relative error. Numerically stable algorithms (e.g., Modified Gram-Schmidt, the careful Householder vector formula $v = x + \text{sign}(\chi_1) \|x\|_2 e_0$, the scaled 2-norm computation) are designed to avoid catastrophic cancellation.

Key Takeaways

Key Formula

- **The SVD is the most important factorization in linear algebra.** It reveals rank, condition number, fundamental subspaces, and the best low-rank approximation.
- **Unitary similarity transformations** are the "safe" algorithmic moves: they preserve length and do not amplify errors.
- **Condition number $\kappa(A)$ measures the problem;** stability measures the algorithm. Bad answers can come from either.
- **Prefer QR or SVD over normal equations for ill-conditioned LLS:** normal equations square the condition number.
- **LU with partial pivoting is the workhorse for general linear systems;** Cholesky is twice as fast for HPD systems.
- **Iterative methods** (Jacobi, GS, SOR, CG) are essential for large *sparse* systems where direct methods are too expensive.
- **The Conjugate Gradient method** is the canonical Krylov subspace method for SPD systems, with optimal A-conjugate search directions and finite (exact-arithmetic) termination in $\leq n$ steps.
- **Practical eigenvalue algorithms** all rely on the implicitly shifted QR algorithm, achieving cubic convergence for Hermitian problems.
- **High performance** comes from casting computation in terms of Level-3 BLAS via blocked algorithms.

End of notes.