Advanced Engineering Mathematics

Matrices

Engr. Kaiveni Tom Dagcuta
Computer Engineering

1 Matrix Algebra

1.1 Definition and Notation of Matrices

A **matrix** is a rectangular array of numbers arranged in rows and columns.

Key Concept

An $m \times n$ matrix has:

- m rows
- n columns

The entry in row i and column j is denoted by a_{ij}.

A general $m \times n$ matrix is written as:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Square and Rectangular Matrices

- If $m = n$, the matrix is called **square**.
- If $m \neq n$, it is called **rectangular**.

Example

Consider

$$A = \begin{bmatrix} 1 & 3 & -2 \\ 4 & 0 & 5 \end{bmatrix}.$$

Matrix A has 2 rows and 3 columns, so it is a 2×3 rectangular matrix.
The element in row 2, column 3 is $a_{23} = 5$.

Row and Column Vectors

A matrix with only one row is called a **row vector**:

$$\begin{bmatrix} 3 & -1 & 4 \end{bmatrix}$$

A matrix with only one column is called a **column vector**:

$$\begin{bmatrix} 3 \\ -1 \\ 4 \end{bmatrix}$$

Engineering Note

Matrices allow us to compactly represent systems of equations, data tables, and linear transformations.

1.2 Special Matrices

Zero Matrix

Key Concept

A **zero matrix** is a matrix whose entries are all zero.

$$O = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Identity Matrix

Key Concept

The **identity matrix** I_n is a square matrix with 1's on the main diagonal and 0's elsewhere.

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

It satisfies:

$$AI = IA = A$$

Symmetric Matrix

Key Concept

A matrix A is **symmetric** if

$$A^T = A.$$

Example

$$A = \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$$

Since $a_{12} = a_{21}$, the matrix is symmetric.

1.3 Matrix Addition and Scalar Multiplication

Matrix Addition

Key Concept

Two matrices can be added if and only if they have the same size.

If $A = [a_{ij}]$ and $B = [b_{ij}]$, then

$$A + B = [a_{ij} + b_{ij}]$$

Example

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & -1 \\ 2 & 0 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 6 & 1 \\ 5 & 4 \end{bmatrix}$$

Scalar Multiplication

Key Concept

If c is a scalar and $A = [a_{ij}]$, then

$$cA = [ca_{ij}]$$

Example

$$2 \begin{bmatrix} 1 & -3 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 2 & -6 \\ 8 & 4 \end{bmatrix}$$

1.4 Matrix Multiplication

Key Concept

If A is an $m \times n$ matrix and B is an $n \times p$ matrix, then

$$AB$$

is defined and results in an $m \times p$ matrix.

The entry in row i, column j of AB is

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

Row-by-Column Rule

Multiply rows of A by columns of B.

Example

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 1 & 5 \end{bmatrix}$$

$$AB = \begin{bmatrix} 1(2) + 2(1) & 1(0) + 2(5) \\ 3(2) + 4(1) & 3(0) + 4(5) \end{bmatrix} = \begin{bmatrix} 4 & 10 \\ 10 & 20 \end{bmatrix}$$

Engineering Note

Matrix multiplication is NOT commutative in general:

$$AB \neq BA.$$

1.5 Linear Systems in Matrix Form

Consider the system:

$$2x_1 + x_2 = 5$$
$$4x_1 - 3x_2 = 1$$

We write it in matrix form as:

$$Ax = b$$

where

$$A = \begin{bmatrix} 2 & 1 \\ 4 & -3 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

Key Concept

The equation $Ax = b$ represents a system of linear equations.

- If $b = 0$, the system is **homogeneous**.
- If $b \neq 0$, the system is **nonhomogeneous**.

1.6 Gaussian Elimination

Gaussian elimination is a systematic procedure for solving a system of linear equations written in matrix form:

$$Ax = b.$$

The idea is to transform the augmented matrix $[A \mid b]$ into an upper triangular (or row echelon) form using elementary row operations.

Elementary Row Operations

- $R_i \leftrightarrow R_j$ (Swap two rows)
- $R_i \to cR_i$, $c \neq 0$ (Scale a row)
- $R_i \to R_i + cR_j$ (Row replacement)

Key Concept

Gaussian elimination transforms a system into upper triangular form, after which the solution is obtained by back substitution.

Example 1: 3×3 System with Unique Solution

Solve:

$$\begin{aligned} x + y + z &= 6 \\ 2x - y + z &= 3 \\ x + 2y - z &= 3 \end{aligned}$$

Augmented matrix:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & -1 & 1 & 3 \\ 1 & 2 & -1 & 3 \end{array}\right]$$

Step 1: Eliminate below first pivot

$$R_2 \to R_2 - 2R_1 \quad R_3 \to R_3 - R_1$$

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & -3 & -1 & -9 \\ 0 & 1 & -2 & -3 \end{array}\right]$$

Step 2: Eliminate below second pivot

$$R_3 \to R_3 + \frac{1}{3}R_2$$

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & -3 & -1 & -9 \\ 0 & 0 & -\frac{7}{3} & -6 \end{array}\right]$$

Now perform back substitution:

$$z = \frac{18}{7}, \quad y = \frac{9}{7}, \quad x = \frac{15}{7}.$$

Engineering Note

Since we obtained three pivots (one in each row), the system has a unique solution.

Example 2: System with Infinitely Many Solutions

Consider:

$$x + y + z = 2$$
$$2x + 2y + 2z = 4$$

Augmented matrix:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 4 \end{array}\right]$$

Eliminate:

$$R_2 \to R_2 - 2R_1$$

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

The second row becomes all zeros.
This corresponds to:

$$x + y + z = 2.$$

We may assign free variables:

$$y = s, \quad z = t.$$

Then:

$$x = 2 - s - t.$$

Key Concept

If at least one variable is free (no pivot in its column), the system has infinitely many solutions.

Example 3: Inconsistent System (No Solution)

Consider:

$$x + y = 1$$
$$x + y = 3$$

Augmented matrix:

$$\left[\begin{array}{cc|c} 1 & 1 & 1 \\ 1 & 1 & 3 \end{array}\right]$$

Eliminate:

$$R_2 \to R_2 - R_1$$

$$\left[\begin{array}{cc|c} 1 & 1 & 1 \\ 0 & 0 & 2 \end{array}\right]$$

The second row represents:

$$0 = 2,$$

which is impossible.

Key Concept

If a row reduces to

$$[0\ 0\ \cdots\ 0 \mid c], \quad c \neq 0,$$

the system is inconsistent and has no solution.

Pivoting and Numerical Stability

If a pivot element is zero (or very small), we interchange rows.

Example

$$\left[\begin{array}{cc|c} 0 & 1 & 2 \\ 3 & 4 & 5 \end{array}\right]$$

Swap rows to obtain a nonzero pivot:

$$R_1 \leftrightarrow R_2.$$

Engineering Note

Row swapping to improve numerical stability is called **partial pivoting**. It is essential in computational linear algebra.

Rank and Solution Classification

Key Concept

The number of pivots obtained in row echelon form is called the **rank** of the matrix.

Let:

$$\text{rank}(A) = r.$$

- If $r = n$ (number of variables) $\Rightarrow$ unique solution.
- If $r < n$ and consistent $\Rightarrow$ infinitely many solutions.
- If $\text{rank}(A) < \text{rank}([A|b]) \Rightarrow$ no solution.

Engineering Interpretation

Engineering Note

In engineering applications:

- Unique solution ⇒ well-determined physical system.
- Infinite solutions ⇒ underdetermined system (degrees of freedom).
- No solution ⇒ inconsistent measurements or incompatible constraints.

2 Eigenvalues and Eigenvectors

2.1 Eigenvalues and the Characteristic Equation

Consider a square matrix $A \in \mathbb{R}^{n \times n}$.

Key Concept

A nonzero vector v is called an **eigenvector** of A if

$$Av = \lambda v$$

for some scalar λ.
The scalar λ is called an **eigenvalue** of A.

Interpretation

The equation

$$Av = \lambda v$$

means:

- The transformation A does not change the direction of v.
- It only scales v by the factor λ.

Thus, eigenvectors represent **invariant directions** of a linear transformation.

Deriving the Characteristic Equation

Starting from:

$$Av = \lambda v$$

$$Av - \lambda v = 0$$

$$(A - \lambda I)v = 0$$

For a nontrivial solution ($v \neq 0$) to exist, we require:

$$\det(A - \lambda I) = 0.$$

Key Concept

The equation

$$\det(A - \lambda I) = 0$$

is called the **characteristic equation**.

The resulting polynomial in λ is called the **characteristic polynomial**.

Example: 2×2 Matrix

Let

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}.$$

Compute:

$$A - \lambda I = \begin{bmatrix} 4 - \lambda & 1 \\ 2 & 3 - \lambda \end{bmatrix}.$$

$$\det(A - \lambda I) = (4 - \lambda)(3 - \lambda) - 2.$$

$$= \lambda^2 - 7\lambda + 10.$$

Set equal to zero:

$$\lambda^2 - 7\lambda + 10 = 0.$$

$$(\lambda - 5)(\lambda - 2) = 0.$$

$$\lambda_1 = 5, \quad \lambda_2 = 2.$$

Engineering Note

A 2×2 matrix always has a quadratic characteristic polynomial.

2.2 Eigenvectors and the Structure of the Solution Space

After computing eigenvalues from

$$\det(A - \lambda I) = 0,$$

we determine eigenvectors by solving

$$(A - \lambda I)v = 0.$$

Key Concept

For a fixed eigenvalue λ, the set of all eigenvectors (together with the zero vector) forms the null space of $(A - \lambda I)$.

Thus eigenvectors are obtained by solving a homogeneous linear system.

Geometric Meaning of the Equation

The equation

$$Av = \lambda v$$

means:

- v is scaled but not rotated.
- v is an invariant direction under transformation A.

If $\lambda > 1 \rightarrow$ stretching
If $0 < \lambda < 1 \rightarrow$ contraction
If $\lambda < 0 \rightarrow$ direction reversal

Dimension of the Eigenspace

Key Concept

The set

$$E_\lambda = \{v \neq 0 \mid (A - \lambda I)v = 0\}$$

is called the **eigenspace** corresponding to λ.

Its dimension equals:

$$\dim(\ker(A - \lambda I)).$$

This dimension is crucial for determining diagonalizability.

Example: Full Computation

Let

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}.$$

Eigenvalues:

$$\lambda_1 = 5, \quad \lambda_2 = 2.$$

For $\lambda = 5$:

$$A - 5I = \begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix}.$$

Row reduce:

$$\begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix} \to \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}.$$

Thus:

$$x - y = 0 \Rightarrow x = y.$$

Eigenvectors:

$$v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We always report eigenvectors up to scalar multiples.

Engineering Note

Each eigenvalue produces its own linear subspace. Different eigenspaces intersect only at the zero vector.

2.3 Defective Matrices and Diagonalizability

Diagonalizability

Key Concept

A matrix A is **diagonalizable** if there exists an invertible matrix P such that

$$A = PDP^{-1},$$

where D is a diagonal matrix containing eigenvalues of A.

Key Concept

A matrix is diagonalizable if and only if it has n linearly independent eigenvectors.

This means:

$$\text{Geometric multiplicity} = \text{Algebraic multiplicity} \quad \text{for all eigenvalues.}$$

Example: Defective Matrix

Consider:

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

Characteristic equation:

$$\det(A - \lambda I) = (2 - \lambda)^2.$$

So the eigenvalue is:

$$\lambda = 2 \quad \text{(algebraic multiplicity 2)}.$$

Now compute eigenvectors:

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

Solve:

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

This gives:

$$y = 0.$$

Thus eigenvectors are:

$$v = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

There is only one independent eigenvector.

Key Concept

Since geometric multiplicity (1) is less than algebraic multiplicity (2), the matrix is **defective** and not diagonalizable.

2.4 Complex Eigenvalues

Eigenvalues of real matrices may be complex.

Example

Consider:

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

Compute:

$$\det(A - \lambda I) = \begin{vmatrix} -\lambda & -1 \\ 1 & -\lambda \end{vmatrix} = \lambda^2 + 1.$$

Solve:

$$\lambda^2 + 1 = 0$$

$$\lambda = \pm i.$$

Key Concept

Complex eigenvalues of real matrices occur in conjugate pairs.

Geometric Meaning

The matrix

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

represents a 90° rotation.

- No real invariant directions exist.
- The transformation corresponds to rotation.

Engineering Note

If eigenvalues are:

- $\lambda = a \pm bi$, the transformation represents rotation combined with scaling.
- $|\lambda| = 1$, pure rotation.

2.5 Eigenvalues in Applications and Stability

Eigenvalues determine the long-term behavior of systems.

Discrete Dynamical Systems

Consider:

$$x_{k+1} = Ax_k.$$

If A has eigenvalues $\lambda_1, \lambda_2, \ldots$:

- If $|\lambda_i| < 1$, solutions decay.
- If $|\lambda_i| > 1$, solutions grow.
- If $|\lambda_i| = 1$, oscillatory behavior.

Key Concept

Stability of the system depends on the magnitude of eigenvalues.

Vibration and Natural Frequencies

In mechanical systems:

$$Kx = \lambda Mx,$$

where:

- K = stiffness matrix
- M = mass matrix

Eigenvalues determine natural frequencies:

$$\omega = \sqrt{\lambda}.$$

Engineering Insight

Engineering Note

Eigenvalues appear in:

- Structural analysis
- Control systems
- Signal processing
- Population models
- Machine learning (PCA)

Summary of Eigenvalue Behavior

Eigenvalue	Behavior
$\|\lambda\| < 1$	$Decay$
$\|\lambda\| > 1$	$Growth$
$\lambda < 0$	$Directionreversal$
$\lambda = a \pm bi$	$Rotation - scaling$

3 Powers of Matrices

3.1 Definition and Basic Properties

Key Concept

If A is a square matrix, the nth power of A is defined as

$$A^n = \underbrace{AAA\cdots A}_{n \text{ times}}.$$

We define:

$$A^0 = I.$$

Important Remarks

- Matrix powers are defined only for square matrices.
- In general,
$$(A+B)^n \neq A^n + B^n.$$
- Matrix multiplication is not commutative, so order matters.

Example

Let

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

Compute A^2:

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

Compute A^3:

$$A^3 = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}.$$

Observe the pattern:

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

3.2 Powers of Special Matrices

1. Diagonal Matrices

Key Concept

If

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

then

$$D^n = \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix}.$$

Powers of diagonal matrices are computed by raising each diagonal entry to the nth power.

2. Identity Matrix

$$I^n = I.$$

3. Nilpotent Matrix

Key Concept

A matrix N is **nilpotent** if

$$N^k = 0$$

for some positive integer k.

Example

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

$$N^2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Thus N is nilpotent of index 2.

3.3 Computing Powers Using Eigenvalues

If a matrix is diagonalizable, powers can be computed efficiently.

Key Concept

If

$$A = PDP^{-1},$$

then

$$A^n = PD^nP^{-1}.$$

Since D is diagonal:

$$D^n = \begin{bmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{bmatrix}.$$

Example

Let

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}.$$

Suppose eigenvalues are:

$$\lambda_1 = 5, \quad \lambda_2 = 2.$$

Then

$$A = PDP^{-1} \quad \Rightarrow \quad A^n = P \begin{bmatrix} 5^n & 0 \\ 0 & 2^n \end{bmatrix} P^{-1}.$$

Engineering Note

This avoids repeated multiplication and is essential for large powers.

3.4 Applications to Recurrence Relations and Dynamical Systems

1. Discrete Dynamical Systems

Consider:

$$x_{k+1} = Ax_k.$$

Iterating:

$$x_k = A^k x_0.$$

Thus understanding A^k determines the long-term behavior.

2. Example: Fibonacci Sequence

The Fibonacci recurrence:

$$F_{n+1} = F_n + F_{n-1}$$

can be written as:

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix}.$$

Thus:

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = A^{n-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Using eigenvalues:

$$\lambda = \frac{1 \pm \sqrt{5}}{2},$$

we obtain Binet's formula.

3. Stability Insight

Key Concept

If all eigenvalues satisfy $|\lambda| < 1$, then

$$A^n \to 0 \quad \text{as} \quad n \to \infty.$$

Engineering Note

Matrix powers allow prediction of long-term behavior in:

- Population models
- Control systems
- Markov chains
- Economic growth models

4 Orthogonal Matrices

4.1 Definition of Orthogonal Matrices

Key Concept

A square matrix $Q \in \mathbb{R}^{n \times n}$ is called **orthogonal** if

$$Q^T Q = I.$$

Equivalently,

$$Q^{-1} = Q^T.$$

Thus the transpose of an orthogonal matrix is its inverse.

Interpretation

If Q is orthogonal, then:

$$Q^T Q = I \Rightarrow Q^T = Q^{-1}.$$

This means:

- Applying Q and then Q^T returns the original vector.
- Orthogonal matrices represent reversible transformations.

Example

Consider

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Compute:

$$Q^T Q = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = I.$$

Thus rotation matrices are orthogonal.

Engineering Note

Orthogonal matrices generalize rotations and reflections in $\mathbb{R}^n$.

4.2 Fundamental Properties of Orthogonal Matrices

1. Length Preservation

Let Q be orthogonal and $x \in \mathbb{R}^n$.

$$\|Qx\|^2 = (Qx)^T(Qx).$$

$$= x^T Q^T Q x.$$

$$= x^T I x = \|x\|^2.$$

Key Concept

Orthogonal transformations preserve vector length.

2. Inner Product Preservation

$$(Qx)^T(Qy) = x^T Q^T Q y = x^T y.$$

Key Concept

Orthogonal matrices preserve inner products.

Thus:

- Angles are preserved.
- Orthogonality is preserved.

3. Determinant

Since

$$Q^T Q = I,$$

taking determinants:

$$\det(Q)^2 = 1.$$

$$\det(Q) = \pm 1.$$

Key Concept

If $\det(Q) = 1$, Q is a rotation. If $\det(Q) = -1$, Q includes a reflection.

4. Stability Advantage

Orthogonal matrices are numerically stable because:

$$\|Qx\| = \|x\|.$$

No amplification of rounding errors occurs.

4.3 Orthonormal Columns and Matrix Structure

Key Concept

A matrix Q is orthogonal if and only if its columns form an orthonormal set.

That means:

$$q_i^T q_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Proof Sketch

Let

$$Q = [q_1 \; q_2 \; \dots \; q_n].$$

Then:

$$Q^T Q = \begin{bmatrix} q_1^T \\ q_2^T \\ \vdots \\ q_n^T \end{bmatrix} [q_1 \; q_2 \; \dots \; q_n].$$

The (i, j) entry equals:

$$q_i^T q_j.$$

Thus:

$$Q^T Q = I$$

if and only if the columns are orthonormal.

Geometric Meaning

The columns of Q form a rotated coordinate system.

Applying Q transforms coordinates from the standard basis to this new orthonormal basis.

4.4 Orthogonal Diagonalization and the Spectral Theorem

Orthogonal matrices play a central role in diagonalizing symmetric matrices.

Key Concept

Spectral Theorem (Real Case)
If A is a real symmetric matrix, then:

$$A = QDQ^T,$$

where:

- Q is orthogonal,
- D is diagonal,
- The diagonal entries are eigenvalues of A.

Why Symmetry Matters

If $A = A^T$, then:

- All eigenvalues are real.
- Eigenvectors corresponding to distinct eigenvalues are orthogonal.

Energy Interpretation

For symmetric A:

$$x^T Ax$$

represents a quadratic form.

Orthogonal diagonalization transforms it into:

$$x^T Ax = y^T Dy.$$

This simplifies:

- Principal axis transformation
- Optimization problems
- PCA

Engineering Importance

Engineering Note

Orthogonal diagonalization appears in:

- Vibration mode analysis
- Structural mechanics
- Principal component analysis
- Image compression
- Least squares methods

5 Diagonalization

5.1 Definition of Diagonalization

Key Concept

A square matrix A is **diagonalizable** if there exists an invertible matrix P such that

$$A = PDP^{-1},$$

where D is a diagonal matrix.

The diagonal entries of D are the eigenvalues of A.

Meaning of the Decomposition

If

$$A = PDP^{-1},$$

then:

$$P^{-1}AP = D.$$

This means:

- In a new coordinate system defined by P,
- The transformation A acts as simple scaling.

Diagonal matrices are easy to compute with:

$$D^n = \begin{bmatrix} \lambda_1^n & 0 & \cdots & 0 \\ 0 & \lambda_2^n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n^n \end{bmatrix}.$$

Interpretation

Diagonalization transforms a complicated linear transformation into independent scaling along eigenvector directions.

5.2 Conditions for Diagonalizability

Key Concept

An $n \times n$ matrix is diagonalizable if and only if it has n linearly independent eigenvectors.

This is equivalent to:

$$\text{Geometric multiplicity} = \text{Algebraic multiplicity}$$

for every eigenvalue.

Important Cases

1. Distinct Eigenvalues

Key Concept

If a matrix has n distinct eigenvalues, it is automatically diagonalizable.

2. Repeated Eigenvalues

Repeated eigenvalues require checking the dimension of eigenspaces.

3. Symmetric Matrices

Key Concept

Every real symmetric matrix is diagonalizable by an orthogonal matrix.

That is,

$$A = QDQ^T.$$

When Diagonalization Fails

If geometric multiplicity is strictly less than algebraic multiplicity, the matrix is defective and cannot be diagonalized.

5.3 Procedure for Diagonalization

Step-by-Step Method

1. Find eigenvalues from $\det(A - \lambda I) = 0$.
2. Compute eigenvectors for each eigenvalue.
3. Form matrix P using eigenvectors as columns.
4. Form D using eigenvalues along the diagonal.

Full Example with Verification

Let

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}.$$

We previously computed:

$$\lambda_1 = 5, \quad \lambda_2 = 2.$$

Eigenvectors:

$$v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}.$$

Construct:

$$P = \begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix}, \qquad D = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}.$$

Compute determinant of P:

$$\det(P) = (1)(-2) - (1)(1) = -3.$$

Thus,

$$P^{-1} = \frac{1}{-3} \begin{bmatrix} -2 & -1 \\ -1 & 1 \end{bmatrix}.$$

Verification of $A = PDP^{-1}$

First compute PD:

$$PD = \begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 5 & -4 \end{bmatrix}.$$

Now multiply by P^{-1}:

$$PDP^{-1} = \begin{bmatrix} 5 & 2 \\ 5 & -4 \end{bmatrix} \frac{1}{-3} \begin{bmatrix} -2 & -1 \\ -1 & 1 \end{bmatrix}.$$

Compute product inside first:

$$= \frac{1}{-3} \begin{bmatrix} (5)(-2) + (2)(-1) & (5)(-1) + (2)(1) \\ (5)(-2) + (-4)(-1) & (5)(-1) + (-4)(1) \end{bmatrix}.$$

$$= \frac{1}{-3} \begin{bmatrix} -10 - 2 & -5 + 2 \\ -10 + 4 & -5 - 4 \end{bmatrix}.$$

$$= \frac{1}{-3} \begin{bmatrix} -12 & -3 \\ -6 & -9 \end{bmatrix}.$$

$$= \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}.$$

Thus,

$$A = PDP^{-1}.$$

Key Concept

We have verified that A is diagonalizable.

Interpretation

Diagonalization shows that:

$$A = PDP^{-1}$$

means:

1. Transform coordinates using P^{-1}.
2. Apply independent scalings (5 and 2) via D.
3. Transform back using P.

The transformation A acts as pure scaling in the eigenvector directions.

Power Computation

Now powers are simple:

$$A^n = PD^nP^{-1}.$$

$$= P \begin{bmatrix} 5^n & 0 \\ 0 & 2^n \end{bmatrix} P^{-1}.$$

Engineering Note

Instead of multiplying A repeatedly, we raise eigenvalues to the nth power. This is the primary computational advantage of diagonalization.

5.4 Applications and Structural Meaning

1. Matrix Powers

Diagonalization simplifies:

$$A^n.$$

2. Differential Equations

For

$$x' = Ax,$$

if

$$A = PDP^{-1},$$

then:

$$x(t) = Pe^{Dt}P^{-1}x_0.$$

Since:

$$e^{Dt} = \begin{bmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{bmatrix}.$$

3. Decoupling of Systems

Diagonalization decouples coupled systems into independent scalar equations.

4. Geometric Interpretation

The transformation:

$$A = PDP^{-1}$$

means:

1. Change basis using P^{-1}.
2. Apply independent scaling using D.
3. Transform back using P.

Key Concept

Diagonalization reveals the intrinsic structure of a linear transformation.

Engineering Importance

Engineering Note

Diagonalization is fundamental in:

- Vibrations and modal analysis
- Control theory
- Principal Component Analysis (PCA)
- Quantum mechanics
- Signal processing

6 LU-Factorization

6.1 Motivation and Idea

When solving a system

$$Ax = b,$$

Gaussian elimination transforms A into an upper triangular matrix.

Key Concept

LU-factorization records the elimination process as a matrix decomposition:

$$A = LU,$$

where

- L is lower triangular,
- U is upper triangular.

Why Is This Useful?

If we factor:

$$A = LU,$$

then solving

$$Ax = b$$

becomes:

$$LUx = b.$$

Let:

$$Ux = y.$$

Then solve in two steps:

1. $Ly = b$ (forward substitution)
2. $Ux = y$ (back substitution)

Engineering Note

If multiple right-hand sides b are given, LU-factorization avoids repeating elimination.

6.2 LU Decomposition (No Pivoting)

Suppose we perform Gaussian elimination on:

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 7 \end{bmatrix}.$$

Step 1: Elimination

We eliminate the entry below the pivot a_{11}.

The multiplier is:

$$m_{21} = \frac{a_{21}}{a_{11}} = \frac{4}{2} = 2.$$

Apply the row operation:

$$R_2 \to R_2 - 2R_1.$$

This produces the upper triangular matrix:

$$U = \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix}.$$

How Does This Produce LU?

Gaussian elimination can be written as matrix multiplication.

The row operation

$$R_2 \to R_2 - 2R_1$$

is equivalent to multiplying A on the left by the elimination matrix:

$$E = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}.$$

Indeed,

$$EA = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & 7 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix} = U.$$

Thus,

$$EA = U.$$

Solving for A

From

$$EA = U,$$

we solve for A:

$$A = E^{-1}U.$$

Now compute the inverse of E:

$$E^{-1} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}.$$

Notice:

- The inverse simply changes -2 to $+2$.
- This number 2 is exactly the elimination multiplier.

We define:

$$L = E^{-1}.$$

Thus:

$$A = LU.$$

Why Does L Look Like This?

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

Properties:

- Ones on the diagonal
- Elimination multipliers below the diagonal
- Zeros above the diagonal

Key Concept

Matrix L stores the elimination multipliers. Matrix U is the resulting upper triangular matrix.

General 3×3 Structure

For a 3×3 matrix:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}.$$

Where:

- $\ell_{21}, \ell_{31}, \ell_{32}$ are elimination multipliers.
- U contains the final pivoted matrix after elimination.

Big Picture Insight

Gaussian elimination can always be viewed as:

$$E_k \cdots E_2 E_1 A = U.$$

Taking inverses:

$$A = E_1^{-1} E_2^{-1} \cdots E_k^{-1} U.$$

The product of all inverse elimination matrices forms L.

$$A = LU.$$

Engineering Note

LU decomposition is simply Gaussian elimination written in matrix form.

6.3 Solving Systems Using LU

Given:

$$A = LU, \quad Ax = b,$$

we solve:

$$LUx = b.$$

Let:

$$Ux = y.$$

First solve:

$$Ly = b.$$

Since L is lower triangular, use forward substitution.

Example

Let:

$$L = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 11 \end{bmatrix}.$$

Step 1: Solve $Ly = b$.

$$\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 11 \end{bmatrix}.$$

From first row:

$$y_1 = 5.$$

Second row:

$$2(5) + y_2 = 11 \Rightarrow y_2 = 1.$$

Thus:

$$y = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

Step 2: Solve $Ux = y$.

$$\begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

From second row:

$$x_2 = 1.$$

First row:

$$2x_1 + 3(1) = 5 \Rightarrow x_1 = 1.$$

Key Concept

LU reduces solving $Ax = b$ to two simpler triangular systems.

6.4 Pivoting and Numerical Stability

LU without pivoting may fail if a pivot is zero.

Example

$$A = \begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix}.$$

The first pivot is zero.
We interchange rows:

$$PA = \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix}.$$

Key Concept

With pivoting:

$$PA = LU.$$

P is a permutation matrix.

Why Pivoting Is Important

- Avoids division by zero.
- Reduces round-off error.
- Improves numerical stability.

Partial Pivoting

Choose the largest absolute value in the pivot column.

This leads to:

$$PA = LU.$$

Engineering Note

In practical computation (MATLAB, NumPy), LU decomposition always uses pivoting.

7 Cryptography

7.1 Hill Cipher Encryption (Matrix-Based Encryption)

The **Hill Cipher** is a classical block cipher that uses matrix multiplication over modular arithmetic. It is a beautiful application of linear algebra, even though it is not secure by modern standards.

Alphabet Mapping and Modular Arithmetic

We map letters to integers:

$$A = 0,\ B = 1,\ \ldots,\ Z = 25.$$

All computations are done modulo 26:

$$y \equiv Kx \pmod{26}.$$

Key Concept

In the Hill cipher, encryption is a linear transformation over the ring $\mathbb{Z}_{26}$.

Handling Negative Values

When reducing modulo 26, negative numbers are converted by adding 26 repeatedly:

$$-3 \equiv 23 \pmod{26}, \quad -29 \equiv -29 + 26 = -3 \equiv 23 \pmod{26}.$$

Block Size and Padding

Choose an $n \times n$ key matrix K. The plaintext is grouped into blocks of size n.

- If the message length is not a multiple of n, we add padding (commonly $X = 23$).
- Each block becomes a column vector $x \in \mathbb{Z}_{26}^n$.

Encryption for each block is:

$$y = Kx \pmod{26}.$$

Key Matrix Requirement

Not every matrix works as a key. We must be able to decrypt.

Key Concept

The key matrix K must be invertible modulo 26, otherwise decryption is impossible.

(We will formalize this condition in the next subsection.)

Example 1 (2×2): Digraph Encryption

Let

$$K = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}.$$

Encrypt the plaintext block "HI".

$$H = 7, \quad I = 8 \quad \Rightarrow \quad x = \begin{bmatrix} 7 \\ 8 \end{bmatrix}.$$

Compute:

$$y = Kx = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 45 \\ 54 \end{bmatrix}.$$

Reduce mod 26:

$$45 \equiv 19, \quad 54 \equiv 2 \pmod{26}.$$

Thus,

$$y = \begin{bmatrix} 19 \\ 2 \end{bmatrix} \Rightarrow (T, C).$$

Ciphertext is **TC**.

Example 2 (2×2): Multi-block Encryption with Padding

Let the same key matrix K be used. Encrypt "HELP".

Convert letters:

$$H = 7,\ E = 4,\ L = 11,\ P = 15.$$

Block into pairs (since $n = 2$):

$$x_1 = \begin{bmatrix} 7 \\ 4 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 11 \\ 15 \end{bmatrix}.$$

Encrypt first block:

$$y_1 = Kx_1 = \begin{bmatrix} 3(7) + 3(4) \\ 2(7) + 5(4) \end{bmatrix} = \begin{bmatrix} 33 \\ 34 \end{bmatrix} \equiv \begin{bmatrix} 7 \\ 8 \end{bmatrix} \pmod{26}.$$

So $y_1 \Rightarrow (H, I)$.

Encrypt second block:

$$y_2 = Kx_2 = \begin{bmatrix} 3(11) + 3(15) \\ 2(11) + 5(15) \end{bmatrix} = \begin{bmatrix} 78 \\ 97 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 19 \end{bmatrix} \pmod{26}.$$

So $y_2 \Rightarrow (A, T)$.

Ciphertext: **HIAT**.

Engineering Note

This example shows that each plaintext block is transformed independently using the same linear map K in $\mathbb{Z}_{26}^n$.

Example 3 (3×3): Trigraph Encryption (More Complex)

Let the key matrix be:

$$K = \begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}.$$

Encrypt the plaintext "ACT" (a classic trigraph example).

$$A = 0, \quad C = 2, \quad T = 19 \Rightarrow x = \begin{bmatrix} 0 \\ 2 \\ 19 \end{bmatrix}.$$

Compute $y = Kx$:

$$y = \begin{bmatrix} 6(0) + 24(2) + 1(19) \\ 13(0) + 16(2) + 10(19) \\ 20(0) + 17(2) + 15(19) \end{bmatrix} = \begin{bmatrix} 67 \\ 222 \\ 319 \end{bmatrix}.$$

Reduce mod 26:

$$67 \equiv 15, \quad 222 \equiv 14, \quad 319 \equiv 7 \pmod{26}.$$

Thus,

$$y = \begin{bmatrix} 15 \\ 14 \\ 7 \end{bmatrix} \Rightarrow (P, O, H).$$

Ciphertext is **POH**.

Key Concept

A larger block size n increases diffusion: each ciphertext letter depends on multiple plaintext letters.

Practical Implementation Notes

- Always choose a key matrix K that is invertible modulo 26.

- Agree on block size and padding rule (e.g., pad with X).
- Encryption is fast: it is just matrix multiplication mod 26.

7.2 Decryption and Modular Matrix Inverses

To decrypt Hill cipher ciphertext, we need the inverse of the key matrix *in modular arithmetic*.

If encryption is:

$$y \equiv Kx \pmod{26},$$

then decryption is:

$$x \equiv K^{-1}y \pmod{26}.$$

Key Concept

Decryption is only possible if K has a modular inverse modulo 26.

Invertibility Condition (Key Feasibility)

In ordinary linear algebra, K is invertible if $\det(K) \neq 0$.

In modular arithmetic, the correct condition is:

Key Concept

A matrix K is invertible modulo 26 if and only if

$$\gcd(\det K, 26) = 1.$$

Why?

Because we need $(\det K)^{-1}$ modulo 26 to compute K^{-1}, and a modular inverse exists only when the number is relatively prime to 26.

Engineering Note

If $\gcd(\det K, 26) \neq 1$, then $\det K$ has no modular inverse, so K^{-1} does not exist in $\mathbb{Z}_{26}$.

Modular Inverse of an Integer (Extended Euclidean Idea)

To find $a^{-1} \pmod{26}$, we want a number t such that:

$$at \equiv 1 \pmod{26}.$$

This is equivalent to:

$$at - 26k = 1$$

for some integer k. This is what the extended Euclidean algorithm finds.

Example

Find $9^{-1} \pmod{26}$.
We test multiples:

$$9 \cdot 3 = 27 \equiv 1 \pmod{26}.$$

Thus:

$$9^{-1} \equiv 3 \pmod{26}.$$

Inverse of a 2×2 Matrix Modulo 26

Let

$$K = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

In ordinary algebra:

$$K^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

In modular arithmetic we replace division by multiplication with the modular inverse of the determinant:

Key Concept

$$K^{-1} \equiv (\det K)^{-1} \operatorname{adj}(K) \pmod{26}.$$

where:

$$\det K = ad - bc, \quad \operatorname{adj}(K) = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

Example: Full Decryption for the 2×2 Key

Use the key:

$$K = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}, \quad \det K = 9.$$

We already found:

$$9^{-1} \equiv 3 \pmod{26}.$$

Adjugate matrix:

$$\text{adj}(K) = \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix}.$$

Now compute K^{-1}:

$$K^{-1} \equiv 3 \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix} = \begin{bmatrix} 15 & -9 \\ -6 & 9 \end{bmatrix} \pmod{26}.$$

Reduce each entry mod 26:

$$15 \equiv 15, \quad -9 \equiv 17, \quad -6 \equiv 20, \quad 9 \equiv 9.$$

Thus:

$$K^{-1} \equiv \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix} \pmod{26}.$$

Example

Decrypt the ciphertext block "TC".

$$T = 19, \quad C = 2 \Rightarrow y = \begin{bmatrix} 19 \\ 2 \end{bmatrix}.$$

Compute:

$$x \equiv K^{-1}y = \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix} \begin{bmatrix} 19 \\ 2 \end{bmatrix} = \begin{bmatrix} 15(19) + 17(2) \\ 20(19) + 9(2) \end{bmatrix} = \begin{bmatrix} 319 \\ 398 \end{bmatrix} \pmod{26}.$$

Reduce:

$$319 \equiv 7, \quad 398 \equiv 8 \pmod{26}.$$

So:

$$x = \begin{bmatrix} 7 \\ 8 \end{bmatrix} \Rightarrow (H, I).$$

We recovered the plaintext **HI**.

How About 3×3 Inverses? (Important Insight)

For a 3×3 key matrix, the same principle holds:

$$K^{-1} \equiv (\det K)^{-1} \operatorname{adj}(K) \pmod{26}.$$

But now:

- $\det K$ is computed from the 3×3 determinant formula,
- $\operatorname{adj}(K)$ requires cofactors (a full cofactor matrix transpose),
- then multiply by $(\det K)^{-1}$ modulo 26.

Engineering Note

Conceptually, decryption is still "multiply by the inverse," but the computation is heavier for larger block sizes. That is why computers are used in practice.

Common Student Pitfalls

- Forgetting to reduce values modulo 26 at the end.
- Mishandling negative entries (always convert to 0–25).
- Choosing a key matrix with $\gcd(\det K, 26) \neq 1$ (non-invertible).

7.3 Security Analysis and Structural Weaknesses

Although mathematically elegant, the Hill cipher is fundamentally insecure. Its weakness comes from the fact that encryption is a **linear transformation**.

Linearity of the Cipher

Encryption is defined as:

$$y \equiv Kx \pmod{26}.$$

This is a linear map over $\mathbb{Z}_{26}^n$.

Key Concept

Because encryption is linear, the entire cipher can be described as solving a system of linear equations.

This linearity makes the system vulnerable.

Known-Plaintext Attack (Core Weakness)

Suppose an attacker knows:

$$x_1 \mapsto y_1, \quad x_2 \mapsto y_2, \quad \ldots, \quad x_n \mapsto y_n.$$

Stack these into matrices:

$$X = [x_1\ x_2\ \ldots\ x_n], \quad Y = [y_1\ y_2\ \ldots\ y_n].$$

From encryption:

$$Y \equiv KX \pmod{26}.$$

If X is invertible modulo 26, then:

$$K \equiv YX^{-1} \pmod{26}.$$

Key Concept

If an attacker obtains n independent plaintext–ciphertext pairs (for an $n \times n$ key), they can recover the key matrix exactly.

Concrete Attack Example (2×2 Key)

Suppose the attacker knows:

Plaintext blocks:

$$x_1 = \begin{bmatrix} 7 \\ 8 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 4 \\ 11 \end{bmatrix}.$$

Ciphertext blocks:

$$y_1 = \begin{bmatrix} 19 \\ 2 \end{bmatrix}, \quad y_2 = \begin{bmatrix} 7 \\ 8 \end{bmatrix}.$$

Construct matrices:

$$X = \begin{bmatrix} 7 & 4 \\ 8 & 11 \end{bmatrix}, \quad Y = \begin{bmatrix} 19 & 7 \\ 2 & 8 \end{bmatrix}.$$

Then:

$$K \equiv YX^{-1} \pmod{26}.$$

The attacker simply computes a matrix inverse modulo 26 — exactly the same operation used for decryption.

Thus the encryption key is fully recovered.

Why Larger Block Size Does Not Fix It

One might think using a 3×3 or 4×4 matrix improves security.

However:

- A 3×3 key requires only 3 independent plaintext blocks.
- A 4×4 key requires only 4 independent blocks.

Once enough linear equations are collected, the key is determined uniquely.

Key Concept

Increasing matrix size increases work slightly, but does not remove the fundamental linear vulnerability.

Structural Reason for Failure

The Hill cipher operates entirely within:

$$\text{Linear Algebra over } \mathbb{Z}_{26}.$$

Modern cryptography avoids purely linear systems because:

- Linear systems are predictable.
- They can be solved with Gaussian elimination.
- They have no avalanche effect (small changes do not create chaotic output).

Comparison with Modern Cryptography

Modern systems (AES, RSA, ECC) use:

- Nonlinear substitutions
- Finite field arithmetic
- Modular exponentiation
- Elliptic curves

These prevent direct linear reconstruction.

Educational Value of the Hill Cipher

Despite its weakness, the Hill cipher is valuable because it demonstrates:

- Matrix multiplication
- Modular inverses
- Determinant-based invertibility
- Linear transformations
- Real-world application of LU-style solving

Engineering Note

The Hill cipher is not secure — but it is a powerful teaching tool for understanding linear algebra in cryptographic systems.

8 Error-Correcting Codes

8.1 Linear Codes and Generator Matrices

Digital communication is subject to noise. Error-correcting codes allow detection and correction of transmission errors.

Finite Field $\mathbb{Z}_2$

Most classical codes operate over the binary field:

$$\mathbb{Z}_2 = \{0, 1\}$$

with arithmetic modulo 2:

$$1 + 1 \equiv 0.$$

Key Concept

Binary linear codes are vector spaces over $\mathbb{Z}_2$.

Definition of a Linear Code

A linear (n, k) code is a k-dimensional subspace of $\mathbb{Z}_2^n$.

- k = number of message bits
- n = number of transmitted bits

Thus:

$$\text{Redundancy} = n - k.$$

Generator Matrix

Key Concept

A generator matrix G is a $k \times n$ matrix whose rows form a basis for the code.

Encoding is performed by:

$$c = mG$$

where:

- m = message vector (row vector)
- c = codeword

Example (3,2) Code

Let:

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

Encode message:

$$m = [1\ 0].$$

$$c = mG = [1\ 0] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = [1\ 0\ 1].$$

All arithmetic is modulo 2.

Geometric View

Codewords form a subspace of $\mathbb{Z}_2^n$.

Engineering Note

Linear algebra structure ensures:

- Sum of codewords is a codeword
- Scalar multiples remain in the code

8.2 Parity-Check Matrices and Error Detection

To detect errors, we introduce a matrix orthogonal to the code.

Parity-Check Matrix

Key Concept

A parity-check matrix H satisfies:

$$GH^T = 0.$$

A vector c is a valid codeword if and only if:

$$Hc^T = 0.$$

Thus H defines linear constraints.

Example (3,2) Code Continued

For:

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix},$$

a valid parity-check matrix is:

$$H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

Verify:

$$Hc^T = 0$$

for all valid codewords.

Error Detection

Suppose transmitted codeword:

$$c = [1\ 0\ 1].$$

If received vector:

$$r = [1\ 1\ 1],$$

compute:

$$Hr^T = [1\ 1\ 1] \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = 1 + 1 + 1 \equiv 1 \pmod 2.$$

Since result is not zero, an error occurred.

Key Concept

If $Hr^T \neq 0$, the received word is not a valid codeword.

8.3 The Hamming (7,4) Code

The Hamming $(7, 4)$ code is a classical single-error-correcting code.

- 4 message bits
- 3 parity bits

- 7 transmitted bits

Key Concept

The Hamming $(7, 4)$ code can detect up to two errors and correct all single-bit errors.

Parity-Check Matrix

A standard parity-check matrix is:

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

Each column is a distinct nonzero 3-bit binary vector.

Key Concept

Each column of H corresponds to the binary representation of a bit position.

Generator Matrix

A compatible generator matrix (transpose form shown) is:

$$G^T = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Encoding is performed by:

$$x = G^T p \pmod 2.$$

Encoding Example

Let the message be:

$$p = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Compute:

$$x = G^T p.$$

Perform multiplication modulo 2:

$$x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Thus the transmitted codeword is:

$$x = 0110011.$$

Engineering Note

All arithmetic in Hamming codes is performed modulo 2.

8.4 Syndrome Detection and Error Correction

When a codeword is transmitted over a noisy channel, errors may occur.

Let:

$$r = x + e,$$

where:

- x = transmitted codeword
- e = error vector

Syndrome Definition

Key Concept

The syndrome is defined as:

$$s = Hr \pmod 2.$$

If r is a valid codeword, then:

$$Hr = 0.$$

If $Hr \neq 0$, an error occurred.

Example 1: Single-Bit Error Correction

Suppose the transmitted codeword was:

$$x = 0110011.$$

Assume bit 6 flips during transmission:

$$r = 0110001.$$

Compute syndrome:

$$s = Hr.$$

Using:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad r = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Compute row-by-row (mod 2).

First syndrome bit:

$$s_1 = r_1 + r_3 + r_5 + r_7 = 0 + 1 + 0 + 1 = 2 \equiv 0.$$

Second syndrome bit:

$$s_2 = r_2 + r_3 + r_6 + r_7 = 1 + 1 + 0 + 1 = 3 \equiv 1.$$

Third syndrome bit:

$$s_3 = r_4 + r_5 + r_6 + r_7 = 0 + 0 + 0 + 1 = 1.$$

Thus:

$$s = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

Now compare with columns of H.
Column 6 of H is:

$$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

Key Concept

The syndrome equals the column of H corresponding to the error position.

Therefore, error occurred in bit 6.
Correct the error by flipping bit 6:

$$0110001 \to 0110011.$$

Recovered codeword is correct.

Example 2: No Error Case

If:

$$r = 0110011,$$

then:

$$s = Hr = 0.$$

Thus no detected error.

Example 3: Double Error Case

Suppose two bits flip:

$$r = 0010001.$$

Compute:

$$s = Hr.$$

The syndrome will be nonzero, but it will correspond to a single column of H, causing incorrect correction.

Engineering Note

Hamming (7,4) corrects all single-bit errors, but may mis-correct double-bit errors.

9 Least Squares

9.1 Overdetermined Systems

Consider a system:

$$Ax = b,$$

where A is an $m \times n$ matrix with $m > n$.
Such systems typically have more equations than unknowns.

Key Concept

An overdetermined system generally has no exact solution.

Instead of solving exactly, we seek x that minimizes the error:

$$\|Ax - b\|.$$

This leads to the least squares problem:

$$\min_x \|Ax - b\|^2.$$

Example

Fit a line $y = mx + c$ to data points:

$$(1, 1),\ (2, 2),\ (3, 2).$$

System:

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

There is no exact solution — we use least squares.

9.2 Orthogonal Projection Interpretation

The vector Ax lies in the column space of A.

If b is not in the column space, we project b onto $\text{Col}(A)$.

Key Concept

The least squares solution makes the error vector

$$e = b - Ax$$

orthogonal to $\text{Col}(A)$.

Thus:

$$A^T(b - Ax) = 0.$$

Geometrically:

- Ax is the orthogonal projection of b.
- The residual is perpendicular to the column space.

Projection Formula

If columns of A are independent:

$$\hat{x} = (A^T A)^{-1} A^T b.$$

9.3 The Normal Equations

From orthogonality:

$$A^T(b - Ax) = 0.$$

Rearrange:

$$A^T Ax = A^T b.$$

Key Concept

The equation

$$A^T Ax = A^T b$$

is called the normal equation.

Properties

- $A^T A$ is symmetric.
- If columns of A are independent, $A^T A$ is invertible.

Thus:

$$x = (A^T A)^{-1} A^T b.$$

Example (Line Fitting)

Using earlier data:

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

Compute:

$$A^T A = \begin{bmatrix} 14 & 6 \\ 6 & 3 \end{bmatrix}.$$

$$A^T b = \begin{bmatrix} 11 \\ 5 \end{bmatrix}.$$

Solve:

$$\begin{bmatrix} 14 & 6 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} 11 \\ 5 \end{bmatrix}.$$

Solution:

$$m = \frac{1}{2}, \quad c = \frac{2}{3}.$$

9.4 QR Factorization Method

Solving normal equations may cause numerical instability.

Instead, use QR factorization:

$$A = QR,$$

where:

- Q has orthonormal columns

- R is upper triangular

Since:

$$Ax = b,$$

becomes:

$$QRx = b.$$

Multiply by Q^T:

$$Rx = Q^T b.$$

Key Concept

Least squares can be solved using:

$$Rx = Q^T b.$$

This avoids computing $A^T A$.

Why QR is Better

- More numerically stable
- Avoids squaring condition number
- Used in practical software (MATLAB, NumPy)

9.5 Applications of Least Squares

Least squares appears everywhere in engineering and science.

1. Linear Regression

Fit model:

$$y = mx + c.$$

2. Polynomial Approximation

Fit quadratic:

$$y = ax^2 + bx + c.$$

Matrix form:

$$A = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \end{bmatrix}.$$

3. Signal Processing

Noise removal via projection.

4. Machine Learning

Least squares underlies:

- Linear regression
- Ridge regression
- Principal Component Analysis

Engineering Insight

Engineering Note

Least squares converts inconsistent systems into optimal approximate solutions via orthogonal projection.