Eco Industry
Oriented University

LINEAR ALGEBRA

Applied Matrices and Vector Spaces in Mining Engineering

Written by:
Bakti Siregar, M.Sc., CDS.

First Edition

Linear Algebra

Applied Matrices and Vector Spaces in Mining Engineering

Bakti Siregar, M.Sc., CDS.

Table of contents

Preface **3**

About the Writer 3

Acknowledgments 3

Feedback & Suggestions 4

1 Systems of Linear Equations **5**

1.1 Functions to Linear Equations 5

1.2 System of Linear Equations 6

1.3 Key Concepts of SLE 6

1.3.1 Unique Solution 6

1.3.2 No Solution 8

1.3.3 Infinite Solutions 9

1.4 Matrix Operations 11

1.4.1 Augmented Matrix 11

1.4.2 Elementary Row Operations 12

1.4.3 Reduced Row Echelon Form 13

1.4.4 Step-by-Step Example 14

1.5 Gaussian REF 14

1.5.1 Row Echelon Form (REF) 14

1.5.2 Gauss–Jordan (RREF) 16

1.5.3 Back Substitution 18

1.6 Applied of SLE 19

1.6.1 Mining Operations 20

1.6.2 Geology Modeling 20

1.6.3 Safety & Risk 21

1.6.4 Resource Allocation . . . 22
1.6.5 Environmental Monitoring . . . 23
Refrences . . . 24

2 Determinants **25**
2.1 Definition of a Determinant . . . 25
2.1.1 Determinat $M_{2\times 2}$ . . . 25
2.1.2 Determinat $M_{3\times 3}$ . . . 26
2.1.3 Determinat $M_{n\times n}$ . . . 27
2.2 Properties of Determinants . . . 33
2.2.1 Triangular Matrices . . . 33
2.2.2 Row Operations . . . 33
2.2.3 Invertibility . . . 34
2.2.4 Multiplicative Property . . . 34
2.2.5 Determinant of Transpose . . . 34
2.2.6 Scalar Multiplication . . . 34
2.2.7 Block Diagonal Matrices . . . 35
2.2.8 Zero Row or Column . . . 35
2.2.9 Linear Dependence . . . 35
2.3 Cramer's Rule . . . 36
2.4 Geometric Interpretation . . . 36
2.4.1 Area in 2D . . . 37
2.4.2 Volume in 3D . . . 38
2.5 Invertibility . . . 40
Refrences . . . 41

3 Matrix Inverse **43**
3.1 Inverse Definition . . . 44
3.2 Inverse Formula . . . 44
3.2.1 Adjoint / Cofactor Method . . . 44
3.2.2 Gauss-Jordan Method . . . 47
3.3 Inverse Properties . . . 49
3.4 Inverse Visualizations . . . 49
3.4.1 2D Matrix . . . 49

3.4.2 3D Matrix Inverse . . . 51
3.5 Applied of Invers . . . 53
3.5.1 Mining Equipment Allocation . . . 53
3.5.2 Ore Transport System . . . 54
3.5.3 Mineral Concentration Modeling . . . 54
3.5.4 Energy Distribution . . . 54
Refrences . . . 54

4 Matrix Factorization 55
4.1 Purpose / Benefits . . . 56
4.2 Types of Matrix Factorization . . . 56
4.2.1 LU Decomposition . . . 56
4.2.2 QR Decomposition . . . 56
4.2.3 Cholesky Decomposition . . . 56
4.2.4 Singular Value Decomposition . . . 57
4.2.5 Non-negative Matrix Factorization . . . 57
4.3 Applications . . . 57
4.3.1 Solving Linear Systems with LU . . . 58
4.3.2 Optimization Problems . . . 59
4.3.3 Recommender Systems . . . 60
4.3.4 Data Compression . . . 61
Refrences . . . 62

5 Vector Spaces 63
5.1 Definition . . . 64
5.2 Properties . . . 65
5.3 Examples of Vector Spaces . . . 65
5.4 Applications . . . 66
5.4.1 Ore Grade Estimation . . . 66
5.4.2 Resource Allocation . . . 66
5.4.3 Geospatial Modeling . . . 66
5.4.4 Production Planning . . . 66
5.4.5 Safety and Risk Analysis . . . 67
References . . . 67

6 Inner Product Spaces **69**
6.1 Definition 70
6.2 Properties 71
6.3 Examples 71
6.4 Applications 72
References 72

7 Orthogonality **73**
7.1 Definition 74
7.2 Properties 74
7.3 Examples 75
7.4 Applications 75
References 76

8 Linear Transformations **77**
8.1 Definition 78
8.2 Properties 78
8.3 Examples 79
8.4 Applications 79
References 79

9 Eigenvalues **81**
9.1 Definition 82
9.2 Properties 82
9.3 Examples 83
9.4 Applications 83
References 83

10 Case Studies **85**
10.1 Ore Grade Modeling 86
10.2 Sensor Data Analysis 86
10.3 Mine Planning and Optimization 86
10.4 Slope Stability Analysis 87
10.5 Ventilation System Optimization 87

In the modern landscape of mining engineering, linear algebra serves as a powerful engine behind data-driven decisions and innovation. From modeling mineral reserves through complex systems of equations, to applying eigenvalue techniques for assessing slope stability, and using matrix factorizations to structure and interpret vast datasets, linear algebra transforms abstract mathematics into practical solutions. By mastering these tools, students and professionals gain the ability to decode complex mining challenges, streamline engineering processes, and construct reliable models that drive efficiency, safety, and technological advancement in the mining industry.

This book offers a practical and application-driven introduction to linear algebra, designed specifically for mining contexts. Key topics include matrices and systems of linear equations, determinants, matrix inverses, matrix factorizations, vector spaces, inner product spaces, orthogonality, linear transformations, and eigenvalues. Each chapter bridges theory with hands-on applications—showing how abstract concepts translate into tools for mineral reserve planning, mine design, and operational optimization.

Beyond the fundamentals, the book highlights real-world case studies where linear algebra directly powers mining innovation: ore grade modeling, slope stability analysis, ventilation network design, and optimization of equipment usage. By blending solid mathematical foundations with cutting-edge engineering challenges, this book equips readers with both the technical depth and the practical mindset to apply linear algebra as a driver of efficiency, safety, and innovation in mining engineering.

Preface

About the Writer

Bakti Siregar, M.Sc., CDSS works as a Lecturer at the ITSB Data Science Program. He earned his Master's degree from the Department of Applied Mathematics at National Sun Yat Sen University, Taiwan. In addition to teaching, Bakti also works as a Freelance Data Scientist for leading companies such as JNE, Samora Group, Pertamina, and PT. Green City Traffic.

He has a strong enthusiasm for projects (and teaching) in the fields of Big Data Analytics, Machine Learning, Optimization, and Time Series Analysis, particularly in finance and investment. His core expertise lies in statistical programming languages such as R Studio and Python. He is also experienced in implementing database systems like MySQL/NoSQL for data management and is proficient in using Big Data tools such as Spark and Hadoop.

Some of his projects can be viewed here: Rpubs, Github, Website, and Kaggle

---

Acknowledgments

Linear Algebra is more than just theory, it's a powerful toolkit for solving real challenges in mining engineering. From handling large systems of equations to applying

eigenvalues in stability analysis, the concepts covered here are designed to connect mathematics directly with mining practice. This book introduces the essentials: matrices, determinants, inverses, factorizations, vector spaces, inner products, orthogonality, linear transformations, and eigenvalues—always with a focus on how they translate into real mining applications.

We thank all readers and learners who bring curiosity and fresh perspectives. Your questions and insights keep this journey alive. Our hope is that this material not only strengthens your foundation in Linear Algebra but also inspires you to use it as a driver of innovation, safety, and efficiency in mining engineering..

Feedback & Suggestions

Your feedback is invaluable in helping us refine and improve this book. We encourage readers to share their thoughts on the clarity, structure, and practical relevance of the material. Suggestions for expanding discussions—whether on matrices and systems of linear equations, determinants and matrix factorizations, vector spaces and orthogonality, or linear transformations and eigenvalues—are highly appreciated.

With your contributions, we aim to make this book a comprehensive and practice-oriented resource on **Linear Algebra** for Mining Engineering. Thank you for your engagement and support in shaping this learning journey.

For feedback and suggestions, please contact:

- siregarbakti@gmail.com
- siregarbakti@itsb.ac.id
- dsciencelabs@outlook.com

Chapter 1

Systems of Linear Equations

Linear algebra begins with one of its most fundamental topics: **Systems of Linear Equations (SLE)**. These systems appear everywhere in **mining engineering, geosciences, mineral processing, and numerical modeling** [1]–[4]. Figure 1.1 shows a **mind map of SLE** and its main concepts, operations, and applications in mining engineering.

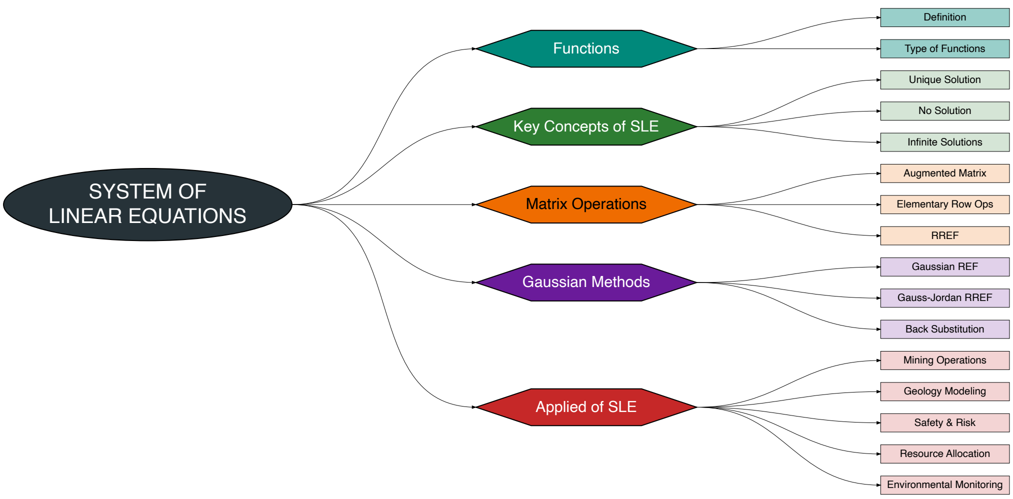

Figure 1.1: Mind Map of System of Linear Equations

1.1 Functions to Linear Equations

A **function** is a relation that connects each input value to exactly one output value [5], [6]. In mathematics, a function is often written as:

$$f(ax) = b$$

- a : Coefficient of the independent variable
- x : Input (independent variable)
- b : Output (dependent variable)

Functions are the foundation of linear equations, since every linear equation can be seen as a special type of function [2].

> ⚠ Note:
>
> A **linear equation** is a function where the variables only appear to the first power and are not multiplied together [1]. In general form:
>
> $$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$$
>
> where $a_1, a_2, \ldots, a_n$ are constants (coefficients), and b is a constant term.
>
> - With **1 variable**: $a_1x_1 = b$
> - With **2 variables**: $a_1x_1 + a_2x_2 = b$
> - With **N variables**: $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$

1.2 System of Linear Equations

A **SLE** is a collection of two or more linear equations involving the same set of variables. The goal is to find values of the variables that satisfy all equations at once [1], [2].

General form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \quad &\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

- m = number of equations
- n = number of variables

1.3 Key Concepts of SLE

1.3.1 Unique Solution

The system has exactly **one solution**. This occurs when the equations represent lines (or planes in higher dimensions) that **intersect at a single point** [1], [2].

Example

Consider the following system:

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

Solution

Solving the System:

$$\begin{aligned}
x_1 + x_2 &= 5 && \text{(first equation)} \\
x_2 &= 5 - x_1 && \text{(express } x_2 \text{ in terms of } x_1\text{)} \\
2x_1 - x_2 &= 1 && \text{(second equation)} \\
2x_1 - (5 - x_1) &= 1 && \text{(substitute } x_2 = 5 - x_1\text{)} \\
2x_1 - 5 + x_1 &= 1 && \text{(simplify)} \\
3x_1 - 5 &= 1 && \text{(combine like terms)} \\
3x_1 &= 6 && \text{(move constant to RHS)} \\
x_1 &= 2 && \text{(solve for } x_1\text{)} \\
x_2 &= 5 - 2 = 3 && \text{(substitute } x_1 = 2 \text{ into } x_2 = 5 - x_1\text{)} \\
(x_1, x_2) &= (2, 3) && \text{(final solution: intersection point)}
\end{aligned}$$

Conclusion:
The unique solution is:

$$(x_1, x_2) = (2, 3)$$

Solution: Visualization

We can visualize both equations as straight lines in the coordinate plane. The intersection point of these lines represents the **unique solution** [7].

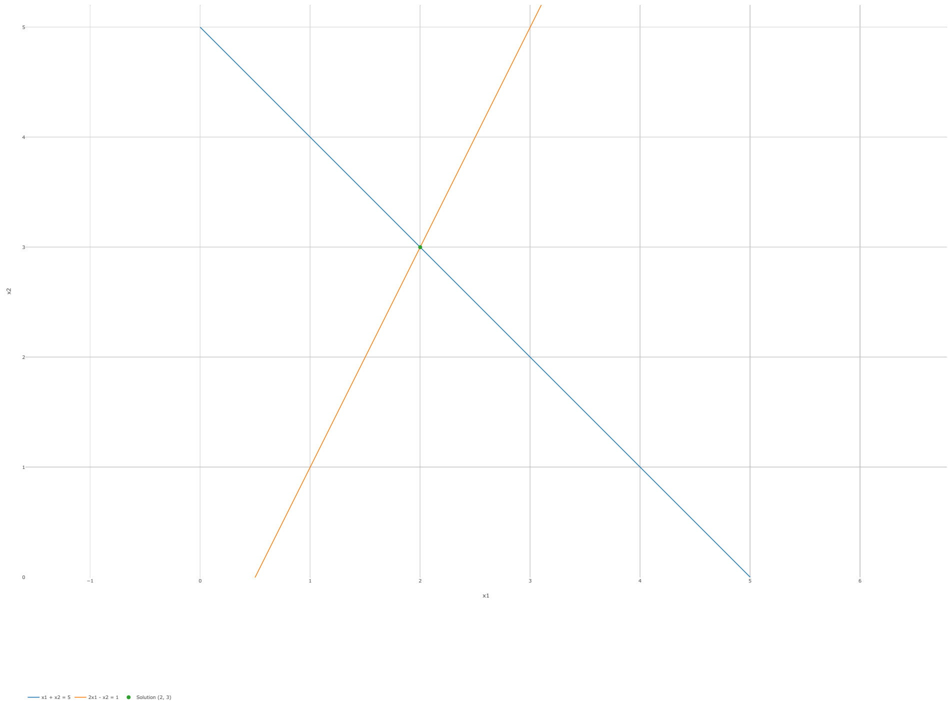

Figure 1.2: Interactive visualization of the system: x1 + x2 = 5 and 2x1 − x2 = 1

1.3.2 No Solution

The system has **no solution**. This happens when the equations are **inconsistent**, typically represented by **parallel lines** that never intersect [1].

Example

Consider the following system:

$$x_1 + x_2 = 4$$
$$x_1 + x_2 = 6$$

Question:
Is there a pair (x_1, x_2) that satisfies both equations simultaneously?

Solution

Let's try solving:

$$x_1 + x_2 = 4$$
$$x_1 + x_2 = 6$$

Subtracting (1) from (2):

$$(x_1 + x_2) - (x_1 + x_2) = 6 - 4$$

$$0 = 2 \quad \text{(contradiction)}$$

Since we arrive at a contradiction, the system is **inconsistent** and has **no solution**.

Solution: Visualization

Graphically, the two equations are **parallel lines**. Since they never intersect, there is **no common solution point**.

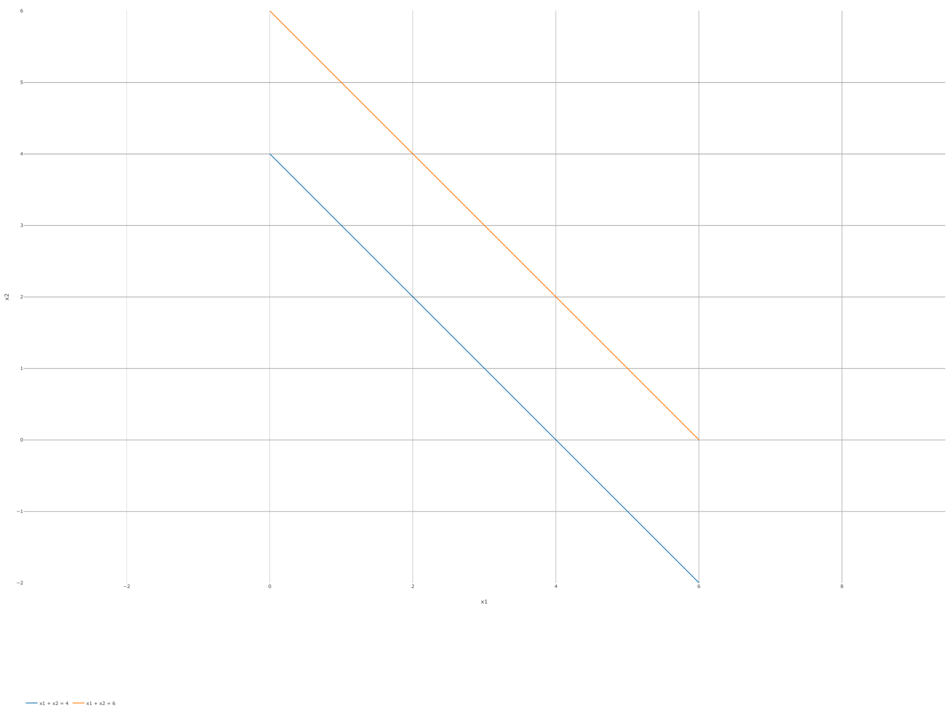

Figure 1.3: Interactive visualization of the inconsistent SLE: x1 + x2 = 4 and x1 + x2 = 6

1.3.3 Infinite Solutions

The system has **infinitely many solutions**. This occurs when the equations are **dependent**, meaning they represent the **same line** (or plane) written in different forms [1].

Example

Consider the following system:

$$x_1 + x_2 = 5$$
$$2x_1 + 2x_2 = 10$$

Question:
How many solutions does this system have?

Solution

Let's check:

$$x_1 + x_2 = 5$$
$$2x_1 + 2x_2 = 10$$

Notice that equation (2) is just **twice equation (1)**:

$$2(x_1 + x_2) = 2 \times 5 = 10$$

Thus, both equations describe the **same line**.
This means there are **infinitely many solutions**.
Every pair (x_1, x_2) such that

$$x_1 + x_2 = 5$$

is a valid solution. For example:

- $(x_1, x_2) = (0, 5)$
- $(x_1, x_2) = (2, 3)$
- $(x_1, x_2) = (4, 1)$
- etc.

Solution: Visualization

Graphically, both equations represent the **same line**. Hence, instead of intersecting at a single point, they **overlap completely**, which means infinitely many solutions.

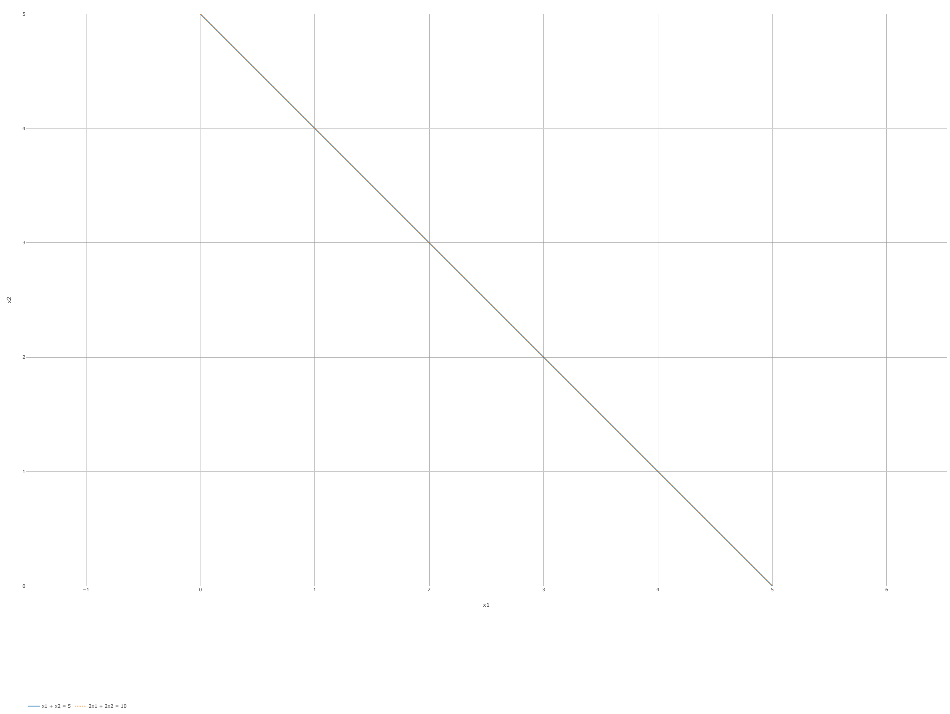

Figure 1.4: Interactive visualization of the dependent SLE: x1 + x2 = 5 and 2x1 + 2x2 = 10

1.4 Matrix Operations

In the final part of **Matrix Operations**, we focus on three main concepts that are widely used to solve systems of linear equations (SLE): **Augmented Matrix**, **Elementary Row Operations (ERO)**, and **Reduced Row Echelon Form (RREF)**. These topics provide both the theoretical foundation and practical algorithms used in engineering and numerical computation [1], [3], [6].

1.4.1 Augmented Matrix

Definition: An **augmented matrix** is a compact representation of a linear system where the coefficient matrix and the constant (right-hand side) vector are written side-by-side, separated by a vertical bar. This representation simplifies applying row operations and implementing algorithms like Gaussian elimination [1].

$$\begin{cases} a_1x_1 + a_2x_2 = b_1 \\ a_3x_1 + a_4x_2 = b_2 \end{cases}$$

The system can be written in augmented matrix form:

$$\left[\begin{array}{cc|c} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \end{array}\right]$$

1.4.2 Elementary Row Operations

There are three elementary row operations that do not change the solution set of a linear system [6]:

1. Swap two rows: $R_i \leftrightarrow R_j$.
2. Multiply a row by a nonzero scalar: $R_i \leftarrow kR_i,\ k \neq 0$.
3. Add a multiple of one row to another: $R_i \leftarrow R_i + kR_j$.

These operations are used to transform the augmented matrix toward Row Echelon Form (REF) or Reduced Row Echelon Form (RREF). Note on determinants (square coefficient matrices): swapping rows changes the sign of determinant, scaling a row scales the determinant, and adding a multiple of another row does not change the determinant [3].

Three basic row operations that do not change the solution set of an SLE:

Step 1: Eliminate entry (2,1)

$$R_2 \to R_2 - \frac{a_3}{a_1} R_1$$

$$\left[\begin{array}{cc|c} a_1 & a_2 & b_1 \\ 0 & a_4 - \frac{a_3}{a_1} a_2 & b_2 - \frac{a_3}{a_1} b_1 \end{array}\right]$$

Step 2: Normalize the pivot in row 2

$$R_2 \to \frac{1}{a_4 - \frac{a_3}{a_1} a_2} R_2$$

$$\left[\begin{array}{cc|c} a_1 & a_2 & b_1 \\ 0 & 1 & \dfrac{b_2 - \frac{a_3}{a_1} b_1}{a_4 - \frac{a_3}{a_1} a_2} \end{array}\right]$$

Step 3: Eliminate entry (1,2)

$$R_1 \to R_1 - a_2 R_2$$

$$\left[\begin{array}{cc|c} 1 & 0 & \dfrac{b_1a_4 - a_2b_2}{a_1a_4 - a_2a_3} \\ 0 & 1 & \dfrac{a_1b_2 - a_3b_1}{a_1a_4 - a_2a_3} \end{array}\right]$$

Note:

- EROs preserve the solution set.
- Determinant effects (for square matrices):
  - Swapping rows changes the sign of determinant.
  - Multiplying a row by k determinant multiplied by k.
  - Adding a multiple of another row determinant unchanged.

1.4.3 Reduced Row Echelon Form

RREF is a simplified form of a matrix obtained by applying **elementary row operations (EROs)**, making the solution easy to read.

Characteristics of RREF [1], [6]:

1. Each nonzero row has a **leading 1** (the first nonzero entry).
2. Each leading 1 appears to the **right** of the leading 1 in the row above.
3. Each pivot column has zeros in all other positions.
4. Any zero rows (if present) are placed at the bottom.

Closed-form solution (2×2 system).

For the system

$$a_1x_1 + a_2x_2 = b_1,$$
$$a_3x_1 + a_4x_2 = b_2,$$

the solution is:

$$x_1 = \frac{b_1a_4 - a_2b_2}{a_1a_4 - a_2a_3}, \quad x_2 = \frac{a_1b_2 - a_3b_1}{a_1a_4 - a_2a_3}.$$

Gaussian elimination reduces a system to *row echelon form* (REF), which is almost upper triangular, and then solves by **back substitution**. **Gauss–Jordan elimination** continues the process until the matrix is in **RREF**, allowing the solution to be read **directly** from the augmented matrix without back substitution.

1.4.4 Step-by-Step Example

Solve:

$$\begin{cases} x + 2y = 5 \\ 3x - y = 4 \end{cases}$$

Initial augmented matrix:

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

Step 1 — Eliminate (2,1): $R_2 \leftarrow R_2 - 3R_1$

$$\left[\begin{array}{cc|c} 1 & 2 & 5 \\ 0 & -7 & -11 \end{array}\right]$$

Step 2 — Make leading 1 in row 2: $R_2 \leftarrow (-\frac{1}{7})R_2$.

$$\left[\begin{array}{cc|c} 1 & 2 & 5 \\ 0 & 1 & \frac{11}{7} \end{array}\right]$$

Step 3 — Eliminate above leading 1: $R_1 \leftarrow R_1 - 2R_2$.

$$\left[\begin{array}{cc|c} 1 & 0 & \frac{13}{7} \\ 0 & 1 & \frac{11}{7} \end{array}\right]$$

Final solution:

$$x = \frac{13}{7}, \quad y = \frac{11}{7}$$

1.5 Gaussian REF

Gaussian methods are systematic ways to solve systems of linear equations (SLE). The main idea: simplify the system step by step (forward elimination) until the matrix is in **Row Echelon Form (REF)**, then solve by **back-substitution** [1], [3].

1.5.1 Row Echelon Form (REF)

The **Row Echelon Form (REF)** is an intermediate step in Gaussian elimination with the following properties [6]:

- The matrix looks like a **triangle**: all entries **below** each pivot are zero.

- Each pivot (the first nonzero entry in a row) appears to the **right** of the pivot in the row above.
- Any rows consisting entirely of zeros appear **at the bottom** of the matrix.

REF is obtained by forward elimination (elementary row operations). After REF is reached, use **back-substitution** starting from the bottom row to compute the solution.

Example

Consider the following augmented matrix:

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

This matrix represents the system:

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 4 \\ -5x_2 + x_3 &= -1 \end{aligned}$$

- In the **first row**, the pivot is the coefficient of x_1, which is 1.
- In the **second row**, the pivot is -5 (for x_2), and the entry below the pivot in the first column is already 0.
- Therefore the matrix is already in **Row Echelon Form**.

Steps to obtain REF (Gaussian elimination — general procedure)

1. **Choose a pivot column:** Start from the leftmost column that has a nonzero entry.
2. **Select a pivot row:** Choose a row with a nonzero entry in that column (often the current row).
3. **(Optional) Scale pivot to 1:** Divide the pivot row by the pivot value if you want a leading 1 (not required for REF; required for RREF).
4. **Eliminate below pivot:** Use row operations to make every entry **below** the pivot equal to 0.
5. **Move to the submatrix below and to the right:** Repeat the process for the remaining rows and columns.
6. **Move zero rows to the bottom:** If any row becomes all zeros, place it at the bottom of the matrix.

Row operations allowed (do not change solution set):

- $R_i \leftrightarrow R_j$ (swap two rows)

- $R_i \leftarrow kR_i$ (multiply a row by a nonzero scalar k)
- $R_i \leftarrow R_i + kR_j$ (add a multiple of one row to another)

Solution

From the REF example we have:

$$\begin{aligned} &(1)\ x_1 + 2x_2 + x_3 = 4 \\ &(2)\ -5x_2 + x_3 = -1 \end{aligned}$$

Solve (2) for x_3 in terms of x_2:

$$-5x_2 + x_3 = -1 \quad \Longrightarrow \quad x_3 = 5x_2 - 1$$

Substitute into (1):

$$x_1 + 2x_2 + (5x_2 - 1) = 4$$

Simplify:

$$x_1 + 7x_2 - 1 = 4 \quad \Longrightarrow \quad x_1 + 7x_2 = 5$$

Thus:

$$x_1 = 5 - 7x_2, \qquad x_3 = 5x_2 - 1$$

Parameterize the solution by letting $x_2 = t$ (free parameter). Then

$$\begin{aligned} x_1 &= 5 - 7t \\ x_2 &= t \\ x_3 &= 5t - 1 \end{aligned}$$

for any real number t.

1.5.2 Gauss–Jordan (RREF)

The **Reduced Row Echelon Form (RREF)** is the final stage of Gaussian elimination, obtained using the **Gauss–Jordan method**. It goes further than **Row Echelon Form (REF)** by ensuring that each pivot column contains a **leading 1 with zeros both below and above it** [1], [6].

Key Properties of RREF:

- Each **pivot (leading 1)** is the only nonzero entry in its column.
- The pivot in each successive row appears **to the right** of the pivot in the row above.
- Rows consisting entirely of zeros, if any, are always at the bottom.

- This is the **simplest form** of a system: the solution can be read directly without back-substitution.

Example

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & b_1 \\ 0 & 1 & 0 & b_2 \\ 0 & 0 & 1 & b_3 \end{array}\right]$$

This corresponds to the system:

$$\begin{aligned} x_1 &= b_1 \\ x_2 &= b_2 \\ x_3 &= b_3 \end{aligned}$$

So the unique solution is:

$$x_1 = b_1, \quad x_2 = b_2, \quad x_3 = b_3$$

Steps to Obtain RREF (Gauss–Jordan Elimination)

1. **Transform to REF**: Use Gaussian elimination to create a triangular form with pivots.

2. **Scale pivots to 1**: Divide each pivot row so that the pivot element becomes exactly 1.

3. **Eliminate above pivots**: Use row operations to make all entries **above** each pivot equal to 0.

4. **Check consistency**: If a row becomes

$$[0 \ \ 0 \ \ 0 \mid b], \quad b \neq 0,$$

   then the system is **inconsistent** (no solution).

5. **Read solution**: Once in RREF, the solution can be read directly.

Consider the system:

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 4 \\ -5x_2 + x_3 &= -1 \end{aligned}$$

The RREF of its augmented matrix is:

$$\left[\begin{array}{ccc|c} 1 & 0 & 7 & 5 \\ 0 & 1 & -\frac{1}{5} & \frac{1}{5} \end{array}\right]$$

This means:

$$\begin{aligned} x_1 + 7x_3 &= 5 \\ x_2 - \tfrac{1}{5}x_3 &= \tfrac{1}{5} \end{aligned}$$

Rewriting:

$$\begin{aligned} x_1 &= 5 - 7x_3 \\ x_2 &= \tfrac{1}{5} + \tfrac{1}{5}x_3 \end{aligned}$$

Let

$$x_3 = t$$

(a free parameter). Then the solution set is:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ \frac{1}{5} \\ 0 \end{bmatrix} + t \begin{bmatrix} -7 \\ \frac{1}{5} \\ 1 \end{bmatrix}, \quad t \in \mathbb{R}.$$

1.5.3 Back Substitution

When we stop at **Row Echelon Form (REF)** rather than continuing to RREF, we must use **back substitution** to solve the system [1], [6].

- Start solving from the **bottom row** (which involves fewer variables).
- Substitute upward step by step until all variables are expressed (or expressed in terms of free parameters).

Example

Consider the augmented matrix in REF:

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

This corresponds to the system:

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 4 \\ -5x_2 + x_3 &= -1 \end{aligned}$$

Step 1: Start with the bottom row, From equation (2):

$$-5x_2 + x_3 = -1$$

Rearrange to express x_2 in terms of x_3:

$$x_2 = \frac{1 + x_3}{5}$$

Step 2: Substitute into the first row, Plug $x_2 = \frac{1+x_3}{5}$ into equation (1):

$$x_1 + 2\left(\frac{1 + x_3}{5}\right) + x_3 = 4$$

Simplify:

$$x_1 + \frac{2}{5} + \frac{2}{5}x_3 + x_3 = 4$$

$$x_1 + \frac{7}{5}x_3 + \frac{2}{5} = 4$$

$$x_1 = 4 - \frac{2}{5} - \frac{7}{5}x_3$$

$$x_1 = \frac{18}{5} - \frac{7}{5}x_3$$

Step 3: Express the solution set

Let $x_3 = t \in \mathbb{R}$ (a free parameter). Then:

$$x_1 = \frac{18}{5} - \frac{7}{5}t$$
$$x_2 = \frac{1 + t}{5}$$
$$x_3 = t$$

Final Solution (Parametric Form)

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{18}{5} \\ \frac{1}{5} \\ 0 \end{bmatrix} + t \begin{bmatrix} -\frac{7}{5} \\ \frac{1}{5} \\ 1 \end{bmatrix}, \quad t \in \mathbb{R}.$$

1.6 Applied of SLE

Systems of Linear Equations (SLE) are widely used in real-world mining and geoscience problems.
They provide a mathematical way to represent **constraints** and **balances** in production, blending, transport, and geological modeling.

1.6.1 Mining Operations

In mining operations, **linear equations** often appear when formulating real-world decision problems.
They help transform complex constraints into mathematical form, making them easier to analyze and solve [8], [9].

Some common applications include:

- **Production planning**: deciding how much ore to extract from different mines to meet total demand.
- **Blending problems**: mixing ores of different grades to achieve required quality.
- **Transportation optimization**: minimizing costs of moving ore from mines to processing plants.

> **ℹ Case Example**
>
> A company extracts ore from **two mines** (Mine A and Mine B).
> Let:
> - x_1 = tons of ore from Mine A
> - x_2 = tons of ore from Mine B
> The company faces the following requirements:
>
> 1. **Production Target**
>
> $$x_1 + x_2 = 100 \quad \text{(tons of ore required)}$$
>
> 2. **Metal Content Requirement**
>
> $$2x_1 + 3x_2 = 240 \quad \text{(metal content requirement)}$$

1.6.2 Geology Modeling

In geology and mining engineering, mathematical equations are often used to **model subsurface structures** and **predict ore grades**. These models are essential for **resource estimation**, **mine planning**, and **risk reduction** in exploration. Some common applications include:

- **Representing subsurface layers** using mathematical equations.
- **Estimating unknown geological parameters** from drilling data.
- **3D modeling of mineral deposits**, which often requires solving large systems of equations.

Case Example

Suppose a geologist wants to predict the **ore grade (G)** based on spatial coordinates (x_1, x_2) from drilling data. A **multiple regression model** can be used:

$$G = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

where:

- G = predicted ore grade (% or g/ton)
- x_1 = Easting coordinate (m)
- x_2 = Northing coordinate (m)
- $\beta_0, \beta_1, \beta_2$ = regression coefficients to be estimated

Assume that drilling results have been collected as follows:

Drill Hole	Coordinate x_1 (Easting, m)	Coordinate x_2 (Northing, m)	Ore Grade G (%)
DH1	10	20	2.5
DH2	15	25	3.0
DH3	20	30	3.5

1.6.3 Safety & Risk

Some common linear applications in underground mining include:

- **Modeling air flow and gas concentration** in underground mines (ventilation equations).
- **Calculating safe load distribution** for mine structures.
- **Risk assessment models** where multiple factors interact linearly.

A general abstract form is:

$$a_1x_1 + a_2x_2 + a_3x_3 = \text{Safe threshold level}$$

Below is a concrete example with physical interpretation and step-by-step solution.

Case Example

Suppose we define:

- x_1 = control setting 1 (e.g., fan group A, dimensionless speed parameter)

- x_2 = control setting 2 (e.g., fan group B or auxiliary ventilation)
- x_3 = control setting 3 (e.g., ventilation door/duct opening)

We have three constraints (example values chosen for consistency):

1. **Total airflow target** (m^3/s):

$$x_1 + x_2 + x_3 = 300$$

2. **Effective fresh-air contribution** (weighted combination of controls):

$$0.5x_1 + 0.3x_2 + 0.2x_3 = 98$$

3. **Safety metric** (e.g., pressure/load distribution):

$$0.2x_1 + 0.1x_2 + 0.7x_3 = 112$$

Augmented matrix form:

$$\begin{bmatrix} 1 & 1 & 1 & | & 300 \\ 0.5 & 0.3 & 0.2 & | & 98 \\ 0.2 & 0.1 & 0.7 & | & 112 \end{bmatrix}$$

1.6.4 Resource Allocation

Applications of linear models in **resource allocation** include:

- Optimizing the use of limited resources (machines, workers, energy).
- Scheduling shifts and equipment under constraints.
- Balancing multiple objectives such as cost, time, and productivity.

Case Example

Suppose a mining company must allocate resources to two tasks:

- x_1 = hours allocated to **excavation work**
- x_2 = hours allocated to **transportation work**

The constraints are:

$$2x_1 + 4x_2 \leq 200 \quad \text{(available labor hours)}$$

$$3x_1 + x_2 \leq 150 \quad \text{(available machine hours)}$$

Additionally, both decision variables must be **non-negative**:

$$x_1 \geq 0, \quad x_2 \geq 0$$

If the **profit function** is defined as:

$$Z = 50x_1 + 40x_2$$

then the **linear programming problem** is:

$$\text{Maximize } Z = 50x_1 + 40x_2$$

subject to:

$$\begin{aligned} 2x_1 + 4x_2 &\leq 200 \\ 3x_1 + x_2 &\leq 150 \\ x_1, x_2 &\geq 0 \end{aligned}$$

1.6.5 Environmental Monitoring

Linear equations can also be applied to **environmental monitoring** in mining operations, where sustainability and safety are critical. Common applications include:

- Tracking pollution levels (air, water, soil) with sensor data.
- Balancing waste treatment inputs and outputs.
- Modeling the dispersion of contaminants using linear systems.

Case Example

A mining company monitors pollution from two main sources:

- x_1 = emission level from the **processing plant** (in tons of CO equivalent per day)
- x_2 = emission level from **transportation activities** (in tons of CO equivalent per day)

Each source contributes differently to the overall **pollution index**:

- The processing plant contributes **2 units** per ton of emission ($c_1 = 2$).
- Transportation contributes **3 units** per ton of emission ($c_2 = 3$).

The government regulation sets a **pollution index target of 100 units**.
Tasks:

1. Write down the linear equation that represents the pollution index requirement.

2. If the company emits $x_1 = 20$ tons from the processing plant, how much x_2 (transportation emission) is allowed to meet the target?

3. If transportation emits $x_2 = 10$ tons, calculate the maximum processing plant emission x_1 allowed.

4. Interpret the meaning of these results in terms of **environmental compliance**.

Hint: Start with the equation

$$2x_1 + 3x_2 = 100$$

Refrences

Chapter 2

Determinants

In the previous chapters, we studied **Systems of Linear Equations (SLE)** and solved them using **row reduction methods** such as **Gaussian elimination** and **Gauss–Jordan elimination**. These methods relied on transforming the augmented matrix into **Row Echelon Form (REF)** or **Reduced Row Echelon Form (RREF)** [1], **lay2012linear?**.

However, row reduction is not the only method to solve linear systems. Another powerful tool is the **determinant of a matrix**, which provides:

- A criterion for whether a matrix is **invertible**.
- A method to solve linear systems directly using **Cramer's Rule**.
- A way to describe geometric properties such as **area** and **volume** [3], **meyer2000?**.

Thus, determinants build a natural bridge between **row operations** and more advanced concepts like **matrix inverse**, **eigenvalues**, and **geometric transformations**. More details about this topic can be visualized in the following **Mind Map**.

2.1 Definition of a Determinant

The **determinant** is a scalar value associated with a square matrix that provides important information about the matrix, such as invertibility, scaling of geometric objects, and solutions of linear systems [1], **lay2012linear?**, **meyer2000?**.

2.1.1 Determinat $M_{2\times 2}$

For a 2×2 matrix:

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

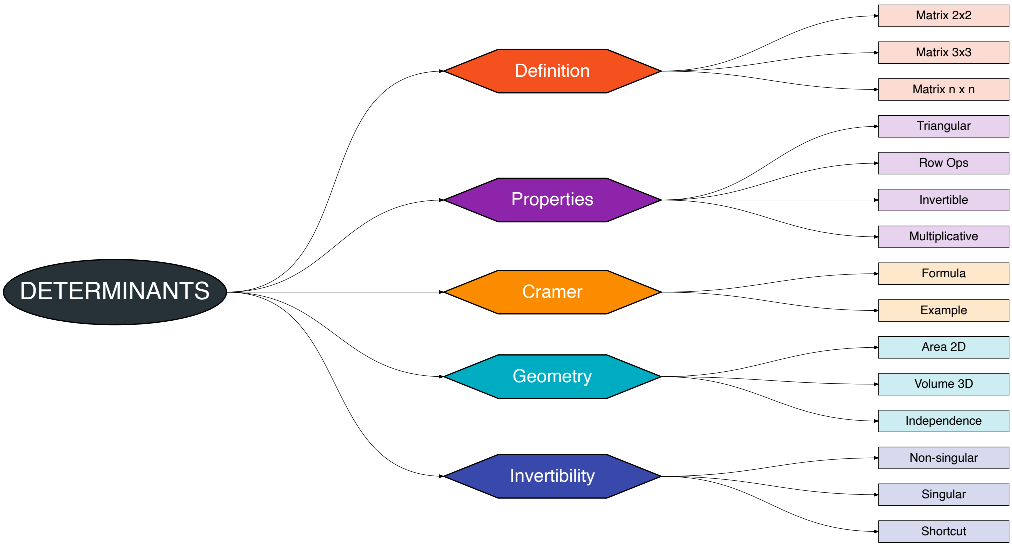

Figure 2.1: Mind Map of Determinants

2.1.2 Determinat $M_{3\times 3}$

When dealing with a 3×3 **matrix**, one convenient method to calculate its determinant is **Sarrus' Rule**. This method provides a simple visual approach that avoids the longer Laplace expansion process.

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, \quad \det(A) = aei + bfg + cdh - ceg - bdi - afh$$

Note:

The **Sarrus Rule** is a shortcut method to compute the determinant.
Step 1. Rewrite the first two columns of A to the right of the matrix:

$$\begin{array}{ccc|cc} a & b & c & a & b \\ d & e & f & d & e \\ g & h & i & g & h \end{array}$$

Step 2. Compute the sum of the products of the **three downward diagonals**:

- $a \cdot e \cdot i$
- $b \cdot f \cdot g$
- $c \cdot d \cdot h$

So the downward sum is:

$$(aei) + (bfg) + (cdh)$$

Step 3. Compute the sum of the products of the **three upward diagonals**:

- $c \cdot e \cdot g$
- $a \cdot f \cdot h$
- $b \cdot d \cdot i$

So the upward sum is:

$$(ceg) + (afh) + (bdi)$$

Step 4. Subtract the two results:

$$\det(A) = (aei + bfg + cdh) - (ceg + afh + bdi)$$

Example

Let

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Solution

Apply Sarrus' Rule:

- Downward diagonals:
  $1 \cdot 5 \cdot 9 + 2 \cdot 6 \cdot 7 + 3 \cdot 4 \cdot 8 = 45 + 84 + 96 = 225$
- Upward diagonals:
  $3 \cdot 5 \cdot 7 + 1 \cdot 6 \cdot 8 + 2 \cdot 4 \cdot 9 = 105 + 48 + 72 = 225$

Thus:

$$\det(A) = 225 - 225 = 0$$

So the matrix A is **singular** (non-invertible).

2.1.3 Determinat $M_{n \times n}$

So far, we have seen how to compute determinants of 2×2 and 3×3 matrices. For larger matrices, however, the computation becomes more complex and tedious. To handle this, two general approaches are commonly used:

1. **Laplace Expansion (Cofactor Expansion)** – expands the determinant along a row or column using minors and cofactors.

2. **Row Reduction / Triangular Form** – simplifies the matrix to an upper or lower triangular form, where the determinant is the product of the diagonal entries [1], **lay2012linear?**, **meyer2000?**.

Laplace Expansion

The **Laplace expansion (Cofactor Expansion)** allows us to compute the determinant of any $n \times n$ matrix by expanding along a row or column.

For a matrix $A = [a_{ij}]$ of order n:

$$\det(A) = \sum_{j=1}^{n} (-1)^{1+j}\, a_{1j}\, M_{1j}$$

where M_{1j} is the determinant of the $(n-1) \times (n-1)$ submatrix obtained by deleting the **first row** and the ***j*-th column** of A.

In general, expanding along the i-th row:

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j}\, a_{ij}\, M_{ij}$$

Here, the factor $(-1)^{i+j}$ ensures the **alternating signs** (checkerboard pattern).

Example

Determinant of a 4×4 matrix by Laplace Expansion
Consider the matrix

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

We will compute $\det(A)$ using **Laplace Expansion** along the first row.

Solution: Laplace Expansion along Row 3

We expand along the **3rd row**:

$$\det(A) = \sum_{j=1}^{4} (-1)^{3+j} a_{3j} M_{3j},$$

where M_{3j} is the minor obtained by deleting row 3 and column j.
Step 1: Identify elements of row 3:

$$R_3 = [0, 5, 2, 1]$$

Notice $a_{31} = 0$, so the first term contributes $\mathbf{0}$.

Step 2: Compute minor M_{32} (delete row 3, column 2):
Submatrix:

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

Compute determinant using **Sarrus' Rule**:

$$\begin{aligned} M_{32} &= 1((-1)\cdot 4 - 2\cdot 3) - 0(4\cdot 4 - 2\cdot 2) + 3(4\cdot 3 - (-1)\cdot 2) \\ &= 1(-4-6) - 0(\cdots) + 3(12+2) \\ &= -10 + 0 + 42 = 32 \end{aligned}$$

Cofactor:

$$C_{32} = (-1)^{3+2} \cdot a_{32} \cdot M_{32} = (-1)^5 \cdot 5 \cdot 32 = -160$$

Step 3: Compute minor M_{33} (delete row 3, column 3):
Submatrix:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 2 \\ 2 & 0 & 4 \end{bmatrix}.$$

Compute determinant:

$$\begin{aligned} M_{33} &= 1(1\cdot 4 - 2\cdot 0) - 2(4\cdot 4 - 2\cdot 2) + 3(4\cdot 0 - 1\cdot 2) \\ &= 1(4-0) - 2(16-4) + 3(0-2) \\ &= 4 - 24 - 6 = -26 \end{aligned}$$

Cofactor:

$$C_{33} = (-1)^{3+3} \cdot a_{33} \cdot M_{33} = (+1) \cdot 2 \cdot (-26) = -52$$

Step 4: Compute minor M_{34} (delete row 3, column 4):
Submatrix:

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

Compute determinant:

$$\begin{aligned} M_{34} &= 1(1\cdot 3 - (-1)\cdot 0) - 2(4\cdot 3 - (-1)\cdot 2) + 0(4\cdot 0 - 1\cdot 2) \\ &= 1(3-0) - 2(12+2) + 0(-2) \\ &= 3 - 28 + 0 = -25 \end{aligned}$$

Cofactor:

$$C_{34} = (-1)^{3+4} \cdot a_{34} \cdot M_{34} = (-1) \cdot 1 \cdot (-25) = 25$$

Step 5: Combine terms

$$\det(A) = 0 + (-160) + (-52) + 25 = -187$$

Final Result:

$$\boxed{\det(A) = -187}$$

Row Reduction Method

While Laplace expansion is useful conceptually, it becomes very inefficient for large matrices because the number of operations grows rapidly. A more practical method is to transform A into an **upper triangular matrix** using **elementary row operations** (Gaussian elimination).

- The determinant of a triangular matrix is the **product of its diagonal entries**.
- However, we must track the effect of each row operation on the determinant:
  1. Swapping two rows $\Rightarrow$ determinant changes sign.
  2. Multiplying a row by k $\Rightarrow$ determinant is multiplied by k.
  3. Adding a multiple of one row to another $\Rightarrow$ determinant unchanged.

Example

Determinant of a 4×4 matrix by row reduction,

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

We will perform Gaussian elimination to transform A into an **upper triangular** matrix U. All row operations used are of the form $R_i \leftarrow R_i + kR_j$ (adding a multiple of one row to another), which do **not** change the determinant.

Solution

Step 0: Initial matrix

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

Step 1: Eliminate entries below pivot $a_{11} = 1$
Use R_1 to eliminate the entries in column 1 of R_2 and R_4:

- $R_2 \leftarrow R_2 - 4R_1$
- $R_4 \leftarrow R_4 - 2R_1$

Compute:

$$R_2 = [4,\ 1,\ -1,\ 2] - 4[1,\ 2,\ 0,\ 3] = [0,\ -7,\ -1,\ -10],$$
$$R_4 = [2,\ 0,\ 3,\ 4] - 2[1,\ 2,\ 0,\ 3] = [0,\ -4,\ 3,\ -2].$$

Thus

$$A^{(1)} = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & -7 & -1 & -10 \\ 0 & 5 & 2 & 1 \\ 0 & -4 & 3 & -2 \end{bmatrix}.$$

Step 2: Pivot at $a_{22} = -7$. Eliminate entries below it (column 2).
We eliminate the (3,2) and (4,2) entries using row 2.

- For row 3: factor $= \dfrac{5}{-7} = -\dfrac{5}{7}$. Use $R_3 \leftarrow R_3 - (-\frac{5}{7})R_2 = R_3 + \frac{5}{7}R_2$.
- For row 4: factor $= \dfrac{-4}{-7} = \frac{4}{7}$. Use $R_4 \leftarrow R_4 - \frac{4}{7}R_2$.

Compute:

$$\tfrac{5}{7}R_2 = \tfrac{5}{7}[0, -7, -1, -10] = [0, -5, -\tfrac{5}{7}, -\tfrac{50}{7}],$$
$$R_3 = [0, 5, 2, 1] + [0, -5, -\tfrac{5}{7}, -\tfrac{50}{7}] = [0,\ 0,\ 2 - \tfrac{5}{7},\ 1 - \tfrac{50}{7}]$$
$$= \left[0,\ 0,\ \tfrac{9}{7},\ -\tfrac{43}{7}\right].$$

and

$$\tfrac{4}{7}R_2 = [0, -4, -\tfrac{4}{7}, -\tfrac{40}{7}],$$
$$R_4 = [0, -4, 3, -2] - [0, -4, -\tfrac{4}{7}, -\tfrac{40}{7}]$$
$$= \left[0,\ 0,\ 3 + \tfrac{4}{7},\ -2 + \tfrac{40}{7}\right] = \left[0,\ 0,\ \tfrac{25}{7},\ \tfrac{26}{7}\right].$$

Thus

$$A^{(2)} = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & -7 & -1 & -10 \\ 0 & 0 & \frac{9}{7} & -\frac{43}{7} \\ 0 & 0 & \frac{25}{7} & \frac{26}{7} \end{bmatrix}.$$

Step 3: Pivot at $a_{33} = \frac{9}{7}$. Eliminate entry below it (column 3).
Eliminate the (4,3) entry. Factor:

$$\text{factor} = \frac{\frac{25}{7}}{\frac{9}{7}} = \frac{25}{9}.$$

Perform $R_4 \leftarrow R_4 - \dfrac{25}{9}R_3$.
Compute:

$$\frac{25}{9}R_3 = \frac{25}{9}\left[0, 0, \tfrac{9}{7}, -\tfrac{43}{7}\right] = \left[0, 0, \tfrac{25}{7}, -\tfrac{1075}{63}\right],$$

$$R_4 = \left[0, 0, \tfrac{25}{7}, \tfrac{26}{7}\right] - \left[0, 0, \tfrac{25}{7}, -\tfrac{1075}{63}\right] = \left[0, 0, 0, \tfrac{26}{7} + \tfrac{1075}{63}\right].$$

Compute the final entry:

$$\tfrac{26}{7} = \tfrac{234}{63}, \qquad \tfrac{234}{63} + \tfrac{1075}{63} = \tfrac{1309}{63}.$$

So

$$R_4 = \left[0, 0, 0, \tfrac{1309}{63}\right].$$

Now the matrix is upper triangular:

$$U = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & -7 & -1 & -10 \\ 0 & 0 & \tfrac{9}{7} & -\tfrac{43}{7} \\ 0 & 0 & 0 & \tfrac{1309}{63} \end{bmatrix}.$$

Step 4: Determinant from diagonal product
Because all operations were of the form $R_i \leftarrow R_i + kR_j$ (determinant-preserving), the determinant of A equals the product of the diagonal entries of U:

$$\det(A) = 1 \cdot (-7) \cdot \frac{9}{7} \cdot \frac{1309}{63}.$$

Simplify:

$$(-7) \cdot \frac{9}{7} = -9,$$

so

$$\det(A) = -9 \cdot \frac{1309}{63} = -\frac{9 \cdot 1309}{63} = -\frac{1309}{7} = -187.$$

Therefore,

$$\boxed{\det(A) = -187.}$$

Remarks:

- Laplace expansion works for any $n \times n$ matrix.
- For larger matrices, row reduction is usually faster and less error-prone.
- This method also connects nicely with the earlier discussion of 3×3 determinants using **Sarrus' Rule**.

2.2 Properties of Determinants

Determinants have several important properties that make them useful in linear algebra. These properties help simplify computations, analyze matrix invertibility, and understand geometric interpretations [1], **lay2012linear?**, **meyer2000?**:

2.2.1 Triangular Matrices

For any $n \times n$ **upper or lower triangular matrix** T:

$$T = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ 0 & t_{22} & \dots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & t_{nn} \end{bmatrix}, \quad \det(T) = \prod_{i=1}^{n} t_{ii}$$

Example

$$T = \begin{bmatrix} 2 & 3 & 1 \\ 0 & -1 & 4 \\ 0 & 0 & 5 \end{bmatrix}, \quad \det(T) = 2 \cdot (-1) \cdot 5 = -10$$

2.2.2 Row Operations

Let A be an $n \times n$ matrix:

Row swap

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad R_1 \leftrightarrow R_2 \Rightarrow \det(\text{swapped}) = -\det(A)$$

Row Scaling

Multiplying a row by a scalar k multiplies the determinant by k.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad 2 \cdot R_1 \Rightarrow B = \begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix}$$

Compute determinants:

$$\det(A) = 1 \cdot 4 - 2 \cdot 3 = -2$$

$$\det(B) = 2 \cdot 4 - 4 \cdot 3 = -4 = 2 \cdot \det(A)$$

Row addition

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad R_2 \to R_2 + 3R_1 \Rightarrow \det = \det(A)$$

2.2.3 Invertibility

A square matrix A is **invertible** iff $\det(A) \neq 0$.

Example

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad \det(A) = 1 \cdot 4 - 2 \cdot 3 = -2 \neq 0$$

2.2.4 Multiplicative Property

For $A, B \in \mathbb{R}^{n \times n}$:

$$\det(AB) = \det(A) \cdot \det(B)$$

Example

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \ B = \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}, \ \det(AB) = (1 \cdot 2)(3 \cdot 4) = 24$$

2.2.5 Determinant of Transpose

$$\det(A^T) = \det(A)$$

Example

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}, \quad \det(A^T) = -2 = \det(A)$$

2.2.6 Scalar Multiplication

$$\det(kA) = k^n \cdot \det(A)$$

Example

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \; k = 2, \; \det(2A) = 2^2 \cdot (-2) = -8$$

2.2.7 Block Diagonal Matrices

$$A = \begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}, \quad \det(A) = \det(B) \cdot \det(C)$$

Example

$$B = [2], \; C = \begin{bmatrix} 1 & 3 \\ 0 & 4 \end{bmatrix}, \; A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 4 \end{bmatrix}, \; \det(A) = 2 \cdot (1 \cdot 4 - 0 \cdot 3) = 8$$

2.2.8 Zero Row or Column

If any row or column is all zeros, then $\det(A) = 0$.

Example

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 4 & 5 & 6 \end{bmatrix}, \quad \det(A) = 0$$

2.2.9 Linear Dependence

If the rows (or columns) are linearly dependent:

$$\det(A) = 0$$

Example

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 0 & 1 & 1 \end{bmatrix}, \quad \text{row } 2 = 2 * \text{row } 1 \Rightarrow \det(A) = 0$$

2.3 Cramer's Rule

Determinants allow us to solve linear systems using **Cramer's Rule**. For a system of n equations with n unknowns $A\mathbf{x} = \mathbf{b}$ where A is an $n \times n$ matrix with $\det(A) \neq 0$, the solution is:

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, 2, \dots, n$$

Here, A_i is the matrix formed by replacing the i-th column of A with the vector $\mathbf{b}$ [1], **lay2012linear?**, **meyer2000?**.

Example

$$a_1x + b_1y = c_1$$
$$a_2x + b_2y = c_2$$

Solution

The solution is:

$$x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}, \quad y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

This method is elegant but becomes computationally expensive for large n, where row-reduction methods are more efficient.

2.4 Geometric Interpretation

Determinants are not only an algebraic tool but also have a **geometric meaning**. For example:

- In **2D**, the absolute value of the determinant of a 2×2 matrix formed by two vectors gives the **area of the parallelogram** spanned by those vectors.
- In **3D**, the absolute value of the determinant of a 3×3 matrix formed by three vectors gives the **volume of the parallelepiped** spanned by those vectors.
- The **sign** of the determinant indicates the **orientation** (whether the vectors preserve or reverse orientation) [1], **lay2012linear?**, **meyer2000?**.

This geometric interpretation provides an intuitive understanding of why a determinant of zero implies **linear dependence** among vectors: the area or volume collapses to zero.

2.4.1 Area in 2D

For two vectors in 2D:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

the determinant of the 2×2 matrix formed by these vectors gives the **signed area** of the parallelogram spanned by **u** and **v**:

$$\det([\mathbf{u}\ \mathbf{v}]) = \begin{vmatrix} u_1 & v_1 \\ u_2 & v_2 \end{vmatrix} = u_1 v_2 - u_2 v_1$$

- The **absolute value** $|\det([\mathbf{u}\ \mathbf{v}])|$ gives the **area**.
- The **sign** indicates the orientation (clockwise or counterclockwise).

Example

A mining engineer is mapping the cross-section of a mineral deposit. Two vectors in the plane represent **edges of a small parallelogram section** of the deposit:

$$\mathbf{u} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}.$$

Determine the **area of the parallelogram** formed by these two vectors.

Solution

The area of the parallelogram is given by the **absolute value of the determinant**:

$$\det([\mathbf{u}\ \mathbf{v}]) = \begin{vmatrix} 2 & 1 \\ 3 & 4 \end{vmatrix} = 2 \cdot 4 - 3 \cdot 1 = 5$$

Thus, the **area** is:

$$\text{Area} = |\det([\mathbf{u}\ \mathbf{v}])| = 5$$

Visualization

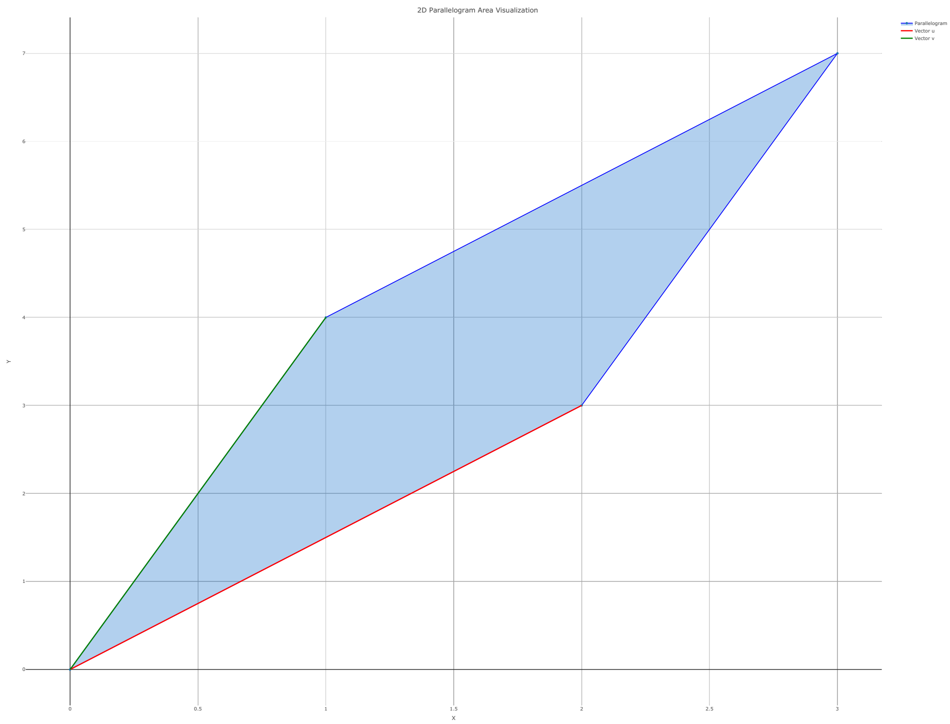

Figure 2.2: Visualization of Parallelogram Formed by Vectors u = [2,3] and v = [1,4]

2.4.2 Volume in 3D

For three vectors in 3D:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix},$$

the determinant of the 3×3 matrix formed by these vectors gives the **signed volume** of the parallelepiped spanned by $\mathbf{u}, \mathbf{v}, \mathbf{w}$:

$$\det([\mathbf{u}\ \mathbf{v}\ \mathbf{w}]) = \begin{vmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ u_3 & v_3 & w_3 \end{vmatrix}$$

- The **absolute value** $|\det([\mathbf{u}\ \mathbf{v}\ \mathbf{w}])|$ gives the **volume**.
- The **sign** indicates the orientation in space (right-hand or left-hand system).

Example: Mining Storage Container

A mining company is designing a **custom-shaped container** to store rare ore. The container is a **parallelepiped** in 3D space, but the edges are **not aligned with the standard axes**. The vectors representing the edges originating from one corner are:

$$\mathbf{u} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}.$$

Tasks:
1. Find the **volume of the container**.
2. Calculate the **area of the parallelogram** formed by edges $\mathbf{v}$ and $\mathbf{w}$. 3. Determine the **height of the parallelepiped** relative to the base formed by $\mathbf{v}$ and $\mathbf{w}$.

Solution

Step 1 – Volume using determinant:
The volume of a parallelepiped formed by vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$ is the absolute value of the determinant of the matrix formed by these vectors:

$$V = \left| \det \begin{bmatrix} 3 & 2 & 1 \\ 1 & 4 & 2 \\ 2 & 1 & 5 \end{bmatrix} \right|$$

Compute the determinant:

$$\det = 3(4\cdot5-2\cdot1)-2(1\cdot5-2\cdot2)+1(1\cdot1-4\cdot2) = 3(18)-2(1)+1(-7) = 54-2-7 = 45$$

So the **volume**:

$$V = |45| = 45 \text{ m}^3$$

Step 2 – Area of the base formed by $\mathbf{v}$ and $\mathbf{w}$:
The area of a parallelogram formed by two vectors is the magnitude of their cross product. Compute the cross product:

$$\mathbf{v} \times \mathbf{w} = \begin{bmatrix} v_2w_3 - v_3w_2 \\ v_3w_1 - v_1w_3 \\ v_1w_2 - v_2w_1 \end{bmatrix} = \begin{bmatrix} 4\cdot5 - 1\cdot2 \\ 1\cdot1 - 2\cdot5 \\ 2\cdot2 - 4\cdot1 \end{bmatrix} = \begin{bmatrix} 18 \\ -9 \\ 0 \end{bmatrix}$$

Magnitude of the cross product:

$$A_{\text{base}} = \|\mathbf{v} \times \mathbf{w}\| = \sqrt{18^2 + (-9)^2 + 0^2} = \sqrt{324 + 81 + 0} = \sqrt{405} \approx 20.12 \text{ m}^2$$

Step 3 – Height relative to the base:
Height of the parallelepiped is:

$$h = \frac{\text{Volume}}{\text{Area of base}} = \frac{45}{20.12} \approx 2.24 \text{ m}$$

Visualization

Figure 2.3: 3D Parallelepiped with Shaded Surfaces

Remark:

Determinants also help determine whether vectors are **linearly independent**:

- If $\det([\mathbf{u}\ \mathbf{v}]) = 0$ in 2D, vectors are collinear.
- If $\det([\mathbf{u}\ \mathbf{v}\ \mathbf{w}]) = 0$ in 3D, vectors are coplanar.

2.5 Invertibility

Determinants provide a quick test for the invertibility of a square matrix [1], **lay2012linear?**, **meyer2000?**:

- **Non-singular matrix ($\det(A) \neq 0$):**
  The matrix A is **invertible**, meaning an **inverse exists**:

$$A^{-1} \text{ exists.}$$

- **Singular matrix ($\det(A) = 0$):**
  The matrix A is **not invertible**, meaning **no inverse exists**:

$$A^{-1} \text{ does not exist.}$$

This directly connects to solving linear systems $A\mathbf{x} = \mathbf{b}$:

- **Non-singular matrix ($\det(A) \neq 0$):**
  The system has a **unique solution**, which can be found using:
  - **Inverse method:**

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

  - **Gaussian elimination / RREF**
- **Singular matrix ($\det(A) = 0$):**
  The system may have:
  - **No solution** (inconsistent system)
  - **Infinitely many solutions** (dependent system)

> Example: Non-singular
>
> $$A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}, \quad \det(A) = 2 \cdot 4 - 1 \cdot 3 = 5 \neq 0$$
>
> - A is invertible.
> - The system $A\mathbf{x} = \mathbf{b}$ has a **unique solution** for any vector $\mathbf{b}$.

> Example: Singular
>
> $$B = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}, \quad \det(B) = 1 \cdot 4 - 2 \cdot 2 = 0$$
>
> - B is **non-invertible**.
> - The system $B\mathbf{x} = \mathbf{b}$ may have **no solution** (if $\mathbf{b}$ is inconsistent) or **infinitely many solutions** (if $\mathbf{b}$ is in the column space of B).

> Remark:
>
> Determinants act as a **shortcut** to check invertibility before attempting more computationally expensive methods like RREF or computing the inverse.

Refrences

Chapter 3

Matrix Inverse

In this chapter, we explore the concept of the **matrix inverse** and its applications. A mind map provides a structured overview of the key topics discussed:

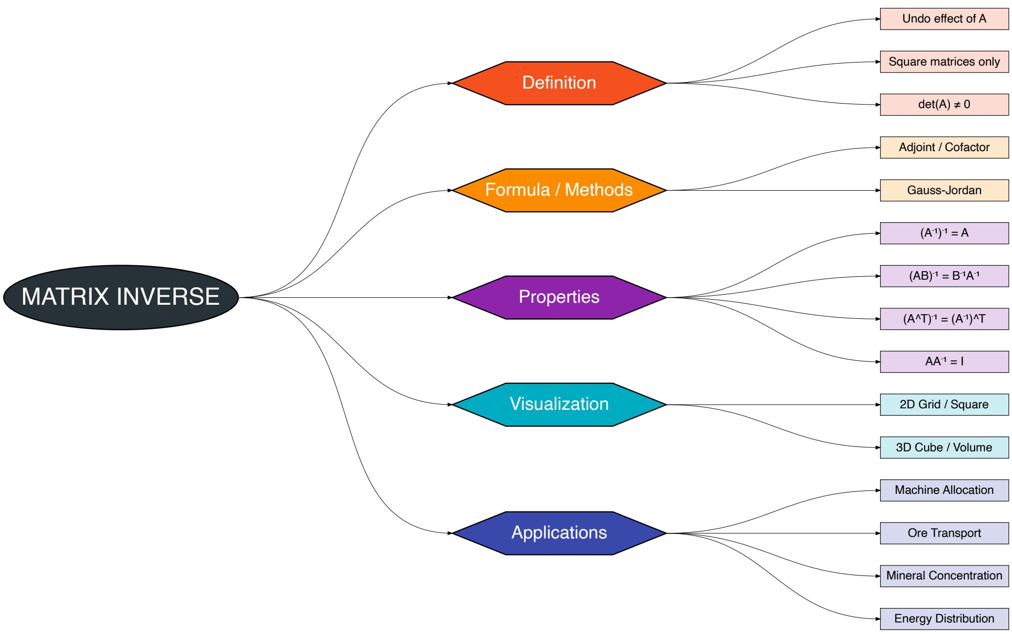

Figure 3.1: Mind Map of Matrix Inverse

The **matrix inverse** is one of the most important concepts in linear algebra [1], [2], **lay2012linear?**. Think of it as the **matrix version of division** for numbers. Just like dividing a number b by a (where $a \neq 0$) gives $x = b/a$, the matrix inverse allows us to "divide" by a matrix to solve equations. It is primarily used for **solving systems of linear equations (SLE)** [3], **meyer2000?**.

For example, if we have $A\mathbf{x} = \mathbf{b}$ where A is a square matrix and $\mathbf{b}$ is a column vector, the solution can be written as $\mathbf{x} = A^{-1}\mathbf{b}$. Here, A^{-1} is the **inverse of matrix** A, which "undoes" the effect of A on $\mathbf{x}$, similar to how $1/a$ undoes multiplication by a for numbers [4], **trefethen1997?**.

⚠ Key Notes:

- The **inverse exists only for square matrices** $(n \times n)$. Rectangular matrices do **not** have inverses in the usual sense.
- Think of A^{-1} as the "undo button" for A:

$$AA^{-1} = A^{-1}A = I_n$$

where I_n is the identity matrix of size $n \times n$. - If a matrix is singular ($\det(A) = 0$), the "undo button" does not exist.

3.1 Inverse Definition

A matrix B is called the **inverse** of a square matrix A if it satisfies:

$$AB = BA = I_n$$

where I_n is the $n \times n$ **identity matrix**.

- **Notation:** $B = A^{-1}$
- **Existence condition:** $\det(A) \neq 0$
- If $\det(A) = 0$, then A is called a **singular matrix**, and the inverse does not exist.

The inverse of a matrix plays the role of "division" in linear algebra: it allows us to solve systems of linear equations efficiently by undoing the effect of multiplication by A **turn0search0?**, **turn0search1?**.

3.2 Inverse Formula

In linear algebra, several methods can be used to compute the inverse of a square matrix. One of the most classical and commonly taught approaches is the **Adjoint (Cofactor) Method**, which expresses the inverse in terms of the determinant and the adjoint of the matrix. This formula provides not only a theoretical foundation but also a practical way to compute inverses for small matrices.

3.2.1 Adjoint / Cofactor Method

For a square matrix $A = [a_{ij}]_{n \times n}$, the inverse is given by:

$$A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A)$$

where $\operatorname{adj}(A)$ is the **adjoint matrix**, obtained by taking the transpose of the cofactor matrix of A.

- **Condition:** $\det(A) \neq 0$
- If $\det(A) = 0$, then A is **not invertible (singular)**.

This method is practical for small matrices (e.g., 2×2 or 3×3), but computationally expensive for large matrices, where Gaussian elimination or LU decomposition is preferred [1], [3], **lay2012linear?**.

Note

Step 1: Adjoint (Adjugate) Matrix
The **adjoint matrix** $\operatorname{adj}(A)$ is defined as the **transpose of the cofactor matrix**:

$$\operatorname{adj}(A) = [C_{ji}]$$

This means:

- Compute all cofactors C_{ij}
- Then take the transpose to form $\operatorname{adj}(A)$.

Step 2 Cofactor
The cofactor C_{ij} is defined as:

$$C_{ij} = (-1)^{i+j} \det(M_{ij})$$

where M_{ij} is the **minor of** A, obtained by removing the i-th row and the j-th column from A.

- The factor $(-1)^{i+j}$ is called the **cofactor sign**.
- The cofactor signs follow a checkerboard pattern:

$$\begin{bmatrix} + & - & + & - & \cdots \\ - & + & - & + & \cdots \\ + & - & + & - & \cdots \\ - & + & - & + & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Step 3 Steps to Compute A^{-1} using the Adjoint Method

1. **Compute the determinant** $\det(A)$. If $\det(A) = 0$, stop (no inverse exists).

2. **Compute all minors** M_{ij} for each element a_{ij}.
3. **Compute the cofactors** using:

$$C_{ij} = (-1)^{i+j} \det(M_{ij})$$

4. **Construct the cofactor matrix** $C = [C_{ij}]$.
5. **Transpose the cofactor matrix** to get $\text{adj}(A)$.
6. **Apply the formula**:

$$A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$$

Example: 2D Matrix

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

- Determinant: $\det(A) = ad - bc$
- Cofactors:
  - $C_{11} = d$, $C_{12} = -c$, $C_{21} = -b$, $C_{22} = a$

Cofactor matrix:

$$C = \begin{bmatrix} d & -c \\ -b & a \end{bmatrix}$$

Adjoint:

$$\text{adj}(A) = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Thus:

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

Example: 3D Matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 1 & 0 & 6 \end{bmatrix}$$

- Determinant:

$$\det(A) = 1(4 \cdot 6 - 5 \cdot 0) - 2(0 \cdot 6 - 5 \cdot 1) + 3(0 \cdot 0 - 4 \cdot 1)$$

$$= 24 - (-10) - 12 = 22$$

- Some cofactors:
  - $M_{11} = \begin{bmatrix} 4 & 5 \\ 0 & 6 \end{bmatrix}$, $\det(M_{11}) = 24 \implies C_{11} = +24$
  - $M_{12} = \begin{bmatrix} 0 & 5 \\ 1 & 6 \end{bmatrix}$, $\det(M_{12}) = -5 \implies C_{12} = +5$
  - $M_{13} = \begin{bmatrix} 0 & 4 \\ 1 & 0 \end{bmatrix}$, $\det(M_{13}) = -4 \implies C_{13} = -4$

(Continue this process for all C_{ij}, then construct the cofactor matrix, transpose it, and finally compute A^{-1}).

⚠ Key Notes:

- The **adjoint method** is practical for **small matrices (2×2, 3×3)**.
- For **larger matrices**, it is usually more efficient to use **Gauss-Jordan elimination** or other numerical methods.

3.2.2 Gauss-Jordan Method

The **Gauss–Jordan method** is one of the most systematic ways to compute the inverse of a square matrix.
Unlike the adjoint/cofactor method, it avoids computing determinants and cofactors, which can become tedious for larger matrices.

If A is an invertible $n \times n$ matrix, then there exists A^{-1} such that:

$$AA^{-1} = I_n$$

To find A^{-1}, we apply **row operations** to reduce A to the identity matrix, while performing the same operations on I_n.
The resulting right-hand side will then be A^{-1} [1], [3], **lay2012linear?**.

ℹ Note:

Step 1: Form the Augmented Matrix
Construct the augmented matrix by placing the identity matrix I_n to the right of A:

$$[A \mid I_n]$$

This creates a block matrix with A on the left and I_n on the right.
Step 2: Apply Gauss–Jordan Elimination
Use elementary row operations to reduce the left block (A) into the identity matrix

I_n. At the same time, apply those same operations to the right block. The goal is to transform:

$$[A \mid I_n] \quad \longrightarrow \quad [I_n \mid A^{-1}]$$

Step 3: Extract the Inverse
Once the left block is the identity matrix I_n, the right block will be the inverse of A:

$$[A \mid I_n] \sim [I_n \mid A^{-1}]$$

Example:

Let

$$A = \begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix}$$

Step 1: Form the augmented matrix

$$[A \mid I_2] = \begin{bmatrix} 2 & 1 & | & 1 & 0 \\ 5 & 3 & | & 0 & 1 \end{bmatrix}$$

Step 2: Apply row operations

1. Make the pivot in the first row equal to 1:
   $R_1 \to \frac{1}{2}R_1$

$$\begin{bmatrix} 1 & 0.5 & | & 0.5 & 0 \\ 5 & 3 & | & 0 & 1 \end{bmatrix}$$

2. Eliminate the 5 below the pivot:
   $R_2 \to R_2 - 5R_1$

$$\begin{bmatrix} 1 & 0.5 & | & 0.5 & 0 \\ 0 & 0.5 & | & -2.5 & 1 \end{bmatrix}$$

3. Scale the second row to make the pivot 1:
   $R_2 \to 2R_2$

$$\begin{bmatrix} 1 & 0.5 & | & 0.5 & 0 \\ 0 & 1 & | & -5 & 2 \end{bmatrix}$$

4. Eliminate the 0.5 above the pivot:
   $R_1 \to R_1 - 0.5R_2$

$$\begin{bmatrix} 1 & 0 & | & 3 & -1 \\ 0 & 1 & | & -5 & 2 \end{bmatrix}$$

Step 3: Extract the inverse

$$A^{-1} = \begin{bmatrix} 3 & -1 \\ -5 & 2 \end{bmatrix}$$

Key Notes:

- This method is highly effective for both **hand calculations (small matrices)** and **computer algorithms (large matrices)**.
- If at any step a pivot element is 0, row exchanges may be needed.
- If the matrix cannot be reduced to I_n, then A is **singular** and has **no inverse**.

3.3 Inverse Properties

The inverse of a matrix has several important properties that are fundamental in linear algebra.These properties describe how inverses behave under various operations such as taking another inverse, multiplying matrices, or transposing them. They also establish the necessary conditions for the existence of an inverse and its relation to the identity matrix **turn0search1?**, **turn0search0?**, **turn0search2?**.

- **Inverse of inverse:** $(A^{-1})^{-1} = A$
- **Inverse of product:** $(AB)^{-1} = B^{-1}A^{-1}$
- **Inverse of transpose:** $(A^T)^{-1} = (A^{-1})^T$
- **Existence:** $\det(A) \neq 0 \Rightarrow A^{-1}$ exists
- **Non-existence:** $\det(A) = 0 \Rightarrow A^{-1}$ does not exist (singular)
- **Identity relation:** $AA^{-1} = A^{-1}A = I_n$

3.4 Inverse Visualizations

The **matrix inverse** can also be understood geometrically. An invertible matrix A represents a **linear transformation** in space. Its inverse A^{-1} reverses that transformation, mapping transformed points back to their original positions.

3.4.1 2D Matrix

The concept of a **matrix inverse** can be better understood in 2D using a **unit square** and its corner points. A 2×2 matrix can be seen as a way to **move points in the plane**. The **inverse matrix** undoes that movement and brings the points back to their original positions.

Example

Imagine working with an **underground mining survey grid**:

- **Original Square:**
  The survey grid starts from $(0,0)$ with corners: $(0,0), (1,0), (0,1), (1,1)$. This represents an **ideal survey map**, perfectly aligned and scaled.

- **After Matrix A:** Due to measurement errors in the field (e.g., compass deviation, magnetic interference, or narrow tunnel conditions), the survey grid may become **skewed or distorted**. The square then turns into a **parallelogram**. In practice, this is like a survey map drawn with errors in scale or orientation.
- **After A^{-1}**
  By applying the **inverse matrix**, the distorted data can be corrected. The parallelogram is restored back to the original square. This is similar to **correcting a survey map** so it matches the proper engineering coordinate system.

Key Notes:

- **Matrix A** = represents the **distortion** in the survey grid.
- **Inverse Matrix A^{-1}** = provides the **correction** that restores the map.
- **Determinant** $\det(A)$ = shows whether the correction is possible.
  - If $\det(A) = 0$, the grid collapses into a line $\rightarrow$ information is lost $\rightarrow$ the map **cannot be corrected**.

Visualization

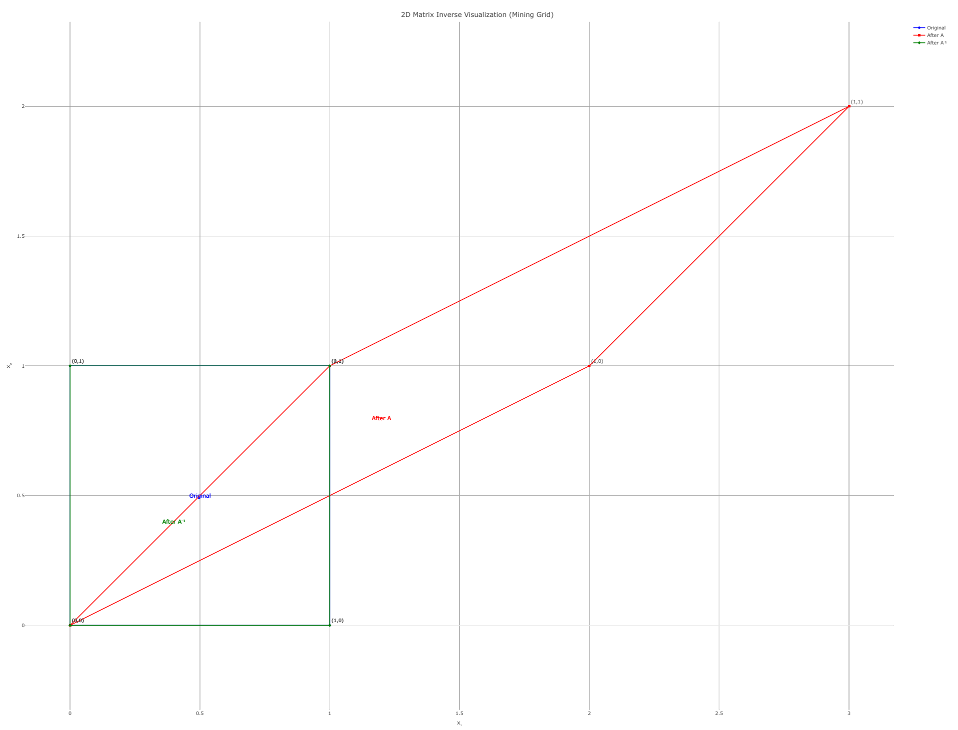

Figure 3.2: 2D Matrix Inverse in Mining Grid

3.4.2 3D Matrix Inverse

A 3×3 matrix in **3D space** is closely related to the concept of **volume** and the **existence of an inverse**:

- If $\det(A) \neq 0$, then A^{-1} exists.
- The determinant $\det(A)$ gives the **volume scaling factor**.
- If $\det(A) = 0$, the **unit cube collapses** into a lower dimension (a plane or a line) $\rightarrow$ no volume $\rightarrow$ no inverse.
- A^{-1} restores the cube to its **original geometry**.

Example

In **mining engineering**, this is important in **block modeling**:

- A modifies the unit cube (mining block).

- $\det(A)$ indicates how much the ore volume is scaled.
- A^{-1} ensures we can **recover the true geometry** for accurate resource estimation.

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 3 \\ 2 & -1 & 1 \end{bmatrix}, \quad A^{-1} = \frac{1}{9} \begin{bmatrix} 4 & -2 & 6 \\ 6 & 1 & -3 \\ -2 & 2 & 1 \end{bmatrix}$$

Key Notes:

- $\det(A) = 9 \quad \Rightarrow \quad$ volume scaled by factor 9.
- **After** A: the original cube is transformed into a **parallelepiped** with 9 times the volume. This represents how a mining block might be distorted due to coordinate transformation or survey error.
- **After** A^{-1}: the parallelepiped is transformed back into the original cube, ensuring that the **true block geometry** is recovered for accurate mine planning and resource estimation.

Visualization

Figure 3.3: Transformation by A and A ^{1} on a Unit Cube

3.5 Applied of Invers

Here are some example problems showing **applications of matrix inversion in mining operations**.

3.5.1 Mining Equipment Allocation

A mine has three types of machines: **Excavator (E)**, **Dump Truck (D)**, and **Conveyor (C)**. Each machine produces a certain output per hour:

Machine	Ore (ton)	Waste (ton)	Energy (kWh)
E	5	2	10
D	2	3	5
C	1	0	3

If the total **ore, waste, and energy required per hour** are:

$$\text{Ore} = 20, \quad \text{Waste} = 15, \quad \text{Energy} = 50$$

Determine the **number of machines of each type** needed using **inverse matrix**.

3.5.2 Ore Transport System

Three locations in a mine have different transport capacities and interdependencies:

$$\begin{cases} x_1 + 2x_2 + x_3 = 100 \\ 2x_1 + x_2 + 3x_3 = 150 \\ x_1 + x_2 + 2x_3 = 120 \end{cases}$$

- x_1, x_2, x_3 = number of trucks used on routes 1, 2, and 3.
  Use **inverse matrix** to determine x_1, x_2, x_3.

3.5.3 Mineral Concentration Modeling

A mineral separation process produces three by-products: **A, B, C**. The relationship between raw material and products is:

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 150 \\ 120 \end{bmatrix}$$

Calculate x_1, x_2, x_3 = raw materials used per hour using A^{-1}.

3.5.4 Energy Distribution

A mine has three sectors: **Excavation, Transportation, and Processing**. Energy (kWh) required in each sector depends on the number of machine units:

$$\begin{bmatrix} 3 & 1 & 2 \\ 2 & 4 & 1 \\ 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 60 \\ 70 \\ 50 \end{bmatrix}$$

Use **inverse matrix** to find the optimal machine distribution x_1, x_2, x_3.

Refrences

Chapter 4

Matrix Factorization

In this chapter, we explore the concept of matrix factorization and its applications. A mind map provides a structured overview of the key topics discussed:

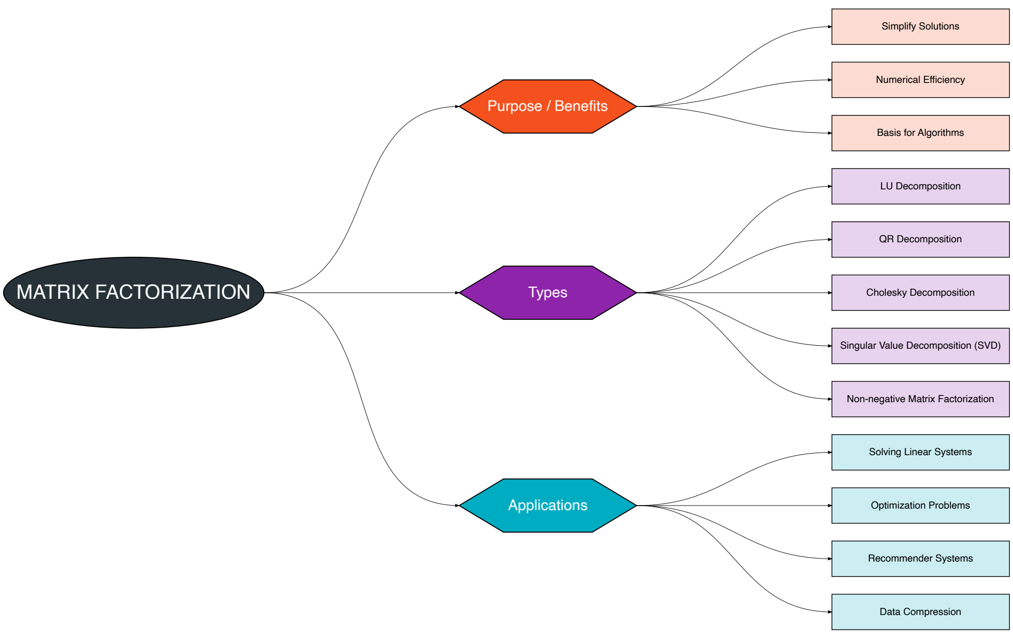

Figure 4.1: Mind Map of Matrix Factorization

Matrix factorization is a powerful tool in linear algebra that decomposes a matrix into a product of two or more matrices. It is widely used in solving linear systems, numerical algorithms, optimization, and data analysis [3], **meyer2000?**.

4.1 Purpose / Benefits

Matrix factorization is useful for several reasons, particularly in simplifying computations and supporting advanced methods in linear algebra.

Reason	**Description**
Simplify Solutions	Decomposing a matrix into simpler matrices can make solving linear systems faster and easier.
Numerical Efficiency	Factorization allows for stable and efficient computations, especially for large matrices.
Basis for Algorithms	Many advanced algorithms in numerical linear algebra rely on matrix factorization as a foundation.

4.2 Types of Matrix Factorization

Several types of matrix factorizations are commonly used:

4.2.1 LU Decomposition

- Decomposes a square matrix A into a product of a lower triangular matrix L and an upper triangular matrix U:

$$A = LU$$

- Useful for solving linear systems and inverting matrices efficiently.

4.2.2 QR Decomposition

- Decomposes a matrix A into an orthogonal matrix Q and an upper triangular matrix R:

$$A = QR$$

- Often used in least squares problems and eigenvalue computations.

4.2.3 Cholesky Decomposition

- Applicable to symmetric, positive-definite matrices.

- Decomposes A as:

$$A = LL^T$$

  where L is a lower triangular matrix.

- Common in optimization and numerical simulations.

4.2.4 Singular Value Decomposition

- Any $m \times n$ matrix A can be decomposed as:

$$A = U\Sigma V^T$$

  where U and V are orthogonal matrices and Σ is diagonal with singular values.

- Widely used in data compression, dimensionality reduction, and recommender systems.

4.2.5 Non-negative Matrix Factorization

- Decomposes a matrix A into matrices W and H with non-negative entries:

$$A \approx WH$$

- Used in feature extraction, text mining, and image processing.

4.3 Applications

Matrix factorization has diverse applications across engineering, data science, and applied mathematics. The table below summarizes key use cases:

Application Area	Description	Methods Commonly Used
Solving Linear Systems	Factorizations such as LU or Cholesky allow faster and more stable solutions to $Ax = b$.	LU, Cholesky
Optimization Problems	Many optimization algorithms rely on matrix factorizations to improve computational efficiency.	QR, Cholesky

Application Area	**Description**	**Methods Commonly Used**
Recommender Systems	SVD and NMF are used in collaborative filtering for predicting user preferences.	SVD, NMF
Data Compression	SVD reduces dimensionality of data while preserving essential structure, widely used in image and signal processing.	SVD

4.3.1 Solving Linear Systems with LU

Suppose we have a linear system:

$$Ax = b, \quad A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ -2 \\ 9 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

LU decomposition:

$$A = LU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 3 \end{bmatrix}$$

Solve in two steps:

1. **Forward substitution** $Ly = b$:

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \\ 9 \end{bmatrix} \implies y_1 = 5,\ y_2 = -12,\ y_3 = 16$$

2. **Backward substitution** $Ux = y$:

$$\begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -12 \\ 16 \end{bmatrix} \implies x_3 = \frac{16}{3},\ x_2 = \frac{1}{6},\ x_1 = \frac{11}{12}$$

Solution:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{11}{12} \\ \frac{1}{6} \\ \frac{16}{3} \end{bmatrix}$$

4.3.2 Optimization Problems

Matrix factorization is widely used in optimization to **improve computational efficiency** and **simplify complex calculations**. Many optimization algorithms require solving linear systems or decomposing matrices multiple times, and factorization helps speed up these computations.

Example: Quadratic Optimization Problem

Suppose we want to minimize a quadratic function:

$$f(x) = \frac{1}{2} x^T Q x - b^T x$$

where Q is a symmetric positive definite matrix, b is a vector, and x is the variable vector:

$$Q = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 5 & 1 \\ 0 & 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Step 1: Factorize Q using Cholesky decomposition

Since Q is symmetric positive definite:

$$Q = LL^T, \quad L = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 0.5 & \sqrt{2.75} \end{bmatrix}$$

Step 2: Solve $Qx = b$ using forward and backward substitution

1. Forward substitution: $Ly = b$

$$\begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 0.5 & \sqrt{2.75} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} \implies y_1 = 1,\ y_2 = -1.5,\ y_3 \approx 2.70$$

2. Backward substitution: $L^T x = y$

$$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0.5 \\ 0 & 0 & \sqrt{2.75} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1.5 \\ 2.70 \end{bmatrix} \implies x_3 \approx 1.63,\ x_2 \approx -1.92,\ x_1 \approx 1.46$$

Solution:

$$x \approx \begin{bmatrix} 1.46 \\ -1.92 \\ 1.63 \end{bmatrix}$$

Key Takeaways:

- Factorization (like LU or Cholesky) allows **efficient repeated solutions** when Q changes slightly, common in iterative optimization.
- Reduces computational cost from $O(n^3)$ to roughly $O(n^2)$ per solve in large systems.
- Ensures **numerical stability**, critical in sensitive optimization problems.

4.3.3 Recommender Systems

Matrix factorization plays a crucial role in **collaborative filtering**, which is widely used in recommender systems to predict user preferences based on historical data.

Problem Setup: User-Item Ratings

Suppose we have a user-item rating matrix R:

$$R = \begin{bmatrix} 5 & 3 & 0 & 1 \\ 4 & 0 & 0 & 1 \\ 1 & 1 & 0 & 5 \\ 0 & 0 & 5 & 4 \end{bmatrix}$$

- Rows represent **users** (u_1, u_2, u_3, u_4)
- Columns represent **items** (i_1, i_2, i_3, i_4)
- A value of 0 indicates a missing rating.

Step 1: Factorize R using low-rank approximation

We approximate R by the product of two smaller matrices:

$$R \approx UV^T$$

where:
- $U \in \mathbb{R}^{4\times 2}$ represents **user features**
- $V \in \mathbb{R}^{4\times 2}$ represents **item features**

Example:

$$U = \begin{bmatrix} 1 & 0.5 \\ 0.9 & 0.4 \\ 0.2 & 1 \\ 0.1 & 0.9 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 0.3 \\ 0.8 & 0.5 \\ 0.2 & 1 \\ 0.5 & 0.7 \end{bmatrix}$$

Step 2: Predict missing ratings

The predicted rating for user u_i and item i_j is:

$$\hat{R}_{ij} = U_i \cdot V_j^T$$

Example: Predict the missing rating for u_1 on item i_3:

$$\hat{R}_{13} = [1, 0.5] \cdot [0.2, 1]^T = 1 * 0.2 + 0.5 * 1 = 0.7$$

Step 3: Recommendation

- Recommend items with **highest predicted ratings** for each user.
- Helps in personalized suggestions even with **sparse data**.

Key Takeaways:

- SVD or NMF (Non-negative Matrix Factorization) are common factorization methods.
- Reduces dimensionality of the rating matrix while capturing latent **user and item features**.
- Enables scalable and accurate recommendations for large datasets.

4.3.4 Data Compression

Matrix factorization is widely used in **data compression** to reduce the dimensionality of datasets while retaining important structure. This is especially common in **images, videos, and signals**.

Problem Setup: Image as a Matrix

Suppose we have a grayscale image represented as a matrix A:

$$A = \begin{bmatrix} 255 & 200 & 180 \\ 240 & 190 & 170 \\ 230 & 180 & 160 \end{bmatrix}$$

Each entry represents the intensity of a pixel (0 = black, 255 = white).

Step 1: Apply Singular Value Decomposition (SVD)

We factorize A into three matrices:

$$A = U\Sigma V^T$$

where:
- U contains **left singular vectors** (features of rows)
- Σ is a diagonal matrix with **singular values** (importance of features)
- V^T contains **right singular vectors** (features of columns)

Example:

$$\Sigma = \begin{bmatrix} 600 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

Step 2: Reduce Rank for Compression

Keep only the largest k singular values (rank-k approximation):

$$A_k = U_k \Sigma_k V_k^T$$

For example, keeping $k = 2$ largest singular values:

$$\Sigma_2 = \begin{bmatrix} 600 & 0 \\ 0 & 50 \end{bmatrix}$$

Step 3: Reconstruct Compressed Data

The compressed matrix A_2 approximates A with fewer data:

$$A_2 \approx \begin{bmatrix} 254 & 199 & 179 \\ 239 & 189 & 169 \\ 231 & 181 & 161 \end{bmatrix}$$

- Notice that the **image is almost identical** but requires less storage.
- Compression ratio improves as k decreases.

Key Takeaways:

- SVD is effective for **low-rank approximation** and **noise reduction**.
- Useful in **image compression**, **signal processing**, and **dimensionality reduction**.
- Retains the **most important features** while discarding less significant information.

Refrences

Chapter 5

Vector Spaces

In this chapter, we explore the concept of **vector spaces** and their fundamental properties. A mind map provides a structured overview of the key topics discussed:

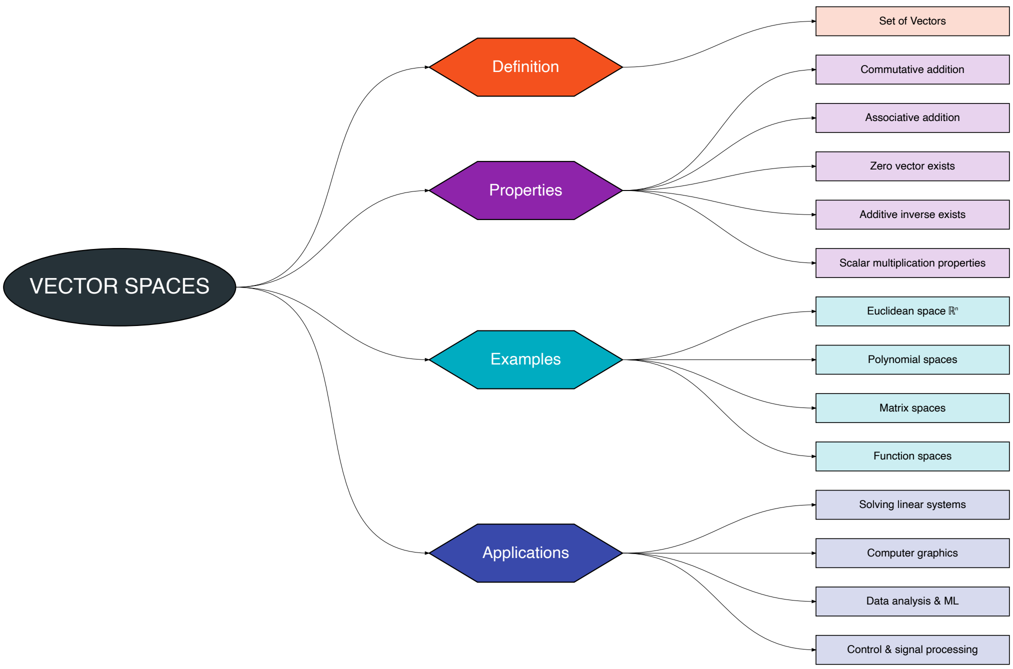

Figure 5.1: Mind Map of Vector Spaces

Vector spaces are a fundamental concept in linear algebra and have significant applications in mining engineering. A vector space is a set of vectors that is closed under vector addition and scalar multiplication, satisfying properties such as commutativity, associativity, the existence of a zero vector, and additive inverses.

In the context of mining:

- **Ore Modeling:** Vectors can represent ore grades at different locations in a mine. This allows for geostatistical analysis and estimation of ore quality across a deposit.

- **Resource Allocation:** Equipment, labor, and material assignments can be expressed as vectors, enabling efficient planning and optimization of mining operations.
- **Geospatial Analysis:** Positions of mining points, boreholes, or sensors are naturally represented in $\mathbb{R}^3$, allowing for 3D modeling, visualization, and spatial transformations.
- **Production Planning:** Production rates over time can be modeled as vectors, facilitating adjustments and optimization of extraction schedules.
- **Risk Assessment:** Safety levels or hazard indices across mine zones can be represented as vectors. Norms of these vectors help quantify overall risk and guide mitigation strategies.

Vector spaces provide a rigorous mathematical framework that underpins **linear system solutions, optimization, and modeling** in mining. They also enable a unified understanding of matrices, polynomials, and function spaces, making them essential for advanced mining analytics and operational decision-making [3], **meyer2000?**.

5.1 Definition

A set V is a **vector space** over a field $\mathbb{F}$ if it satisfies the following conditions:

1. **Closure under addition:**
For all $u, v \in V$, $u + v \in V$.
2. **Closure under scalar multiplication:**
For all $v \in V$ and $c \in \mathbb{F}$, $cv \in V$.
3. **Zero vector exists:**
There exists a vector $0 \in V$ such that $v + 0 = v$ for all $v \in V$.
4. **Additive inverse exists:**
For each $v \in V$, there exists $-v \in V$ such that $v + (-v) = 0$.
5. **Commutativity of addition:**
$u + v = v + u$ for all $u, v \in V$.
6. **Associativity of addition:**
$(u + v) + w = u + (v + w)$ for all $u, v, w \in V$.
7. **Distributivity of scalar multiplication over vector addition:**
$a(u + v) = au + av$.
8. **Distributivity of scalar multiplication over field addition:**
$(a + b)v = av + bv$.

9. **Compatibility of scalar multiplication:**
   $(ab)v = a(bv)$.

10. **Identity element of scalar multiplication:**
    $1v = v$.

    These properties ensure that vector addition and scalar multiplication behave in a consistent and predictable way.

5.2 Properties

Vector spaces exhibit several important properties that can be used to simplify analysis and computations:

Property	Description
Commutative Addition	$u + v = v + u$
Associative Addition	$(u + v) + w = u + (v + w)$
Zero Vector	Exists $0 \in V$ such that $v + 0 = v$
Additive Inverse	For $v \in V$, $-v \in V$ satisfies $v + (-v) = 0$
Scalar Multiplication	$cv \in V$ for $c \in \mathbb{F}$ and $v \in V$
Distributive Laws	$a(u + v) = au + av$, $(a + b)v = av + bv$
Compatibility	$(ab)v = a(bv)$
Multiplicative Identity	$1v = v$

5.3 Examples of Vector Spaces

Vector spaces can take many forms. Common examples include:

1. **Euclidean space $\mathbb{R}^n$**
   The set of all n-dimensional real vectors:

$$\mathbb{R}^n = \{x = [x_1, x_2, ..., x_n]^T : x_i \in \mathbb{R}\}$$

2. **Polynomial spaces**
   All polynomials of degree $\leq n$:

$$P_n = \{p(x) = a_0 + a_1 x + ... + a_n x^n : a_i \in \mathbb{R}\}$$

3. **Matrix spaces**
   All $m \times n$ matrices over $\mathbb{R}$:

$$M_{m\times n} = \{A = [a_{ij}] : a_{ij} \in \mathbb{R}\}$$

4. **Function spaces**
   Continuous functions on an interval $[a, b]$:

$$C[a, b] = \{f : [a, b] \to \mathbb{R}, f \text{ continuous}\}$$

5.4 Applications

Vector spaces provide a powerful framework to model and analyze mining operations. Examples include:

5.4.1 Ore Grade Estimation

Ore grades at different locations in a mine can be represented as vectors:

$$\mathbf{g} = [g_1, g_2, g_3, ..., g_n]^T$$

where g_i is the ore grade at location i.
Using vector operations and interpolation techniques (e.g., Kriging), we can estimate ore quality across the deposit.

5.4.2 Resource Allocation

Equipment, labor, and material allocation can be modeled as vectors:

$$\mathbf{x} = [x_1, x_2, ..., x_m]^T$$

where x_i represents the amount of resource i assigned.
Vector addition and scalar multiplication help in scaling and combining allocation strategies for optimal production.

5.4.3 Geospatial Modeling

Positions of mining points, boreholes, or sensors can be expressed in $\mathbb{R}^3$:

$$\mathbf{p}_i = [x_i, y_i, z_i]^T$$

where (x_i, y_i, z_i) are coordinates of point i.
Operations in vector spaces enable transformations, rotations, and translations for 3D mine modeling and visualization.

5.4.4 Production Planning

Production rates over multiple time periods can be represented as vectors:

$$\mathbf{r} = [r_1, r_2, ..., r_T]^T$$

Vector addition allows combining different schedules, while scalar multiplication adjusts production levels to meet targets efficiently.

5.4.5 Safety and Risk Analysis

Risk levels of different zones in a mine can be modeled as vectors:

$$\mathbf{s} = [s_1, s_2, ..., s_k]^T$$

Vector norms ($\|\mathbf{s}\|$) provide a measure of overall risk, and vector operations help in evaluating mitigation strategies across zones.

Summary:
Using vector spaces, mining engineers can represent complex multi-dimensional data, perform calculations efficiently, and optimize operational decisions. This provides a strong mathematical foundation for geostatistics, production planning, and risk assessment in mining.

References

Chapter 6

Inner Product Spaces

In this chapter, we explore the concept of **inner product spaces** and their fundamental properties. A mind map provides a structured overview of the key topics discussed:

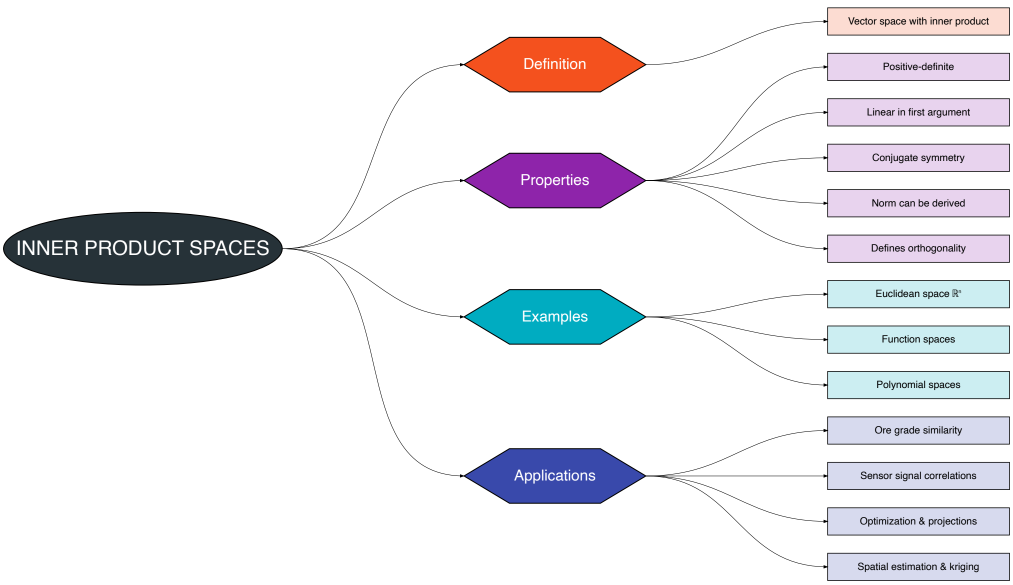

Figure 6.1: Mind Map of Inner Product Spaces

Inner product spaces extend vector spaces by introducing a notion of **length and angle** between vectors. They provide a structured way to measure similarity, correlation, and projections of data represented as vectors.

In the context of mining engineering:

- **Ore Similarity:** Ore grades at different locations can be represented as vectors in $\mathbb{R}^n$. The inner product allows computation of **similarity measures** between samples, aiding in ore classification and blending **chiles2012?**, **rubinstein2016?**.

- **Sensor Signal Analysis:** Signals from monitoring equipment or geotechnical sensors can be treated as vectors. Inner products enable correlation analysis, anomaly detection, and pattern recognition [3].
- **Resource Allocation Optimization:** Tasks such as distributing equipment, workforce, and materials can be modeled as vectors, and inner products can support **least-squares optimization** to minimize cost or maximize efficiency **meyer2000?**.
- **Spatial Estimation / Geostatistics:** Inner products underlie **kriging** and other spatial interpolation methods, enabling accurate estimation of mineral concentrations at unsampled locations **chiles2012?**.
- **3D Modeling and Projections:** Positions of boreholes, tunnels, or ore bodies in $\mathbb{R}^3$ can be analyzed using inner products to compute angles, distances, and orthogonal projections, useful for mine planning and visualization **rubinstein2016?**.

Inner product spaces provide a rigorous way to quantify **length, similarity, and orthogonality** of vectors, which is crucial for **optimization, geostatistical modeling, and operational decision-making** in mining. They extend the utility of vector spaces by enabling precise measurement and projection operations that underpin advanced analytics and planning [3], **meyer2000?**, **chiles2012?**, **rubinstein2016?**.

6.1 Definition

An **inner product space** is a vector space V equipped with an **inner product** $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ (or $\mathbb{C}$) that satisfies:

1. **Positivity:**
$$\langle v, v \rangle \geq 0, \quad \forall v \in V$$
and $\langle v, v \rangle = 0 \iff v = 0$.
2. **Linearity in the first argument:**
$$\langle au + bv, w \rangle = a\langle u, w \rangle + b\langle v, w \rangle$$
for scalars a, b and vectors $u, v, w \in V$.
3. **Conjugate symmetry (for complex spaces):**
$$\langle u, v \rangle = \overline{\langle v, u \rangle}$$
4. **Real symmetry (for real spaces):**
$$\langle u, v \rangle = \langle v, u \rangle$$

6.2 Properties

The inner product induces several important properties:

- **Norm / Length:**

$$\|v\| = \sqrt{\langle v, v\rangle}$$

- **Distance:**

$$d(u, v) = \|u - v\|$$

- **Angle between vectors:**

$$\cos\theta = \frac{\langle u, v\rangle}{\|u\|\|v\|}$$

- **Orthogonality:**
  Two vectors u, v are orthogonal if $\langle u, v\rangle = 0$.

- **Projection:**
  The projection of u onto v is

$$\text{proj}_v(u) = \frac{\langle u, v\rangle}{\langle v, v\rangle} v$$

6.3 Examples

1. **Euclidean space** $\mathbb{R}^n$
   Standard inner product:

$$\langle u, v\rangle = \sum_{i=1}^{n} u_i v_i$$

2. **Function space** $L^2[a, b]$

$$\langle f, g\rangle = \int_a^b f(x)g(x)dx$$

3. **Polynomial space**

$$\langle p, q\rangle = \sum_{i=0}^{n} p(i)q(i)$$

4. **Matrix space**

$$\langle A, B\rangle = \text{trace}(A^T B)$$

6.4 Applications

- **Ore Similarity and Blending:**
  Vectors represent ore grades at different locations. Inner products measure similarity for classification and blending strategies.

- **Sensor Data Analysis:**
  Signals from monitoring sensors can be represented as vectors. Inner products allow **correlation and anomaly detection**.

- **Resource Allocation Optimization:**
  Allocation of equipment, labor, and materials can be modeled as vectors. Inner products are used in **least-squares optimization** to minimize costs or maximize efficiency.

- **Spatial Estimation / Geostatistics:**
  Inner products underpin **kriging** and other interpolation methods for estimating mineral concentrations at unsampled locations.

- **3D Modeling and Projections:**
  Positions of boreholes, tunnels, and ore bodies in $\mathbb{R}^3$ allow calculations of angles, distances, and orthogonal projections, aiding mine planning and visualization.

- **Risk Assessment:**
  Safety indices across mine zones can be represented as vectors. Norms and projections help quantify risk and guide mitigation strategies.

References

Chapter 7

Orthogonality

In this chapter, we explore the concept of orthogonality in vector and inner product spaces, along with its fundamental properties. A mind map provides a structured overview of the key topics discussed:

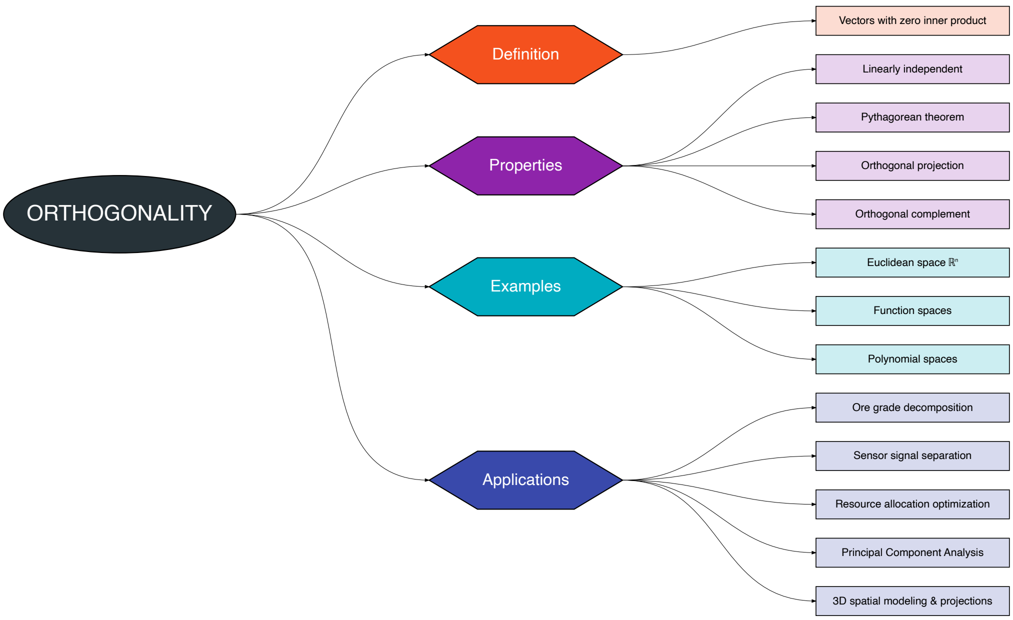

Figure 7.1: Mind Map of Orthogonality in Inner Product Spaces

Orthogonality is a key concept in inner product spaces, describing vectors that are **perpendicular** to each other. Two vectors u and v are orthogonal if their inner product satisfies $u \cdot v = 0$. Orthogonal vectors provide a framework for decomposition, projections, and simplification of vector representations.

In the context of mining engineering:

- **Ore Classification:** Different ore samples can be represented as vectors in $\mathbb{R}^n$.

Orthogonal vectors indicate uncorrelated ore characteristics, which is useful for separating ore types or creating independent quality indices **chiles2012?**.

- **Sensor Data Decorrelation:** Sensor measurements across mine sites can be transformed into orthogonal components to reduce redundancy and highlight independent signal features, aiding in anomaly detection and predictive maintenance [3].

- **Projection for Resource Optimization:** Tasks such as workforce or equipment assignment can be projected onto orthogonal directions to isolate independent effects, supporting **least-squares optimization** and operational efficiency **meyer2000?**.

- **Geostatistical Modeling:** Spatial data vectors can be decomposed into orthogonal components, facilitating kriging, spatial variance analysis, and reducing multicollinearity in predictive models **rubinstein2016?**.

- **3D Spatial Planning:** Orthogonality in $\mathbb{R}^3$ helps in tunnel design, shaft alignment, and modeling ore body orientations, ensuring minimal interference and accurate geometric calculations **chiles2012?**.

Orthogonality provides a robust tool for simplifying **complex vector interactions**, enabling decomposition, projection, and independent analysis. This is essential for **optimization, geostatistical modeling, and operational planning** in mining, extending the analytical capabilities of inner product spaces [3], **meyer2000?**, **chiles2012?**, **rubinstein2016?**.

7.1 Definition

Two vectors u and v in an inner product space V are **orthogonal** if their inner product is zero:

$$\langle u, v \rangle = 0$$

Key points:

- Orthogonal vectors are "perpendicular" in a generalized sense.
- Orthogonality extends to sets of vectors: a set $\{v_1, v_2, \dots, v_n\}$ is orthogonal if $\langle v_i, v_j \rangle = 0$ for $i \neq j$.
- If additionally $\|v_i\| = 1$, the set is **orthonormal**.

7.2 Properties

- **Pythagorean Theorem:**
  If $u \perp v$, then

$$\|u + v\|^2 = \|u\|^2 + \|v\|^2$$

- **Linear Independence:**
  Nonzero orthogonal vectors are linearly independent.

- **Projection:**
  The orthogonal projection of u onto v is

$$\text{proj}_v(u) = \frac{\langle u, v \rangle}{\langle v, v \rangle} v$$

- **Orthogonal Complement:**
  For a subspace $W \subset V$,

$$W^{\perp} = \{v \in V : \langle v, w \rangle = 0, \forall w \in W\}$$

7.3 Examples

1. **Euclidean space $\mathbb{R}^2$ or $\mathbb{R}^3$**
   Standard basis vectors are orthogonal:

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

2. **Function space $L^2[a, b]$**
   Functions f and g are orthogonal if

$$\int_a^b f(x)g(x)dx = 0$$

3. **Polynomial space**
   Polynomials $p_i(x)$ and $p_j(x)$ can be orthogonal under a weighted inner product

$$\langle p_i, p_j \rangle = \int_a^b p_i(x)p_j(x)w(x)dx = 0$$

7.4 Applications

- **Ore Grade Decomposition:**
  Orthogonal vectors can represent independent ore grade variations. This allows separation of correlated and uncorrelated components in ore modeling **chiles2012?**.

- **Signal Processing / Sensor Analysis:**
  Orthogonal signals reduce interference and enable independent feature extraction from geotechnical or seismic sensors [3].

- **Resource Allocation Optimization:**
  Using orthogonal vectors in task allocation ensures **non-overlapping responsibilities**, minimizing redundancy in equipment or workforce assignments **meyer2000?**.

- **3D Spatial Modeling:**
  Orthogonal basis vectors in $\mathbb{R}^3$ support **coordinate transformations**, visualization of tunnels, boreholes, and ore bodies, and accurate geometric calculations for mine planning **rubinstein2016?**.

- **Principal Component Analysis (PCA):**
  Orthogonal directions (principal components) are used to reduce dimensionality of ore grade datasets while preserving maximum variance, aiding in **geostatistical modeling and risk assessment** [3].

References

Chapter 8

Linear Transformations

In this chapter, we explore the concept of **linear transformations** in vector and inner product spaces, along with their fundamental properties. A mind map provides a structured overview of the key topics discussed:

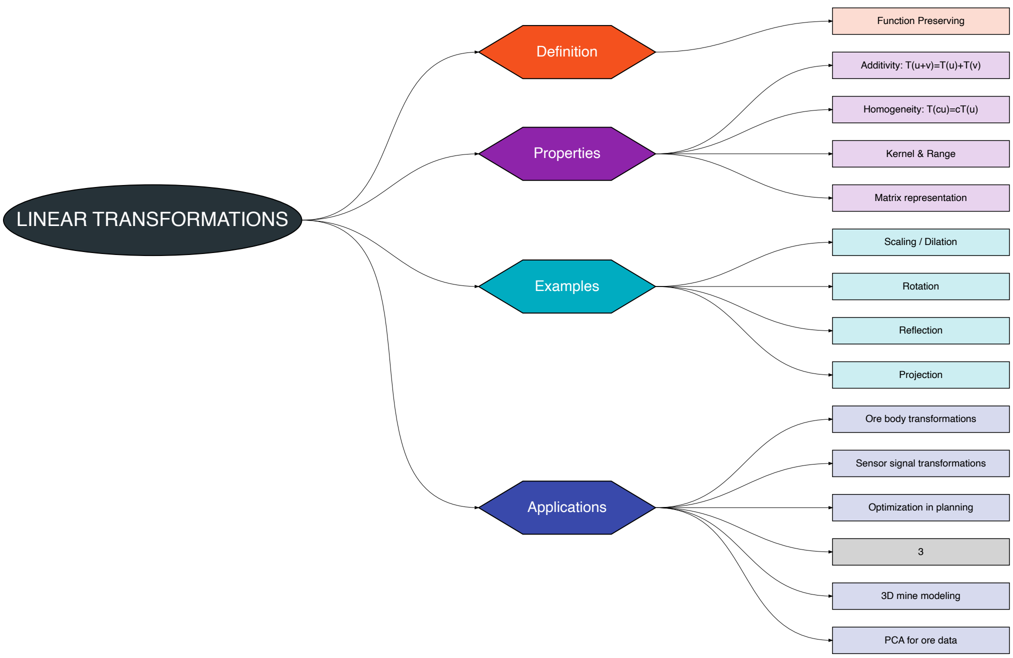

Figure 8.1: Mind Map of Linear Transformations

Linear transformations are functions between vector spaces that preserve vector addition and scalar multiplication. They provide a framework to map vectors from one space to another while maintaining the structure of the space.

In the context of mining engineering:

- **Ore Body Mapping & Alignment:** Linear transformations can represent rotations, scaling, and translations of ore bodies to **align geological survey data** from multiple sources, ensuring consistency in 3D models **chiles2012?**.

- **Geotechnical Sensor Data Processing:** Measurements from strain gauges, accelerometers, or seismic sensors can be **linearly transformed** to extract independent components, filter noise, and improve signal interpretation [3].

- **Production & Resource Planning:** Extraction schedules, equipment deployment, and workforce assignments can be modeled as linear transformations to **optimize operations** under given constraints **meyer2000?**.

- **3D Mine Design & Modeling:** Tunnel layouts, boreholes, and ore body geometries can be transformed in $\mathbb{R}^3$ to enable **visualization, simulation, and coordinate system adjustments** for mine planning **rubinstein2016?**.

- **Dimensionality Reduction of Ore Grade Data:** Linear transformations such as **Principal Component Analysis (PCA)** identify orthogonal directions that maximize variance, helping to simplify datasets, reduce computational complexity, and enhance **geostatistical modeling** [3].

8.1 Definition

A transformation $T : V \to W$ between vector spaces V and W is **linear** if, for all $u, v \in V$ and scalars $c \in \mathbb{R}$:

$$T(u + v) = T(u) + T(v), \quad T(cu) = c\,T(u)$$

Key concepts:

- **Kernel (Null Space):**

$$\ker(T) = \{v \in V : T(v) = 0\}$$

- **Range (Image):**

$$\text{range}(T) = \{T(v) : v \in V\}$$

- **Matrix Representation:**
  Every linear transformation can be represented by a **matrix** relative to chosen bases, enabling computation and composition.

8.2 Properties

- **Additivity:**

$$T(u + v) = T(u) + T(v)$$

- **Homogeneity (Scalar Multiplication):**

$$T(cu) = c\,T(u)$$

- **Kernel & Range:** Describe the null space and image of T.
- **Matrix Representation:** Enables computation, composition, and application of transformations.

8.3 Examples

1. **Scaling / Dilation:**

$$T(x) = kx$$

   Stretches or shrinks vectors by a factor k.
2. **Rotation:**
   Rotates vectors around an origin in $\mathbb{R}^2$ or $\mathbb{R}^3$.
3. **Reflection:**
   Reflects vectors across a line, plane, or hyperplane.
4. **Projection:**
   Projects vectors onto a subspace.

8.4 Applications

- **Ore Body Transformations:**
  Linear transformations can adjust survey data for **modeling and alignment** of ore bodies **chiles2012?**.
- **Sensor Signal Analysis:**
  Transform signals from monitoring equipment for **feature extraction or filtering**, enabling better interpretation of geotechnical or seismic data [3].
- **Optimization of Resources:**
  Apply linear mappings to **planning constraints and allocation strategies** for workforce, equipment, and materials **meyer2000?**.
- **3D Mine Modeling:**
  Transform mine coordinates for **visualization, simulation, and planning** in $\mathbb{R}^3$ **rubinstein2016?**.
- **PCA Transformations:**
  Reduce dimensionality of ore grade datasets while preserving key variance, aiding **geostatistical modeling and predictive analytics** [3].

References

Chapter 9

Eigenvalues

In this chapter, we explore the concept of **eigenvalues and eigenvectors**, their properties, and applications. A mind map provides a structured overview of the key topics discussed:

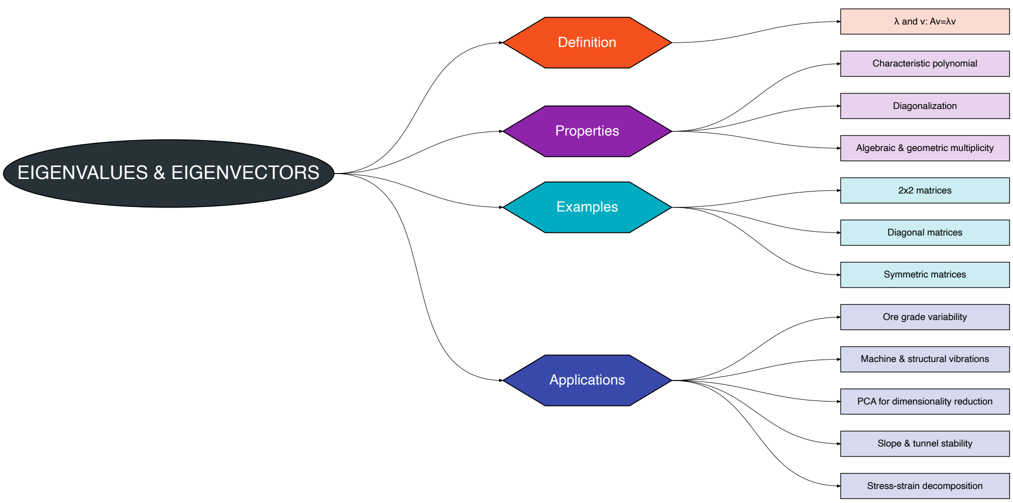

Figure 9.1: Mind Map of Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors are fundamental concepts in linear algebra. For a linear transformation represented by a matrix A, an eigenvector v satisfies:

$$Av = \lambda v$$

where λ is the **eigenvalue** associated with v. Eigenvalues describe **scaling factors** along certain directions (eigenvectors) and are central to understanding the behavior of linear systems.

In the context of mining engineering:

- **Ore Grade Variability:** Eigenvectors can identify principal directions of variation in ore properties, and eigenvalues quantify the magnitude of these variations **chiles2012?**.
- **Machine & Structural Vibrations:** Vibration modes of mining equipment or tunnels correspond to eigenvectors, with eigenvalues representing frequencies, aiding maintenance and safety analysis [3].
- **Principal Component Analysis (PCA):** Eigenvectors of the covariance matrix determine principal components, enabling dimensionality reduction of ore grade datasets while preserving maximum variance [3].
- **Slope & Tunnel Stability:** Eigenvalue analysis of stiffness or stress matrices helps predict failure directions and magnitudes, improving safety and design **meyer2000?**.
- **Stress-Strain Decomposition:** In rock mechanics, eigenvectors of the stress tensor indicate principal stress directions, and eigenvalues quantify stress magnitude for modeling and simulation **rubinstein2016?**.

9.1 Definition

An eigenvector $v \neq 0$ of a square matrix A satisfies:

$$Av = \lambda v$$

where λ is the corresponding **eigenvalue**.

- Eigenvectors define **directions preserved** by the transformation.
- Eigenvalues define **scaling factors** along these directions.

9.2 Properties

- **Characteristic Polynomial:**

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

  Solutions λ are eigenvalues of A.
- **Diagonalization:**
  If A has n linearly independent eigenvectors, A can be written as:

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

where D is a diagonal matrix of eigenvalues and the columns of P are the corresponding eigenvectors.

- **Multiplicity:**
  - **Algebraic multiplicity:** Number of times an eigenvalue appears as a root.
  - **Geometric multiplicity:** Dimension of the eigenspace corresponding to the eigenvalue.

9.3 Examples

1. **2x2 Matrix:**

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

2. **Diagonal Matrix:** Eigenvalues are the diagonal entries, and eigenvectors are standard basis vectors.
3. **Symmetric Matrix:** Real eigenvalues with orthogonal eigenvectors.

9.4 Applications

- **Ore Grade Analysis:** Identify principal directions of variability and correlations between ore properties.
- **Vibration Analysis:** Determine natural frequencies and mode shapes of mining equipment or structures.
- **PCA for Ore Data:** Reduce dimensionality while preserving variance for geostatistical modeling.
- **Stability Analysis:** Assess slope, tunnel, or foundation stability using eigenvalue decomposition of stiffness matrices.
- **Stress-Strain Modeling:** Decompose stress tensors to identify principal stresses and directions in rock mechanics.

References

Chapter 10

Case Studies

In this chapter, we present **case studies** applying all the concepts covered in the previous chapters, from systems of linear equations to eigenvalues, within the context of mining engineering. A mind map provides a structured overview of how the topics interrelate:

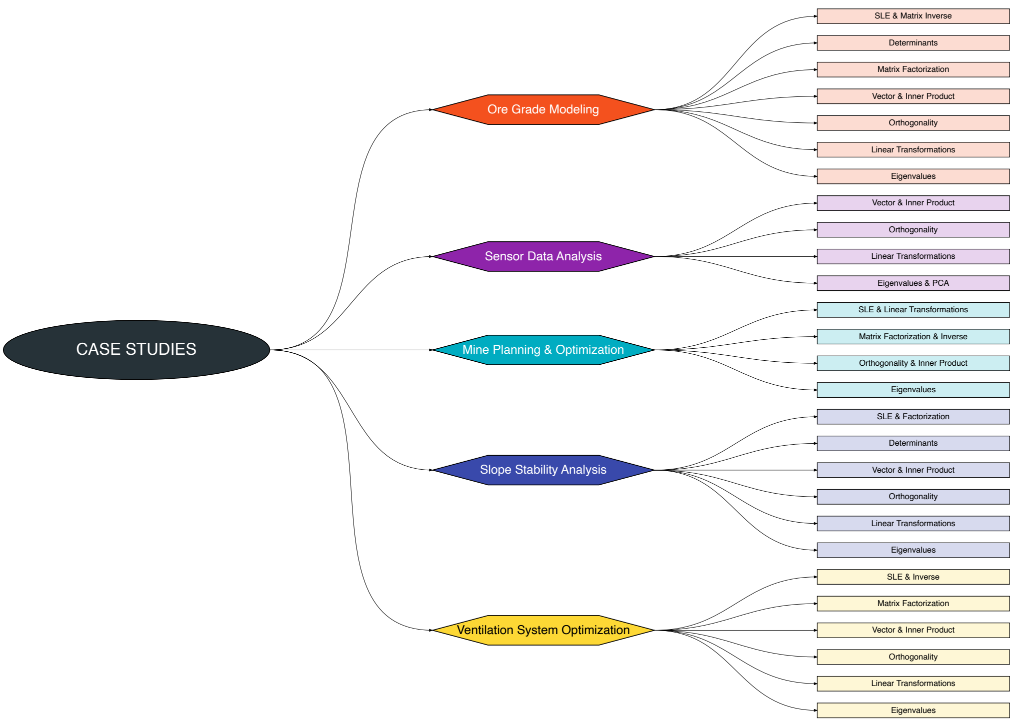

Figure 10.1: Mind Map of Case Studies in Mining Engineering

10.1 Ore Grade Modeling

- **SLE & Matrix Inverse:** Solve for unknown ore grade distributions from sample measurements using linear systems.
- **Determinants:** Check solvability and dependency of ore sampling equations.
- **Matrix Factorization (LU/Cholesky):** Efficiently solve large sparse systems representing ore grade correlations.
- **Vector & Inner Product Spaces:** Represent ore grades as vectors and measure similarity between samples.
- **Orthogonality:** Decompose ore grade vectors into independent components.
- **Linear Transformations:** Rotate or scale ore body data for alignment of surveys.
- **Eigenvalues:** Determine principal directions of ore grade variability for PCA and risk assessment.

10.2 Sensor Data Analysis

- **Vector Spaces & Inner Products:** Model sensor readings as vectors to measure correlations.
- **Orthogonality:** Separate independent signal components for anomaly detection.
- **Linear Transformations:** Filter and transform signals for feature extraction.
- **Eigenvalues & PCA:** Reduce dimensionality while preserving important variance in geotechnical sensor datasets.

10.3 Mine Planning and Optimization

- **SLE & Linear Transformations:** Model resource allocation, scheduling, and material flows.
- **Matrix Factorization & Inverse:** Solve optimization problems efficiently using linear algebra.
- **Orthogonality & Inner Product:** Project tasks onto independent directions to minimize interference.
- **Eigenvalues:** Analyze stiffness matrices, stress tensors, and vibration modes to ensure tunnel stability and equipment safety.

10.4 Slope Stability Analysis

- **SLE & Matrix Factorization:** Solve equilibrium equations for slopes under various load conditions.
- **Determinants:** Check dependency of stress and force equations to ensure solvability.
- **Vector & Inner Product Spaces:** Represent force vectors and displacement vectors in slope models.
- **Orthogonality:** Decompose stress and displacement vectors into independent components.
- **Linear Transformations:** Transform slope models under different coordinate systems for simulation.
- **Eigenvalues:** Determine critical directions of stress and potential failure planes.

10.5 Ventilation System Optimization

- **SLE & Matrix Inverse:** Solve airflow distribution and pressure balance equations in underground tunnels.
- **Matrix Factorization (LU/Cholesky):** Efficiently solve large systems representing complex ventilation networks.
- **Vector & Inner Product Spaces:** Model airflow vectors and velocity correlations.
- **Orthogonality:** Separate independent airflow components for identifying bottlenecks.
- **Linear Transformations:** Simulate airflow changes due to opening/closing of shafts or fans.
- **Eigenvalues:** Identify dominant modes in airflow or pressure fluctuations for system stability and optimization.

[1] Strang, G., Introduction to linear algebra, Wellesley-Cambridge Press, Wellesley, MA, 2016
[2] Anton, H., Elementary linear algebra, Wiley, New York, NY, 2013
[3] Golub, G. H. and Loan, C. F. V., Matrix computations, Johns Hopkins University Press, Baltimore, MD, 2013
[4] Saikia, S., Recent advances in linear algebra applications, *Linear Algebra and Its Applications*, vol. 591, 123–145, 2020

[5] Stewart, J., Calculus: Early transcendentals, Cengage Learning, Boston, MA, 2015

[6] Lay, D. C., Linear algebra and its applications, Pearson, Boston, MA, 2016

[7] Boyce, W. E. and DiPrima, R. C., Elementary differential equations and boundary value problems, *Wiley*, 2017

[8] Samanta, T. and Ghosh, S., Numerical methods for linear systems, *Applied Mathematics and Computation*, vol. 130, 101–115, 2002

[9] Chen, C. T., Linear system theory and design, Oxford University Press, Oxford, UK, 2003