Below is the **expanded and detailed explanation for all sections**, focusing on deeper insights, step-by-step processes, practical examples, and exam-ready clarifications for **Mathematical Foundations for Machine Learning**.

---

1. Linear Systems and Their Solutions

Key Insights

A linear system is a set of linear equations that represent relationships between variables. It can be represented as:

$$Ax = b$$

where:

- A: Coefficient matrix ($m \times n$).
- x: Unknown vector ($n \times 1$).
- b: Output vector ($m \times 1$).

Detailed Case Analysis

Case 1: No Solution

- A system has no solution if b lies outside the column space of A.
- **Example**:

$$x_1 + x_2 = 3, \quad x_1 - x_2 = 1, \quad x_1 + x_2 = 5.$$

Reducing the augmented matrix:

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

The third row implies $0x_1 + 0x_2 = 2$, which is inconsistent.

Case 2: Unique Solution

- Occurs when A has full column rank ($\text{rank}(A) = n$).
- **Example**:

$$x_1 + x_2 = 4, \quad 2x_1 - x_2 = 1.$$

Solving yields:

$$x_1 = 1, \quad x_2 = 3.$$

Case 3: Infinite Solutions

- Occurs when A does not have full column rank, but b lies in the column space.
- **Example**:

$$x_1 + x_2 = 3, \quad 2x_1 + 2x_2 = 6.$$

This system reduces to:

$$x_1 + x_2 = 3, \quad 0 = 0.$$

Hence:

$$x_1 = 3 - x_2, \quad x_2 \text{ is free.}$$

Geometric Interpretation

1. **2D Systems**:
   - Each equation represents a line.
   - Solutions correspond to intersections of lines.
2. **3D Systems**:
   - Each equation represents a plane.
   - Solutions can be points, lines, or planes.
3. **Higher Dimensions**:
   - Equations represent hyperplanes in R^n.

---

2. Splitting a 10 × 5 System

Scenario

For $Ax = b$, where A is 10×5, splitting into two subsystems ($A_1 x = b_1$, $A_2 x = b_2$) seems tempting. However, this is problematic because:

1. Constraints from $A_2 x = b_2$ influence solutions to $A_1 x = b_1$.

2. The combined system may lose consistency if solved separately.

Conclusion: Solve the entire system at once for consistency.

---

3. Row Operations and REF/RREF

Gaussian Elimination

Gaussian elimination reduces A to **Row-Echelon Form (REF)**:

1. Eliminate lower triangular entries.
2. Use pivoting to simplify rows.

Example: Transform:

$$\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ 1 & 2 & -1 \end{bmatrix}$$

to REF:

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

REF to RREF

To transition to Reduced Row-Echelon Form (RREF):

1. Make all pivots 1.
2. Eliminate entries above pivots.

Counting Operations

For an $n \times n$ matrix:

- Gaussian elimination: $O(n^3)$.
- RREF conversion: Additional $O(n^2)$.

---

4. Combined Systems

Key Idea

If $Ax_1 = b_1$ and $Ax_2 = b_2$, then $A(x_1 + x_2) = b_1 + b_2$.

---

5. Non-Zero Matrices with $AB = 0$ and $BA \neq 0$

Example:

1. Let:
$$A = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}.$$
2. Compute:
$$AB = 0, \quad BA \neq 0.$$

6. Positive Definiteness

A matrix $S = A^T A$ is **positive definite** if:

$$x^T Sx > 0, \quad \forall x \neq 0.$$

Proof:

$$x^T Sx = ||Ax||^2 \geq 0.$$

Diagonal entries of S are sums of squares of A's columns, ensuring positivity.

---

7. Determinants

The determinant $\det(A)$ represents the **volume scaling factor** of the transformation A.

Recursive Formula:

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

---

8. Eigenvalues and Eigenvectors

Definition

For $A \in \mathrm{R}^{n \times n}$, λ is an eigenvalue if:

$$Av = \lambda v,$$

where $v \neq 0$ is the eigenvector.

Applications

1. PCA for dimensionality reduction.
2. Stability analysis in systems.

---

9. Singular Value Decomposition (SVD)

Definition

For $A \in \mathrm{R}^{m \times n}$, SVD decomposes A as:

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

where:

- U: Left singular vectors.
- V: Right singular vectors.
- Σ: Diagonal matrix of singular values.

Applications:

1. Dimensionality reduction.

2. Image compression.

---

10. Regularization in Machine Learning

Key Idea

Regularization prevents overfitting by penalizing large model weights.

Techniques:

1. **L2 Regularization (Ridge)**:

$$\text{Loss} = ||Ax - b||^2 + \lambda ||x||^2.$$

2. **L1 Regularization (Lasso)**:

$$\text{Loss} = ||Ax - b||^2 + \lambda ||x||_1.$$