

# Math 2B Notes

**Student:** Nathaniel Santiago

**Professor:** Jeff Anderson

## Lesson 5: Inner Products and Vector Norms

### Section 2.3 — Inner Products

#### What an Inner Product Is

- Think of an inner product (aka “dot product”) as a way to measure how **aligned** two vectors are.
- For two vectors in  $\mathbb{R}^n$ :

$$x \cdot y = x_1y_1 + x_2y_2 + \cdots + x_ny_n$$

→ The result is *just a number*, not a vector.

#### Why We Care

- Dot products let us:
  - Measure **similarity** between two data vectors (e.g., your exam scores vs. the weight vector)
  - Compute **projections, angles, and lengths**
  - Build linear models using **weighted sums**

#### Example from notes (grade model)

- Your class scores become a vector

$$g = [q/300, e1/100, e2/100, f/100]^T$$

- The weight vector is

$$c = [0.10, 0.25, 0.25, 0.40]^T$$

- Final percentage =  $g \cdot c$
- → literally a weighted sum via dot product.

#### Inner Products & Riemann Sums

- Approximating integrals numerically can also be written as a dot product:

$$\sum f(x_i^*)h = f \cdot h$$

where  $f = [f(x_1^*), \dots, f(x_n^*)]^T$  and  $h = [h, \dots, h]^T$

- So dot products show up even in calculus when we discretize things.

### Key Takeaways — Section 2.3

- **Dot product** = multiply corresponding components + add.
- Gives a **number** that represents **alignment**.
- Many real-world **weighted formulas** are just dot products.
- **Numerical integration** (area under curves) can be written as a dot product.
- Inner products are the **foundation** for norms, projections, orthogonality, and least-squares.

### ISE Connection — Section 2.3

- **Weighted decision-making:** Inner products are exactly how multi-criteria decision models compute scores. (Think: supplier ranking, job scheduling, prioritizing tasks.)
- **Performance modeling:** Your grade calculation is basically how ISE models combine different KPIs (e.g., cost, time, reliability).
- **Demand forecasting / predictive models:** Linear prediction = dot product between “feature vector” and “weight vector”.
- **Cost or risk aggregation:** Dot products combine many small factors into one meaningful metric.

## Lesson 6: Linear Combinations, Span, and Linear (In)Dependence

### Linear Combinations

A **linear combination** of vectors is basically when you take some vectors, scale each one, and add them up. Formally, if

$$a_1, \dots, a_n \in \mathbb{R}^m, \quad x_1, \dots, x_n \in \mathbb{R},$$

then a vector  $b$  is a linear combination if

$$b = x_1a_1 + x_2a_2 + \dots + x_na_n.$$

**Idea:** multiply the vectors by scalars → add them all together → get one final vector.

### Example Problem

Given

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad a_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

write

$$\begin{pmatrix} 3 \\ -2 \\ 5 \end{pmatrix}$$

as a linear combination.

**Solution:**

$$3a_1 + (-2)a_2 + 5a_3.$$

## Span

The **span** of a set of vectors is the set of *all* possible linear combinations of those vectors:

$$\text{Span}\{a_1, \dots, a_n\} = \{x_1a_1 + \dots + x_na_n : x_k \in \mathbb{R}\}.$$

**Geometric intuition:**

- 1 nonzero vector  $\rightarrow$  line through the origin.
- 2 non-collinear vectors  $\rightarrow$  plane through the origin.
- More vectors could span the whole space or still something smaller.

## Example Problem

Let

$$v = \begin{pmatrix} 5 \\ 2 \end{pmatrix}.$$

Find  $\text{Span}\{v\}$ .

**Solution:**

$$\text{Span}\{v\} = \left\{ t \begin{pmatrix} 5 \\ 2 \end{pmatrix} : t \in \mathbb{R} \right\},$$

a line with slope  $2/5$ .

## Linear Dependence

A set of vectors is **linearly dependent** if one of the vectors can be written as a linear combination of the others. Equivalently, the set is dependent if there exist scalars, not all zero, such that

$$x_1a_1 + x_2a_2 + \dots + x_na_n = 0.$$

## Quick ways to check dependence

- Includes the **zero vector**  $\Rightarrow$  automatically dependent.
- More vectors than components ( $n > m$ )  $\Rightarrow$  guaranteed dependent.
- One vector is a **scalar multiple** of another.

### Example Problem

Are the vectors

$$a_1 = \begin{pmatrix} 1 \\ -3 \end{pmatrix}, \quad a_2 = \begin{pmatrix} -3 \\ 9 \end{pmatrix}$$

linearly dependent?

**Solution:**

$$a_2 = -3a_1,$$

so they are dependent.

### Linear Independence

A set of vectors is **linearly independent** if none of the vectors can be created from the others. Formally, the equation

$$x_1a_1 + \cdots + x_na_n = 0$$

has only the **trivial solution**

$$x_1 = x_2 = \cdots = x_n = 0.$$

### Intuition

Each vector adds *new* information—none of them are redundant.

### Example Problem

Are the vectors

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

linearly independent?

**Yes**—no combination of two can create the third.

### Checking Whether a Vector is a Linear Combination

To check if a vector  $b$  is in the span of  $a_1, a_2$ , solve

$$x_1a_1 + x_2a_2 = b.$$

### Example Problem

Determine whether

$$b = \begin{pmatrix} 2 \\ -2 \\ -4 \end{pmatrix}$$

is a linear combination of

$$a_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

We solve

$$x_1 \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \\ -4 \end{pmatrix}.$$

This gives the system:

$$\begin{cases} x_1 = 2 \\ x_1 + x_2 = -2 \\ x_2 = -4 \end{cases}$$

**Check:** Middle equation:  $2 + (-4) = -2$ .

**Therefore, yes,**  $b$  is a linear combination.

## Self-Check Questions

### Linear Combinations

- Can I explain what a linear combination is in simple words?
- Can I write a target vector as a combination of given vectors?

### Span

- Can I visualize the span of 1, 2, or 3 vectors?
- Do I know how to check whether a vector lies in a given span?

### Linear Dependence

- Can I recognize dependence using shortcuts (scalar multiple, zero vector,  $n > m$ )?
- Can I solve the equation  $x_1a_1 + \dots + x_na_n = 0$ ?

### Linear Independence

- Can I explain independence as “no redundant vectors”?
- Can I confirm independence using the trivial-solution test?

### Modeling Concepts

- If given a physical system (RGB, circuits, temperature), can I identify the vectors?
- Can I express the model as a linear combination?

## Lesson 7: Matrices and Matrix Modeling

### What is a Matrix?

A **matrix** is just a rectangular grid of numbers. If a matrix has  $m$  rows and  $n$  columns, we say its dimensions are  $m \times n$ . Every individual number in the matrix is called an **entry** (or element).

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

### Key Points

- Vectors are matrices too — they're just  $m \times 1$  or  $1 \times n$ .
- The **row dimension** is the number of rows  $m$ ; the **column dimension** is the number of columns  $n$ .

### Incidence Matrices (Graphs)

Matrices are great for storing **connectivity data**. Graphs have:

- Nodes/vertices
- Edges connecting pairs of nodes

We can use a matrix to encode which nodes each edge touches.

### Undirected Incidence Matrix

For an undirected graph:

$$a_{ik} = \begin{cases} 1 & \text{if edge } e_i \text{ touches node } u_k, \\ 0 & \text{otherwise.} \end{cases}$$

Each **row** = edge, each **column** = node.

### Example: Undirected Graph

Six edges, four nodes:

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

This matrix encodes all graph connectivity information.

### Directed Incidence Matrix

For a directed graph, entries become:

$$a_{ik} = \begin{cases} 1 & \text{edge leaves node } u_k, \\ -1 & \text{edge enters node } u_k, \\ 0 & \text{otherwise.} \end{cases}$$

**Sign** matters here (unlike the undirected case).

### Example: Directed Graph

For a small digraph with 5 edges and 4 nodes:

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

### Wireframe Models (Computer Graphics)

Matrices can store **geometric models**. A **vertex matrix** stores coordinates of each point (vertex). An **edge table** lists which vertices are connected.

### 2D Wireframe Example (Triangle)

Vertices:

$$v_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Vertex matrix:

$$V = \begin{pmatrix} 0 & -1 & 2 \\ 0 & 1 & 1 \end{pmatrix}$$

Edges:

- 1: 1 → 2
- 2: 2 → 3
- 3: 3 → 1

This gives a **wireframe** triangle — no faces, just edges and points.

## 3D Wireframe Models

Same idea as 2D, but each vertex has 3 coordinates.

Example: square-based pyramid:

$$v_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, v_2 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}, v_3 = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}, v_4 = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}, v_5 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}.$$

Vertex matrix:

$$V = \begin{pmatrix} 0 & 2 & 2 & 0 & 1 \\ 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

Edges listed in an edge table tell which vertices connect.

## Polygon Mesh Models

A **mesh model** includes:

- Vertex list
- Face list (each face = triangle given by 3 vertices)

Example face table:

Face 1: (1, 2, 4)  
Face 2: (2, 3, 4)  
Face 3: (1, 2, 5)

...

These are used for advanced graphics (cars, animals, 3D models).

## Entries of a Matrix

Every matrix entry is determined by:

- Row index  $i$
- Column index  $k$
- Value  $a_{ik}$

## Entry Operator

$$\text{Entry}_{ik}(A) = a_{ik}$$

This just picks out the value in position  $(i, k)$ .

## Example

If  $A \in \mathbb{R}^{6 \times 6}$ , then

$$\text{Entry}_{5,3}(A) = a_{53}.$$

## Matrices as Digital Images

A grayscale digital image can be stored as a matrix  $A$ , where each entry is a pixel value.

- Rows = vertical pixel position
- Columns = horizontal pixel position
- Entry value = brightness

Example: 3-bit grayscale image (values 0–7).

The matrix entries produce blocky pixel-art images when plotted.

## Equal Matrices

Two matrices are **equal** if:

- They have the same dimensions
- Every matching entry is equal

Example: Two graphs with identical connectivity will have the same incidence matrix — even if drawn differently.

## Self-Check Questions

### Matrix Basics

- Can I identify the row and column dimensions of any matrix?
- Can I explain what an entry  $a_{ik}$  represents?

### Incidence Matrices

- Do I know the difference between undirected and directed incidence entries?
- Given a graph, can I build its incidence matrix?

## Wireframe Models

- Can I create a vertex table and edge table for a shape?
- Can I write a vertex matrix from coordinates?

## Polygon Meshes

- Can I explain why faces require 3 vertices?
- Do I understand how faces and vertices define a 3D object?

## Digital Images

- How does digitization turn a physical image into a matrix?
- What does each entry represent in a digital image matrix?

# Lesson 8: Anatomy of Matrices

## Matrix Shapes

A matrix  $A \in \mathbb{R}^{m \times n}$  has  $m$  rows and  $n$  columns.

- Tall and narrow:  $m > n$
- Square:  $m = n$
- Short and wide:  $m < n$

The shape helps determine the method used: tall matrices appear in least squares, square matrices in linear systems, and wide matrices in underdetermined problems.

**Confused:** Why does the shape of a matrix affect which method we use?

## Subscript Notation

Use subscripts to show dimensions:  $A_{m \times n}$ .

Each element  $a_{ik}$  represents a scalar at row  $i$ , column  $k$ .

**Confused:** How can checking dimensions prevent multiplication errors?

## Entries of a Matrix

- Zero entry:  $a_{ik} = 0$

- Nonzero entry:  $a_{ik} \neq 0$
- Leading entry: first nonzero element in a row

$$\text{numel}(A) = m \times n$$

$$\text{nnz}(A) = \text{number of nonzero entries}$$

## Sparsity Structure

Matrix symbols:

$$1 \rightarrow \text{entry} = 1, \quad \iota \rightarrow \text{nonzero}, \quad \diamond \rightarrow \text{any value}, \quad 0 \rightarrow \text{zero}$$

**Confused:** What advantage does sparsity provide for large computations? And doesn't this seem like a method used to reduce data storage like downloading images as a JPEG?

## Main Diagonal

Entries where  $i = k$  lie on the main diagonal.

- Diagonal matrix: only diagonal entries nonzero.
- Identity matrix: ones on the diagonal, zeros elsewhere.

**Confused:** Why is the identity matrix the multiplicative identity in matrix algebra?

## Triangular Matrices

- Lower-triangular:  $a_{ik} = 0$  for  $i < k$
- Unit lower-triangular: diagonal entries = 1
- Upper-triangular:  $a_{ik} = 0$  for  $i > k$

**Confused:** Why are triangular matrices useful in LU decomposition?

## Bands of a Matrix

A band is the set of entries where  $i - k = d$ .

- $d = 0$ : main diagonal
- $d = 1$ : superdiagonal

- $d = -1$ : subdiagonal

**Confused:** How does bandwidth affect computational efficiency?

### Colon Notation (MATLAB Style)

- $A(:, k)$  →  $k^{\text{th}}$  column vector
- $A(i, :)$  →  $i^{\text{th}}$  row vector

**Confused:** Why is colon notation useful in coding?

### Row and Column Partitions

$$A = \begin{bmatrix} A(1, :) \\ A(2, :) \\ \vdots \\ A(m, :) \end{bmatrix} \quad A = [A(:, 1) \ A(:, 2) \ \dots \ A(:, n)]$$

Each row is  $1 \times n$ , each column is  $m \times 1$ .

**Confused:** How does the column partition help interpret  $A\vec{x}$  as a linear combination?

### Outer Products and Matrix Units

$$xy^T = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ x_2y_1 & x_2y_2 & \cdots & x_2y_n \\ \vdots & & \ddots & \vdots \\ x_my_1 & x_my_2 & \cdots & x_my_n \end{bmatrix}$$

Matrix unit:  $E_{ik} = e_i e_k^T$ , all zeros except 1 at position  $(i, k)$ .

**Confused:** How can any matrix be represented as a sum of outer products?

## Lesson 9: Matrices from Outer Products and Operations

### Outer Product

Each row of  $xy^T$  equals  $x_iy^T$ ; each column equals  $y_kx$ . Used to build matrices and rank-one systems.

**Confused:** What's the difference between outer and inner products?

### Matrix Addition and Scalar Multiplication

$$(A + B)_{ik} = a_{ik} + b_{ik}, \quad (\alpha A)_{ik} = \alpha a_{ik}$$

#### Properties:

- Commutativity:  $A + B = B + A$
- Associativity:  $A + (B + C) = (A + B) + C$
- Additive identity:  $A + 0 = A$
- Distributivity:  $\alpha(A + B) = \alpha A + \alpha B$

**Confused:** Why must matrices be the same size to be added?

### Rank-One Updates

$$A + xy^T$$

Efficient way to modify matrices using a low-rank adjustment.

#### Examples:

- Shear:  $S_{ik}(c) = I + ce_i e_k^T$
- Dilation:  $D_j(c) = I + (c - 1)e_j e_j^T$
- Transposition: swaps two rows or columns

**Confused:** Why are rank-one updates efficient for large matrices?

### Special Matrix Types

- Shear  $\rightarrow S_{ik}(c) = I + ce_i e_k^T$
- Dilation  $\rightarrow D_j(c) = I + (c - 1)e_j e_j^T$
- Transposition  $\rightarrow P_{ik} = e_i e_k^T + e_k e_i^T + \sum_{j \neq i, k} e_j e_j^T$

- Givens Rotation  $\rightarrow$  rotates in the  $i, k$  plane
- Gauss Transform  $\rightarrow L_k = I - ve_k^T$

**Confused:** What's the geometric difference between a Givens rotation and a shear?

### Transpose of a Matrix

$$A^T : \text{swap rows and columns}$$

**Properties:**

- $(A^T)^T = A$
- $(A + B)^T = A^T + B^T$
- $(cA)^T = cA^T$
- $(AB)^T = B^T A^T$

**Confused:** Why does the order reverse under transposition?

## Lesson 8 & 9: Example Problems and Understanding

### Example 1: Finding the type of Matrix and its shape

Let

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

- Dimensions:  $A \in \mathbb{R}^{3 \times 3}$
- It has zeros below the main diagonal  $\rightarrow$  upper-triangular matrix
- Main diagonal entries are {2, 3, 1}

Since this matrix has zeros below the main diagonal entry of 2, 3, 1, it's an upper-triangular matrix.

Triangular matrices are nice because you can solve systems quickly using substitution.

### Example 2: Using Colon Notation

Given

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

Then:

$$A(2, :) = [4 5 6], \quad A(:, 3) = \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix}$$

From what I can understand, colon notation could help with isolating rows or columns in something like MATLAB or python,

### Example 3: Building a Matrix with an Outer Product

Let

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad y = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

Compute the outer product:

$$xy^T = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} [4 5] = \begin{bmatrix} 4 & 5 \\ 8 & 10 \\ 12 & 15 \end{bmatrix}$$

Outer products make rectangular matrices by multiplying a column by a row. It's the opposite of an inner product, which makes a scalar.

### Example 4: Creating a Shear Matrix

Let's make a shear in  $\mathbb{R}^3$ :

$$S_{13}(2) = I_3 + 2e_1 e_3^T$$

Since

$$e_1 e_3^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

we get

$$S_{13}(2) = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**My understanding:** This adds twice the third column into the first. If I multiply  $S_{13}(2)$  by a vector, it “shears” the space — tilting it slightly.

### Example 5: Rank-One Update

Given

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

Then

$$xy^T = \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}, \quad A + xy^T = \begin{bmatrix} 4 & 4 \\ 6 & 9 \end{bmatrix}$$

**My understanding:** A rank-one update changes the matrix in one direction (like adding a “weighted plane” to it). It’s useful in optimization and iterative algorithms.

### Example 6: Matrix Transpose Properties

Let

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

Then:

$$(A + B)^T = \begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 \\ 1 & 4 \end{bmatrix}$$

and

$$A^T + B^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

**My understanding:** It checks out:  $(A + B)^T = A^T + B^T$ . The transpose just swaps rows and columns without changing the actual data pattern.

### Example 7: Understanding the Identity Matrix

Let

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} 5 \\ -3 \\ 2 \end{bmatrix}$$

Then  $I_3x = x$ .

**My understanding:** The identity matrix doesn't change any vector. It's like multiplying a number by 1 but for matrices.

### Example 8: Diagonal Matrix and Dilation

Let  $D_3(5) = I_3 + (5 - 1)e_3e_3^T$ :

$$D_3(5) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

**My understanding:** This scales the third coordinate by 5 — a “dilation” that stretches along one axis.

### Example 9: Constructing from Matrix Units

$$E_{12} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad E_{21} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

Then any 2x2 matrix

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

can be written as

$$A = aE_{11} + bE_{12} + cE_{21} + dE_{22}$$

**My understanding:** Matrix units are like “building blocks” — I can form any matrix by combining them.

## Lesson 10: Matrix-Vector Mult

### Matrix Inverses

A square matrix  $A$  is called **invertible** (or non-singular) if there exists another matrix  $A^{-1}$  such that multiplying  $A$  by  $A^{-1}$  (on either side) gives the identity matrix  $I$ :

$$AA^{-1} = I \quad \text{and} \quad A^{-1}A = I$$

If no such matrix exists,  $A$  is **singular**.

**Key points:**

- Only square matrices  $(n \times n)$  can have two-sided inverses.
- The inverse is unique for invertible matrices.
- You can think of the inverse as "undoing" the multiplication by  $A$ .

**Inverses of Elementary Matrices:** Elementary matrices (like shear, swap, and scale matrices) are cool because they are always invertible, and their inverses have very simple forms:

- Shear matrix  $S_{ik}(c)$  inverse is  $S_{ik}(-c)$ .
- Transpose (swap) matrix  $P_{ik}$  is its own inverse:  $P_{ik}^{-1} = P_{ik}$ .
- Diagonal scaling matrix  $D_j(c)$  inverse is  $D_j(1/c)$ .

**Inverse of a 2x2 Matrix:** For a  $2 \times 2$  matrix  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ , if the determinant  $\det(A) = ad - bc \neq 0$ , the inverse is:

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

This formula shows the importance of the determinant — if it's zero, no inverse exists.

### The Invertible Matrix Theorem (IMT)

This theorem collects a ton of **equivalent conditions** that all describe when a square matrix  $A$  is invertible. If any one of these is true, all of them are true.

**Key conditions (IMT):**

- $A$  is invertible.
- $A$  has  $n$  pivot positions (so no zero pivots in elimination).
- The equation  $A\vec{x} = \vec{0}$  only has the trivial solution  $\vec{x} = \vec{0}$ , meaning columns of  $A$  are linearly independent.

- Any  $\vec{b}$  has a unique solution for  $A\vec{x} = \vec{b}$ .
- The columns of  $A$  span  $\mathbb{R}^n$ .
- $A^T$  is also invertible.
- The determinant of  $A$  is not zero.
- $A$  has full rank ( $\text{rank}(A) = n$ ).
- Zero is not an eigenvalue of  $A$ .

## LU Factorization Without Pivoting

LU Factorization is a way to write a square matrix  $A$  as the product of two special matrices:  $A = LU$ .

- $L$ : a lower-triangular matrix with 1's on the diagonal (called unit lower-triangular).
- $U$ : an upper-triangular matrix.

This is super useful because solving  $A\vec{x} = \vec{b}$  becomes two easier problems:

1. First solve  $L\vec{y} = \vec{b}$  for  $\vec{y}$  by forward substitution.
2. Then solve  $U\vec{x} = \vec{y}$  for  $\vec{x}$  by backward substitution.

### How to get LU (Without Pivoting):

- Use a series of elementary row operations to turn  $A$  into an upper-triangular matrix  $U$ .
- The lower-triangular matrix  $L$  holds the multipliers used in the elimination process.
- $L$  is the product of the inverses of the elementary matrices that reduce  $A$  to  $U$ .

### Why is LU Factorization so handy?

- If you have to solve many systems with the same  $A$  but different  $\vec{b}$  vectors, you can reuse  $L$  and  $U$  — saves tons of work.
- It also provides insights into the structure of  $A$  and helps with understanding numerical stability and algorithms.

## Lesson 11 Notes: Matrix–Matrix Multiplication

Matrix–matrix multiplication is really just a scaled-up version of matrix–vector multiplication. Instead of applying a matrix to one vector, we apply it to a whole *set* of vectors (the columns or rows of another matrix). There are **four main ways** to think about matrix multiplication, and even though they all give the same result, each one is useful in different situations.

### Conformability

Given

$$A \in \mathbb{R}^{m \times p}, \quad X \in \mathbb{R}^{p \times n},$$

the product  $AX$  exists only if the **inner dimensions match**. In other words:

$$\text{columns of } A = \text{rows of } X.$$

If this condition holds, the resulting matrix has the **outer dimensions**:

$$AX \in \mathbb{R}^{m \times n}.$$

**Key phrase:** “Inner dimensions must agree. Outer dimensions become the result.”

#### 1. Linear Combination of Columns

This viewpoint says each column of the product is built from linear combinations of columns of  $A$ :

$$B(:, k) = AX(:, k).$$

**Meaning:** The  $k$ th column of  $B$  uses the entries of the  $k$ th column of  $X$  as weights on the columns of  $A$ .

##### Example: Scaling One Column

Given

$$A = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \quad D_1(2) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

the product  $AD_1(2)$  **doubles column 1** and leaves the others unchanged. This happens because the first column of  $D_1(2)$  is  $(2, 0, 0)^T$ , meaning:

$$2 \cdot \text{col}_1(A) + 0 \cdot \text{col}_2(A) + 0 \cdot \text{col}_3(A).$$

#### 2. Linear Combination of Rows

In  $XA$ , each row of the output is a linear combination of the rows of  $A$ :

$$B(i, :) = X(i, :) A.$$

### Example: Row Scaling

Let

$$D = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}.$$

Then  $B = DA$  scales the rows of  $A$  by 4, 2, 1, and 0.5 respectively.

### 3. Dot Product Definition

This is the “compute each entry by hand” method:

$$b_{ik} = A(i, :) \cdot X(:, k).$$

**Meaning:** Each entry is the dot product of a row of  $A$  and a column of  $X$ .

### Example

Let

$$X = \begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 1 & 2 \\ 6 & 2 & 4 \end{bmatrix}.$$

Compute  $B = XA$ . For example:

$$b_{11} = [1 \ 0] \begin{bmatrix} 2 \\ 6 \end{bmatrix} = 2.$$

Repeating for all entries gives:

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

### 4. Outer Product Expansion

This method expresses the product as a sum of rank-one matrices:

$$AX = A(:, 1)X(1, :) + A(:, 2)X(2, :) + \cdots + A(:, p)X(p, :).$$

Each term is an **outer product**. This viewpoint is extremely useful for understanding rank and matrix factorizations.

### Example (Same $X$ and $A$ )

$$XA = X(:, 1)A(1, :) + X(:, 2)A(2, :)$$

which again yields

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

## Using Matrix Multiplication for Row/Column Operations

### Column operations (multiply on the right)

- **Scale** a column: use a diagonal matrix  $D_k(c)$ .
- **Swap** columns  $i$  and  $j$ : use a permutation matrix  $P_{ij}$ .
- **Add**  $c$  times column  $i$  to column  $j$ : use a shear matrix.

### Row operations (multiply on the left)

Same ideas as column operations, but applied to rows.

#### Example: Swap Columns 2 and 3

$$P_{23} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then  $AP_{23}$  swaps columns 2 and 3 of  $A$ .

## Dot Product Identities

Useful relationships:

$$\begin{aligned} x \cdot y &= y^T x, & y \cdot x &= x^T y, \\ (Ax) \cdot y &= x \cdot (A^T y), & x \cdot (Ay) &= (A^T x) \cdot y. \end{aligned}$$

These become important when studying orthogonality, projections, and symmetric matrices.

## Algebraic Properties of Matrix Multiplication

- **Associativity:**  $(AB)C = A(BC)$
- **Left distributive:**  $A(B \pm C) = AB \pm AC$
- **Right distributive:**  $(A \pm B)C = AC \pm BC$
- **Identity:**  $AI = A = IA$
- **Zero:**  $A0 = 0 = 0A$
- **Transpose rule:**  $(AB)^T = B^T A^T$
- **Scalar multiplication:**  $(\alpha A)B = A(\alpha B) = \alpha(AB)$

## Mini Practice Examples

### Add 2 times column 2 to column 3

Use the shear matrix

$$S_{23}(2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}.$$

Then  $AS_{23}(2)$  performs the operation.

### Swap rows 1 and 4

Left multiply:

$$P_{14} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

## Lesson 12 Notes: Nonsingular Linear Systems

### Square Linear Systems

We study systems of the form

$$Ax = b,$$

where  $A$  is an  $n \times n$  matrix,  $x$  is the unknown, and  $b$  is the output. This is basically the “reverse problem” of matrix-vector multiplication: instead of computing  $Ax$ , we ask what  $x$  produced  $b$ .

### Range of a Matrix

The vector  $b$  is in the **range of  $A$**  if it can be expressed as a linear combination of the columns of  $A$ . Formally:

$$\text{Range}(A) = \text{Span}\{A(:, 1), A(:, 2), \dots, A(:, n)\}.$$

If  $b$  is not in this span, the system has **no solution**.

### Diagonal Systems

For a diagonal matrix

$$A = \begin{bmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix},$$

the system  $Ax = b$  decouples into simple equations:

$$x_i = \frac{b_i}{d_{ii}}.$$

**Solution types:**

- **No solution:**  $d_{ii} = 0$  but  $b_i \neq 0$
- **Infinite solutions:**  $d_{ii} = 0$  and  $b_i = 0$
- **Unique solution:** all  $d_{ii} \neq 0$

**Example (Diagonal System)**

Solve

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \\ 12 \end{bmatrix}.$$

We get:

$$x_1 = 2, \quad x_2 = -2, \quad x_3 = 2.$$

**Upper-Triangular Systems (Backward Substitution)**

Let

$$U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & u_{nn} \end{bmatrix}.$$

Then

$$x_n = \frac{y_n}{u_{nn}}, \quad x_i = \frac{1}{u_{ii}} \left( y_i - \sum_{j=i+1}^n u_{ij} x_j \right).$$

**Example (Backward Substitution)**

Solve:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix} x = \begin{bmatrix} 10 \\ 20 \\ 18 \end{bmatrix}.$$

We compute:

$$x_3 = 3, \quad x_2 = 1.25, \quad x_1 = -1.5.$$

**Lower-Triangular Systems (Forward Substitution)**

Solve from the top down:

$$x_1 = \frac{b_1}{\ell_{11}}, \quad x_i = \frac{1}{\ell_{ii}} \left( b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right).$$

### Example (Forward Substitution)

$$\begin{bmatrix} 2 & 0 & 0 \\ 1 & 3 & 0 \\ 4 & -2 & 5 \end{bmatrix} x = \begin{bmatrix} 6 \\ 9 \\ 3 \end{bmatrix}.$$

We find:

$$x_1 = 3, \quad x_2 = 2, \quad x_3 = -1.$$

### Gaussian Elimination

We use elementary matrices to convert any system  $Ax = b$  into an upper-triangular system  $Ux = y$ .

Steps:

1. Pick pivot
2. Use shear matrices to eliminate entries below pivot
3. Move right and down
4. Continue until upper-triangular form
5. Solve  $Ux = y$  with backward substitution

### Regular Matrices

A matrix is **regular** if it can be reduced to an upper-triangular matrix with all nonzero pivots using only **shear matrices**. No row swaps required.

## Lesson 13 Notes: Matrix Inverses

### Definition of an Invertible Matrix

A square matrix  $A$  is **invertible** (or **nonsingular**) if there exists a matrix  $A^{-1}$  such that

$$AA^{-1} = A^{-1}A = I.$$

If such a matrix does not exist,  $A$  is **singular**.

### Elementary Matrices and Their Inverses

All elementary matrices are invertible:

$$S_{ik}(c)^{-1} = S_{ik}(-c), \quad P_{ik}^{-1} = P_{ik}^T, \quad D_i(c)^{-1} = D_i(1/c).$$

## Cramer's Rule for a $2 \times 2$ Inverse

For

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

the inverse is

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

### Example

Let

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

Then:

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

## Properties of Inverses

For invertible matrices  $A, B$ :

- $A^{-1}$  is unique
- The system  $Ax = b$  has a unique solution
- $(AB)^{-1} = B^{-1}A^{-1}$
- $(A^T)^{-1} = (A^{-1})^T$
- Any invertible matrix can be expressed as a product of elementary matrices

## RREF Solves the Linear System

If  $U = \text{RREF}(A)$  and  $Ux = y$  is equivalent to  $Ax = b$ , then both systems share the **exact same solution set**. This is why RREF is so important.

## Lesson 14: The Invertible Matrix Theorem (IMT)

The **Invertible Matrix Theorem** is basically a huge “all-or-nothing” checklist. If *one* statement is true for an  $n \times n$  matrix  $A$ , then *all* of them are true. If one fails, they all fail. Super convenient.

### Core Idea

A matrix  $A$  is **invertible** (aka **nonsingular**) if there exists  $A^{-1}$  such that

$$AA^{-1} = I_n \quad \text{and} \quad A^{-1}A = I_n.$$

## IMT Part 1 — Algebra + Transformations

All of the following are equivalent:

- There exists a matrix  $C$  with  $CA = I$ .
- There exists a matrix  $D$  with  $AD = I$ .
- $A$  is **invertible**.
- $A$  is row-equivalent to an upper triangular matrix with all diagonal entries  $\neq 0$ .
- $A$  has  $n$  **pivot positions**.
- The equation  $Ax = 0$  has only the **trivial solution**.
- The columns of  $A$  are **linearly independent**.
- The linear transformation  $T(x) = Ax$  is **one-to-one**.
- The equation  $Ax = b$  has a **unique solution** for all  $b$ .
- The columns of  $A$  **span**  $\mathbb{R}^n$ .
- The linear transformation  $T(x) = Ax$  is **onto**  $\mathbb{R}^n$ .
- $A^T$  is invertible.

### Example: Checking Invertibility

Let

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

Row-reduction gives

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

Only one pivot  $\Rightarrow$  not invertible.

By IMT, this means:

- columns are not independent
- $\det(A) = 0$
- $Ax = b$  does not have a unique solution for all  $b$

## IMT Part 2 — Rank, Null Space, Determinant

Also equivalent to invertibility:

- $\det(A) \neq 0$ .
- Columns of  $A$  form a basis for  $\mathbb{R}^n$ .
- $\text{Col}(A) = \mathbb{R}^n$ .
- $\dim(\text{Col}(A)) = n$ .
- $\text{rank}(A) = n$ .
- $\text{Null}(A) = \{0\}$ .
- $\dim(\text{Null}(A)) = 0$ .
- $(\text{Col}(A))^\perp = \{0\}$ .
- $(\text{Null}(A))^\perp = \mathbb{R}^n$ .
- $\text{Row}(A) = \mathbb{R}^n$ .

### Example: Determinant Check

Let

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

Compute  $\det(A)$  via cofactor expansion:

$$\det(A) = 3(-9) + 1(-5) + 2(-4) = -40 \neq 0.$$

Thus  $A$  is invertible.

## IMT Part 3 — Eigenvalues & SVD

Equivalent to invertibility:

- 0 is **not** an eigenvalue of  $A$ .
- $A$  has  $n$  **nonzero singular values**.

## Lesson 15: LU Factorization

### What is LU Factorization?

We write a matrix as

$$A = LU,$$

where:

- $L$  is **unit lower triangular** (1's on diagonal)
- $U$  is **upper triangular**

Why do this? It makes solving systems  $Ax = b$  way faster.

## Why LU Helps

Instead of solving  $Ax = b$  directly, we do:

$$A = LU \quad \Rightarrow \quad LUx = b.$$

Let  $Ux = y$ . Then:

$$Ly = b \quad (\text{forward substitution})$$

$$Ux = y \quad (\text{backward substitution})$$

Once  $A$  is factored, we can reuse  $L$  and  $U$  for any new right-hand-side vector  $b$ .

## Conditions for LU Without Pivoting

To compute LU directly (no row swaps),  $A$  must:

- be square,
- have nonzero diagonal entries during elimination,
- have nonzero leading principal minors.

## Example: LU Factorization (3x3)

Let

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

### Eliminate Column 1

Multipliers:

$$m_{21} = 2, \quad m_{31} = 1.$$

After elimination:

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

## Eliminate Column 2

$$m_{32} = 2.$$

Final:

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

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

## Solving $Ax = b$ Using LU

Let

$$b = \begin{bmatrix} 4 \\ 9 \\ 6 \end{bmatrix}.$$

### Step 1: Forward Substitution ( $Ly = b$ )

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 9 \\ 6 \end{bmatrix}.$$

$$y_1 = 4, \quad y_2 = 1, \quad y_3 = 0.$$

### Step 2: Backward Substitution ( $Ux = y$ )

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

$$x_3 = 0, \quad x_2 = 1, \quad x_1 = 1.$$

Thus

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

## Gauss Transformation Matrices

A Gauss matrix has the form

$$L_k = I - \ell_k e_k^T,$$

where  $\ell_k$  stores multipliers used to eliminate entries below the pivot.

Inverse:

$$L_k^{-1} = I + \ell_k e_k^T.$$

These combine to build the full  $L$  factor.

## Lesson 16: Determinants

### Why Determinants Matter

$$Ax = b,$$

where  $A$  is an  $n \times n$  matrix. For square matrices, we know the possibilities for solutions:

- **No solution**
- **Exactly one solution**
- **Infinitely many solutions**

By the **Invertible Matrix Theorem**, if  $A$  is invertible, then  $Ax = b$  always has exactly one solution. So if we can quickly check whether  $A$  is invertible, we immediately know the behavior of the system.

Reducing  $A$  to RREF works, but it's expensive for large matrices. Enter the **determinant**: a function that tells us instantly whether  $A$  is invertible.

$$\det(A) \neq 0 \iff A \text{ is invertible}, \quad \det(A) = 0 \iff A \text{ is singular}.$$

### Defining the Determinant

A function

$$\det : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$$

is a **determinant function** if it satisfies all of the following:

1.  $\det(I_n) = 1$
2. If  $A$  has an all-zero row, then  $\det(A) = 0$
3.  $\det(S_{ik}(c)A) = \det(A)$  (shear: adding  $c$  times row  $k$  to row  $i$ )
4.  $\det(P_{ik}A) = -\det(A)$  (row swap)
5.  $\det(D_i(c)A) = c \cdot \det(A)$  (scale a single row)

There is exactly *one* function satisfying these rules: the determinant.

## Permutation Formula (Theoretical Definition)

**Theorem (Permutation Definition).** For  $A \in \mathbb{R}^{n \times n}$ ,

$$\det(A) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) a_{1,\sigma(1)} a_{2,\sigma(2)} \cdots a_{n,\sigma(n)}.$$

This is the “official” definition, although it’s extremely tedious to compute by hand for big  $n$ .

### Example: $2 \times 2$ Determinant

Let

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

The two permutations of  $\{1, 2\}$  give:

$$\det(A) = a_{11}a_{22} - a_{12}a_{21}.$$

### Example: $3 \times 3$ Determinant

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

Using the six permutations of  $S_3$ :

$$\det(A) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}.$$

### Shortcut: $3 \times 3$ Diagonal Diagram

A practical way to compute a  $3 \times 3$  determinant is the diagonal method:

$$\begin{array}{ccccccc} + & + & + & - & - & - \\ a_{11} & a_{12} & a_{13} & a_{11} & a_{12} & & \\ a_{21} & a_{22} & a_{23} & a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & a_{31} & a_{32} & & \end{array}$$

Multiply along the three “blue” diagonals and add; multiply along the three “red” diagonals and subtract.

## Properties of Determinants

**Theorem.** For  $A, B \in \mathbb{R}^{n \times n}$  and scalar  $c \in \mathbb{R}$ :

1. If  $A$  is triangular, then  $\det(A) = a_{11}a_{22} \cdots a_{nn}$ .
2.  $\det(A) = \det(A^T)$ .

3.  $\det(AB) = \det(A)\det(B)$ .
4. If  $S$  is invertible, then  $\det(SAS^{-1}) = \det(A)$ .
5. Row swap:  $\det(P_{ik}A) = -\det(A)$ .
6. Shear:  $\det(S_{ik}(c)A) = \det(A)$ .
7. Row-scale:  $\det(D_i(c)A) = c \det(A)$ .
8. Scalar multiple of matrix:  $\det(cA) = c^n \det(A)$ .
9.  $A$  is invertible  $\iff \det(A) \neq 0$ .

### Example: Determinant of a Triangular Matrix

If

$$U = \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix},$$

then

$$\det(U) = u_{11}u_{22}.$$

Same idea extends to  $3 \times 3$  and beyond.

### Using Determinants to Check Linear Independence

Given vectors  $v_1, \dots, v_n$ , form a matrix  $A = [v_1 \ \cdots \ v_n]$ . Then:

$$\det(A) \neq 0 \iff \{v_1, \dots, v_n\} \text{ are linearly independent.}$$

### Other Useful Facts

- If  $A$  is invertible, then

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

- For any scalar  $c$ ,

$$\det(cA) = c^n \det(A).$$

### Challenge Example: Vandermonde Determinant

For three data points  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ , the associated Vandermonde matrix is

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix}.$$

Using row operations, one can show:

$$\det(A) = (x_2 - x_1)(x_3 - x_1)(x_3 - x_2),$$

which is nonzero exactly when all  $x_i$  are distinct.

## Lesson 17: Row Echelon Form (REF) & Reduced Row Echelon Form (RREF)

### REF (Row Echelon Form)

A matrix is in **REF** if:

- all the zero rows sit at the *bottom* (like they slid down there),
- each row's first nonzero entry (the **pivot**) is to the *right* of the pivot above it,
- everything under a pivot is zero.

Basically it looks like a staircase that someone drew kinda crooked.

### RREF (Reduced Row Echelon Form)

Same as REF but with stricter rules:

- pivots are **1**,
- each pivot is the **only** nonzero entry in its column.

So pivot columns look super clean.

### Elementary Matrices (our legal moves)

- **Shear**  $S_{ki}(c)$ :  $\text{row}_k \leftarrow \text{row}_k + c \cdot \text{row}_i$
- **Swap**  $P_{ik}$ : switches row  $i$  and row  $k$
- **Dilation**  $D_i(c)$ : multiplies row  $i$  by  $c$  (nonzero)

Gaussian elimination is literally just multiplying by a bunch of these.

### General algorithm to get RREF

1. Find the first pivot (top-left-ish). Swap if it's zero.
2. Zero everything under the pivot.
3. Move diagonally down-right to hunt for the next pivot.
4. Turn each pivot into 1 (dilation).
5. Clear everything above pivots so pivot-columns are nice.

## Example Problem (quick + scribbly)

Turn

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

into RREF.

(not writing all the intermediate matrices — this is just the vibe of my notes)

- Row2 – 2·Row1 → all zeros
- Row3 – 3·Row1 → something like [0, 18, 0, -1]
- Scale pivot to 1, clean above

Eventually you get something like

$$\begin{bmatrix} 1 & -3 & 2 & -1 \\ 0 & 1 & 0 & -\frac{1}{18} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Pivot cols = 1,2. Free cols = 3,4.  $\Rightarrow$  2 free vars  $\Rightarrow$  2 independent homogeneous solutions.

## Lesson 18: Solution Sets for General Linear Systems

**Homogeneous system**  $Ax = 0$

Always has at least the zero solution. But we want the nonzero ones.

Key fact (the big one):

$$Ax = 0 \iff \text{RREF}(A)x = 0.$$

So we always solve the homogeneous system using the RREF version.

**How to find solutions to**  $Ax = 0$

1. Compute  $U = \text{RREF}(A)$ .
2. Identify pivot vs nonpivot columns.
3. Each **nonpivot column** corresponds to one **free variable**.
4. Set a free variable = 1 (others = 0), solve for pivot variables.
5. This gives one vector  $z_i$ .
6. Do this for all free vars  $\rightarrow$  these  $z_i$  form the basis of the nullspace.

Number of linearly independent solutions = **number of nonpivot columns**.

### Example (from notes)

Given

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

Pivot columns: 1 and 3 Nonpivot columns: 2, 4, 5  $\Rightarrow$  3 independent solutions.

(Just writing the solutions directly)

$$z_1 = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad z_2 = \begin{bmatrix} 1 \\ 0 \\ -2 \\ 1 \\ 0 \end{bmatrix}, \quad z_3 = \begin{bmatrix} -3 \\ 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}.$$

So the whole homogeneous solution set is

$$x = c_1 z_1 + c_2 z_2 + c_3 z_3.$$

### General (non-homogeneous) system $Ax = b$

A solution exists iff

$$b \in \text{Span}(A(:, 1), \dots, A(:, n)).$$

If at least one solution exists, then the full solution set is always

$$x = x_p + z,$$

where

- $x_p$  is any particular solution,
- $z$  is any solution of  $Ax = 0$ .

This is the “particular + homogeneous” formula.

### Example sketch

Suppose RREF gives

$$Ux = y = \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix}.$$

Solve for pivot variables  $\rightarrow$  get some  $x_p$ . Then add the homogeneous solutions we found earlier.

So the final solution set looks like:

$$x = x_p + c_1 z_1 + c_2 z_2 + c_3 z_3.$$