



*Linear Algebra*

# ***Matrices Part 2: Matrix Expansion Concept***

Automotive Intelligence Lab.



HANYANG UNIVERSITY



# Contents

## ■ Matrix multiplication

## ■ Determinant

## ■ Matrix spaces (column, row, nulls)

## ■ Inverse matrix, column space, and null space

## ■ Rank

## ■ Rank applications

## ■ Matrix norms

## ■ Summary

## ■ Code exercises

Linear Transformation

$$Ax = b$$

$$x \rightarrow \boxed{A} \xrightarrow{} b$$

# Matrix multiplication



HANYANG UNIVERSITY



# Composition of the Two Separate Transformations (1)

## ■ Matrix multiplication

- ▶ Applying one **transformation** and then another.
- ▶ One of linear transformation

## ■ This transformation can be described with a matrix its own by **i-hat** and **j-hat** like other linear transformations.

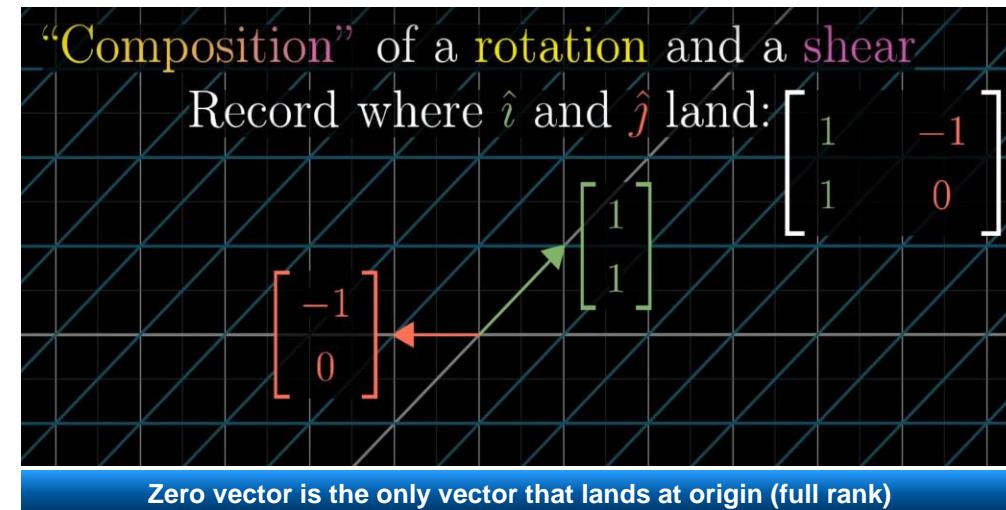
## ■ Example of applying **rotation** and then **shear** matrix →

- ▶  $\hat{i}$  ends up at  $(1,1)$ .
- ▶  $\hat{j}$  ends up at  $(-1,0)$ .

## ■ So, applying rotation and then shear is a **one single action**, rather than two successive ones.

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

Shear                      Rotation



# Composition of the Two Separate Transformations (2)

## ■ Numerical example of adjusting rotation and shear.

- ▶ Apply rotation then shear matrix as shown below.

1. Multiply the matrix  $\begin{bmatrix} x \\ y \end{bmatrix}$  with the rotation matrix on the left.
2. Multiply the matrix calculated at previous stage by the shear matrix on the left.
3. Then, its result will be same as just applying new composition matrix.

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

Shear      Rotation      Composition  
Rotation

## ■ The new matrix is called “Product of the matrix”.

- ▶ Apply transformation by the matrix on the right, then apply the transformation by the matrix on the left.

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

Right to left  
Product of new matrix

# Another Example of Matrix Product

■ Example of multiplying matrix  $M1 = \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}$ , and  $M2 = \begin{bmatrix} 1 & -2 \\ 1 & 0 \end{bmatrix}$ .

- ▶ Multiplication of  $M1$  and  $M2$

- ▶ How to get  $\hat{i}$ ?



- ▶ How to get  $\hat{j}$ ?



- ▶ Result of matrix product

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

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

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

# Generalization of Matrix Product

■ Write the example in general case, where matrix  $M_1$  is  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ , and  $M_2$  is  $\begin{bmatrix} e & f \\ g & h \end{bmatrix}$ .

- ▶ Multiplication of  $M_1$  and  $M_2$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

- ▶ How to get  $\hat{i}$ ?

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e \\ g \end{bmatrix} = e \begin{bmatrix} a \\ c \end{bmatrix} + g \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} ae + bg \\ ce + dg \end{bmatrix}$$

- ▶ How to get  $\hat{j}$ ?

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} f \\ h \end{bmatrix} = f \begin{bmatrix} a \\ c \end{bmatrix} + h \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} af + bh \\ cf + dh \end{bmatrix}$$

- ▶ Result of matrix product

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

# Proof of $M_1M_2 \neq M_2M_1$

## ■ Proof of $M_1M_2 \neq M_2M_1$ by an example of shear and rotation.

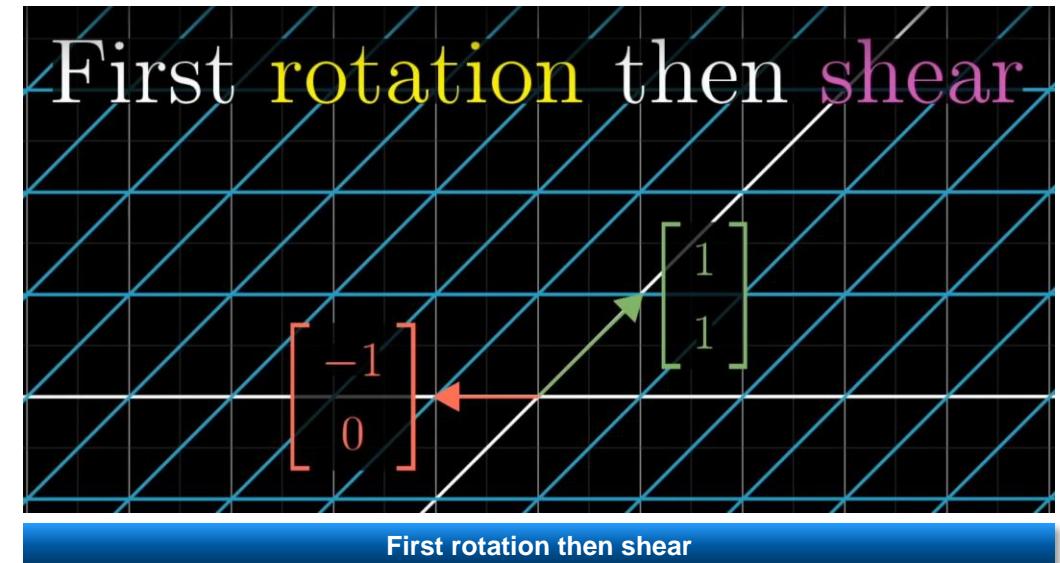
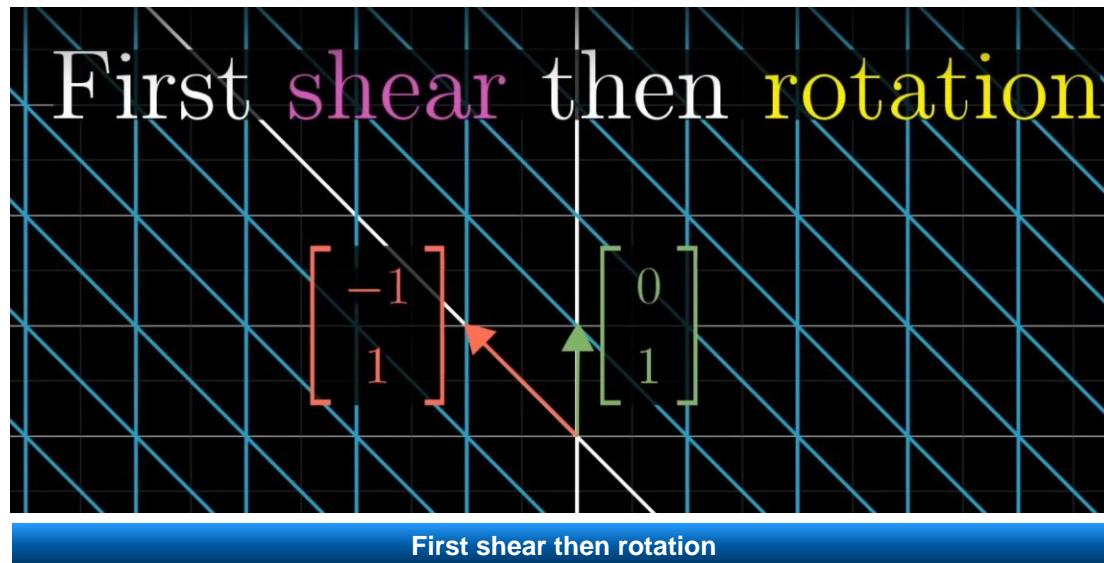
### ► First shear then rotation.

- I-hat end up at (0,1).
- J-hat ends up at (-1,1).

### ► First rotation then shear.

- I-hat end up at (1,1).
- J-hat ends up at (-1,0).

■ So, the order of matrix totally matters.

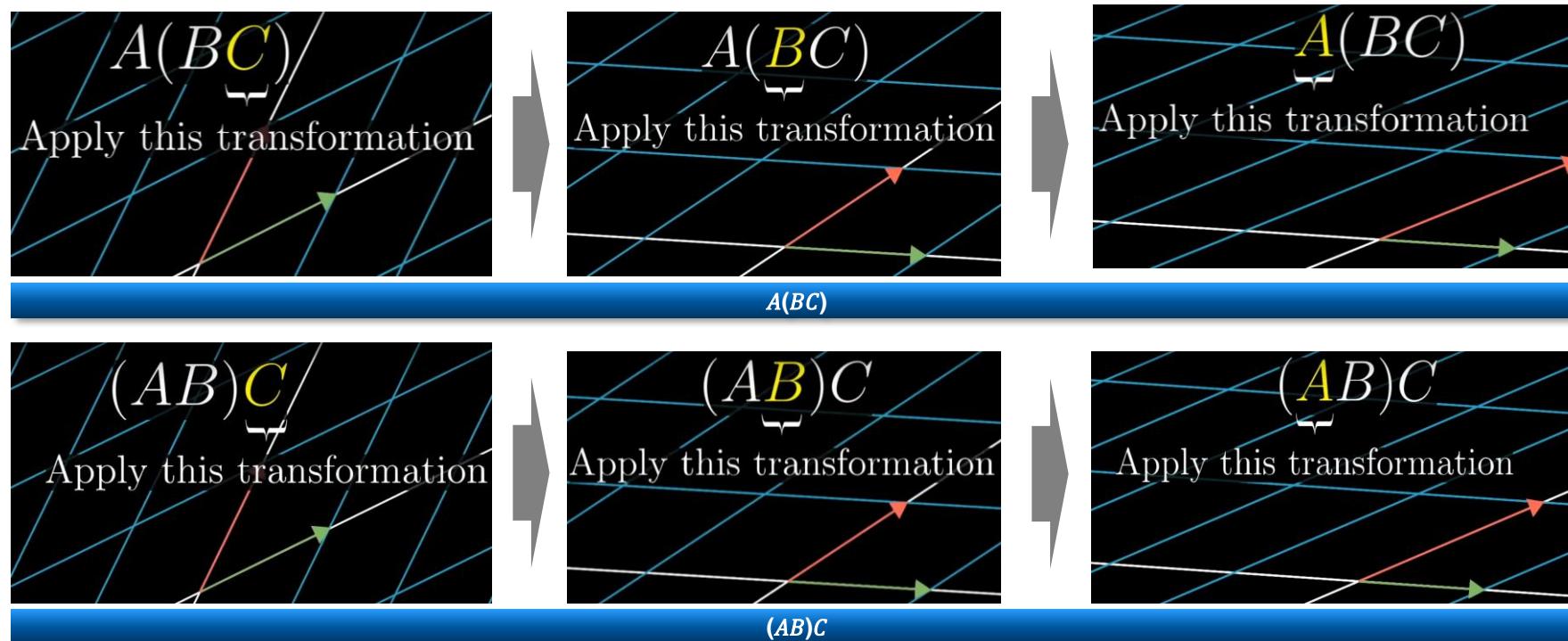


# Proof of Matrix Associativity

## ■ Meaning of matrix associativity

- ▶ Assume that we have three matrices  $A, B$  and  $C$ .
- ▶ If multiplying them all together, it shouldn't matter if first computing  $A$  times  $B$ , then multiply the result by  $C$ , or first multiplying  $B$  times  $C$ , then multiply that result by  $A$  on the left.
- ▶  $(AB)C = A(BC)$

## ■ Same as matrix multiplication as applying one transformation after another, so $(AB)C = A(BC)$



# Determinant I



HANYANG UNIVERSITY



# Definition of Determinant

## ■ A number associated with a square matrix

- ▶ In abstract linear algebra, the determinant is a keystone quantity in several operations.
- ▶ In practice, it can be numerically unstable for large matrices.
  - Due to underflow and overflow issues.

## ■ The two most important properties

1. Defined only for Square matrices.
2. Zero for singular (dependent vector set, reduced-rank) matrices

## ■ Notated as $\det(A)$ or $|A|$ — determinant

- ▶  $\Delta$  is used when it doesn't refer to a specific matrix.

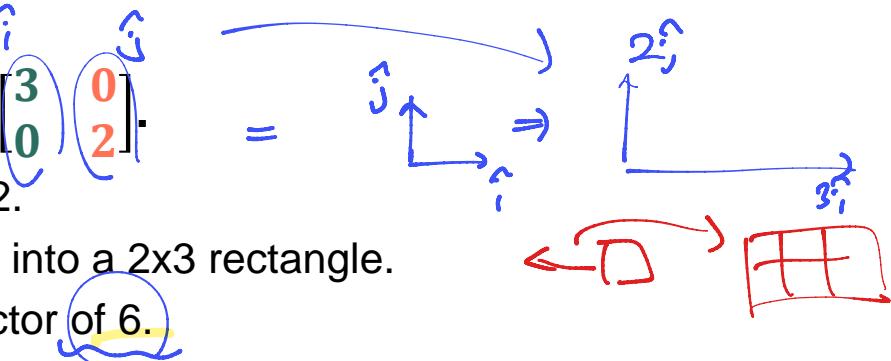
Cigen value  
자연치수

$$\begin{array}{c} \textcircled{N} \rightarrow [n \times n] \rightarrow \textcircled{N} \\ \left[ \begin{array}{cc} 1 & 2 \\ 2 & 4 \end{array} \right] \rightarrow 10 \end{array}$$

# Examples of Scaling through Transformation

## ■ Measure the amount of scaling by the example of $\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$ .

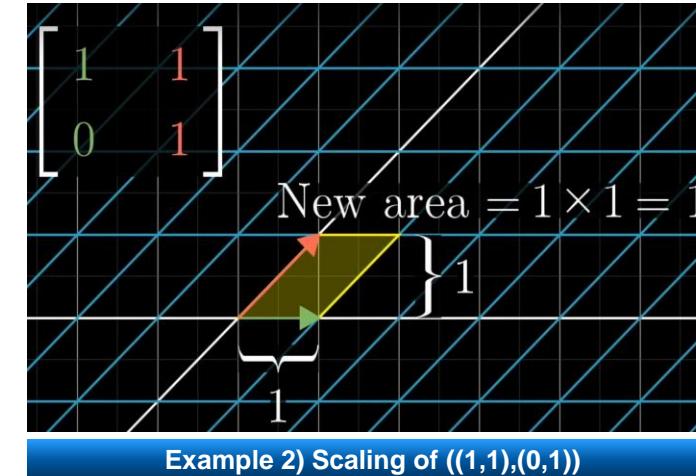
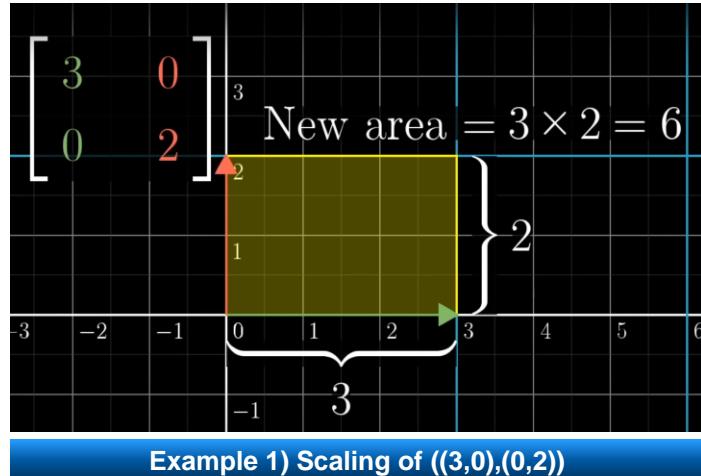
- ▶  $\hat{I}$  scales by the factor of 3, and  $\hat{J}$  scales by the factor of 2.
- ▶ The matrix started out with  $1 \times 1$  rectangle and then turns into a  $2 \times 3$  rectangle.
- ▶ So, the linear transformation has scaled its area by a factor of 6.



## ■ Measure the amount of scaling by the example of $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ .

- ▶  $\hat{I}$  scales remains the same, and  $\hat{J}$  moves to  $(1,1)$ .
- ▶ The  $1 \times 1$  rectangle gets slanted and turned into a parallelogram which still has the area of 1.

■ If you know how much the unit square changes, you can know how the area of the possible region in space changes.



# Determinant of a Transformation

변환의 영역을 계산하는 것, 즉 **적분**.

변환의 스케일

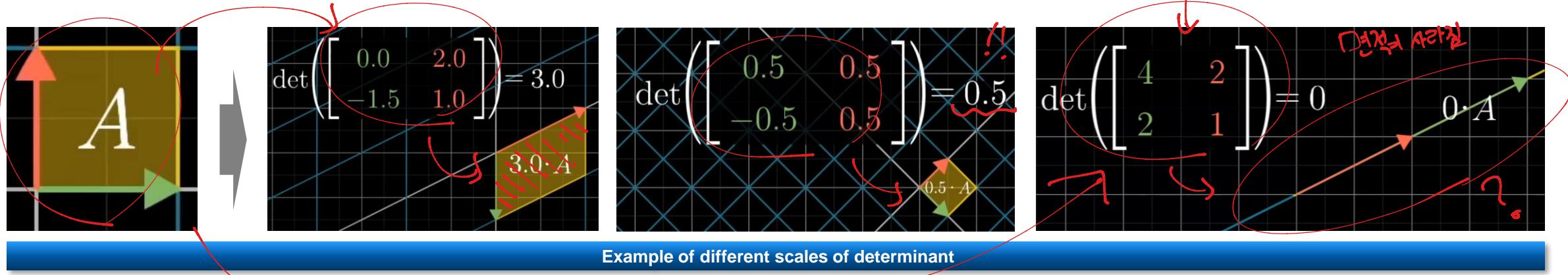
Determinant

## Meaning of determinant

- ▶ **Scaling** factor by which a linear transformation changes any area.

## Examples

- ▶ If transformation **increases** the area of a region by a factor of 3, the determinant of a transformation will be 3.
- ▶ If transformation **squishes** down all areas by a factor of  $\frac{1}{2}$ , the determinant of a transformation will be  $\frac{1}{2}$ .
- ▶ If transformation **squishes all of space onto a line**, or a **single point**, the determinant of a 2D transformation is 0.



# Negative Value of Determinant

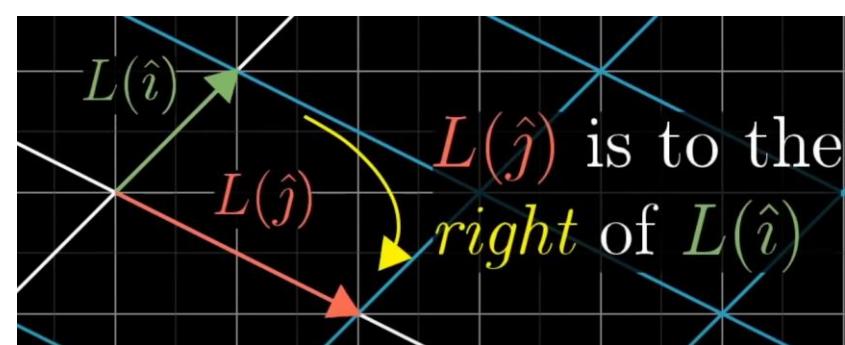
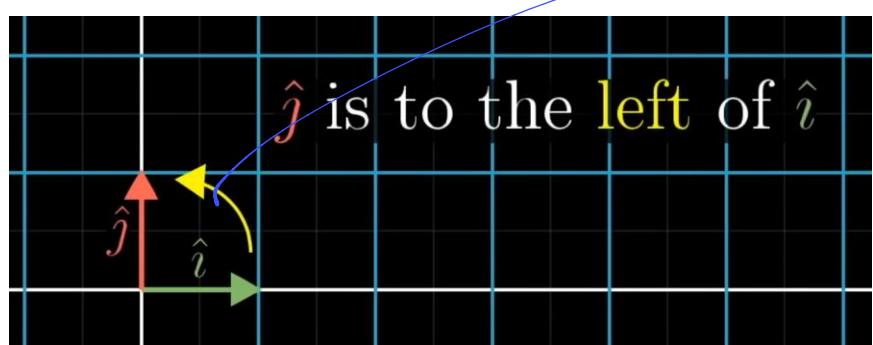
■ Scaling area by a negative number means “*inverting orientation*”

■ In the aspect of  $\hat{i}$  and  $\hat{j}$ ,

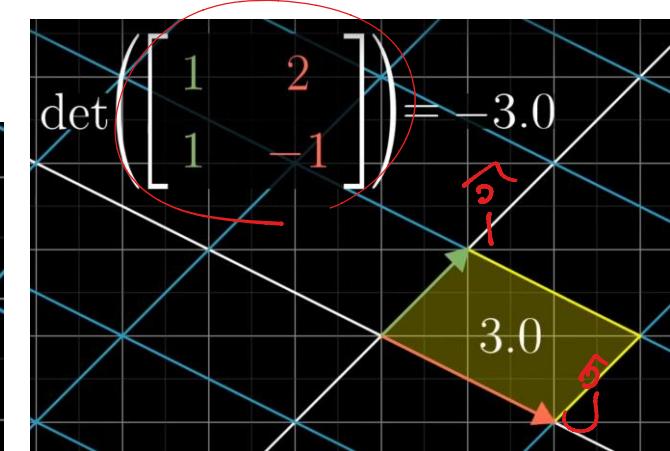
- ▶  $\hat{j}$  is to the left of  $\hat{i}$ , and after a transformation,  $j\text{-hat}$  is now on the right of  $i\text{-hat}$ .
- ▶ So, the orientation of space has been inverted.

■ Example

- ▶ If transformation **scales** the area of a region by a factor of 3 and its **space gets flipped over**, the determinant of a transformation will be -3.



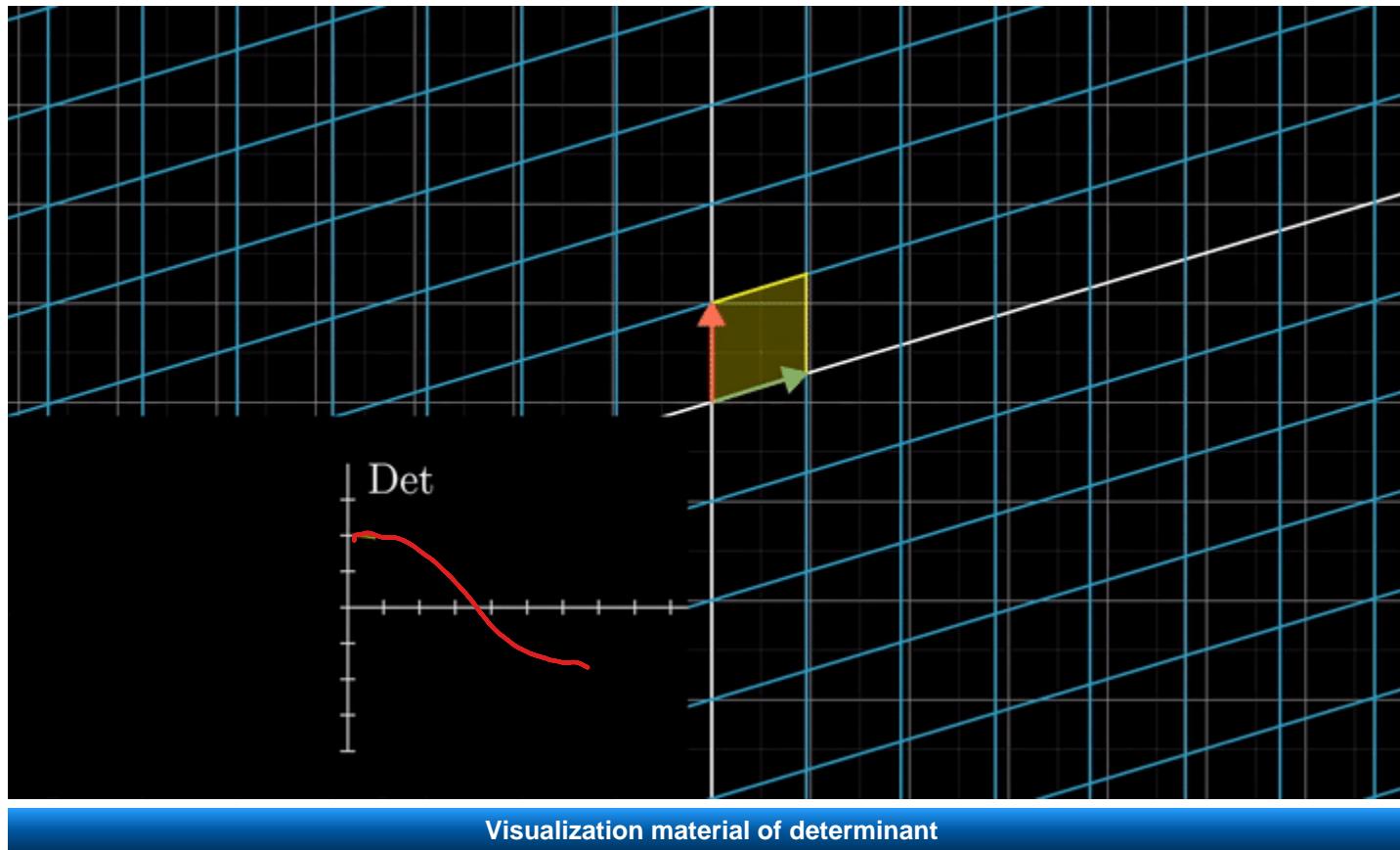
Visualization of negative value determinant



Example of determinant -3

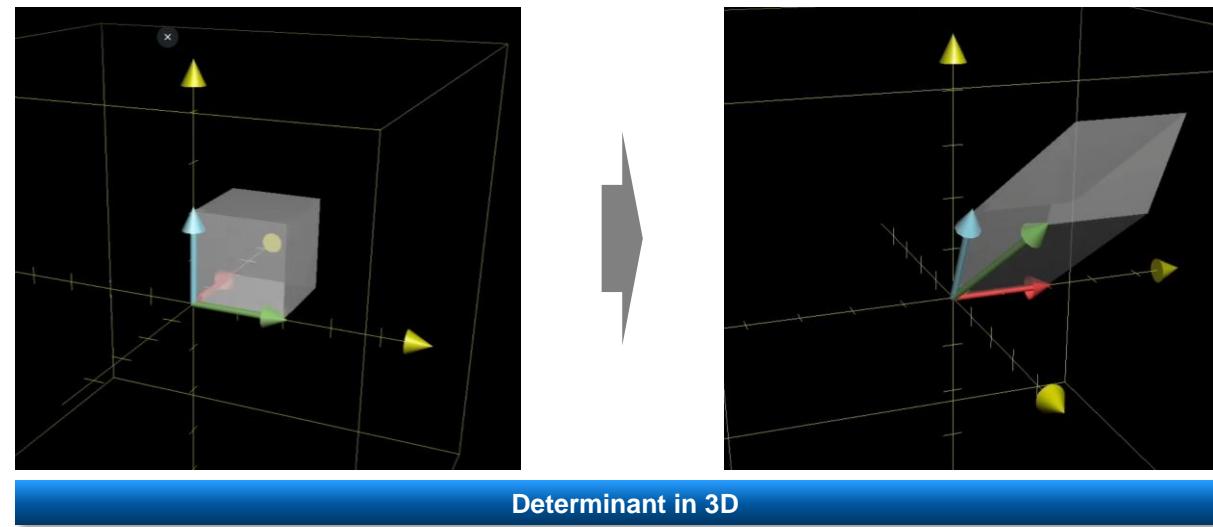
# Visualization of Determinant

## ■ Why does negative area relate to orientation-flipping?



# Intuition Understanding of Determinant

- Determinant in 3D tells how much a transformation scales things, and how much volumes get scaled.
- In the aspect of  $\hat{I}$  and  $\hat{J}$ ,  $\hat{K}$ 
  - ▶ 3D determinant can be described with **cube** whose edges are resting on the **basis vectors**,  $\hat{I}, \hat{J}$ , and  $\hat{K}$ .
  - ▶ After transformation, cube might get warped into slanty ways, named “**parallelepiped**”.
- So, determinant determines the **volume of parallelepiped that the cube turns into**.

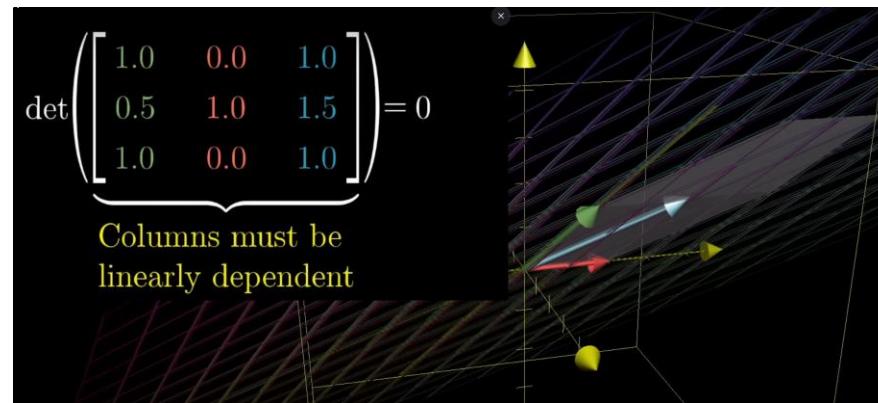


Determinant in 3D

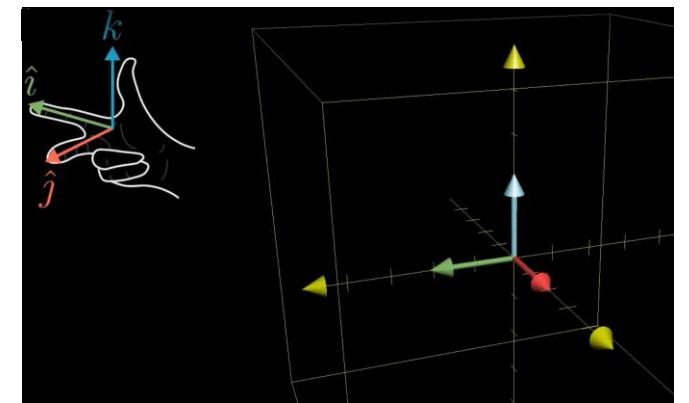
# Examples of Determinant in 3D

## ■ Examples

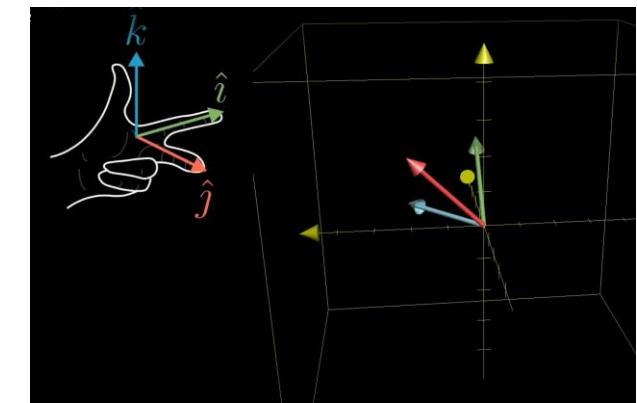
- ▶ Determinant of 0 would mean that all of space is **squished onto something with 0 volume**, meaning either a flat plane, a line, or, in the most extreme case, onto a single point.
  - This means that the columns of the matrix are “**linearly dependent**”.
- ▶ Determinant of negative would mean that “**flip of orientation**”.



Determinant of 0 means linearly dependent



Visualization of flipped orientation (when determinant is negative)

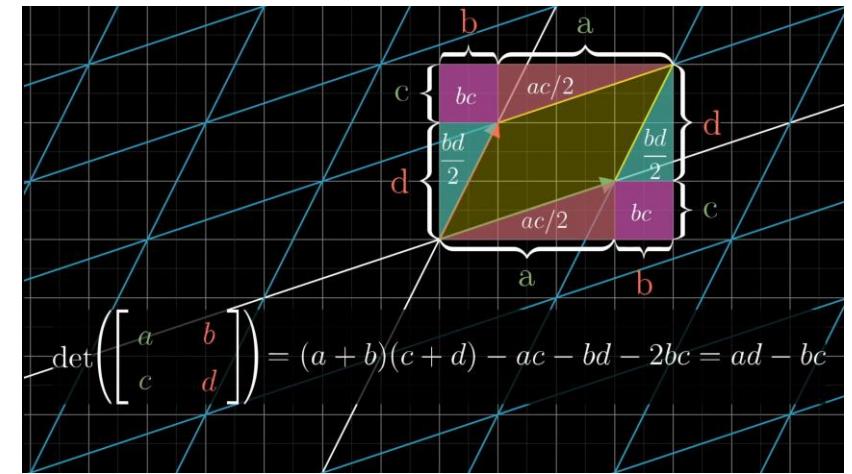
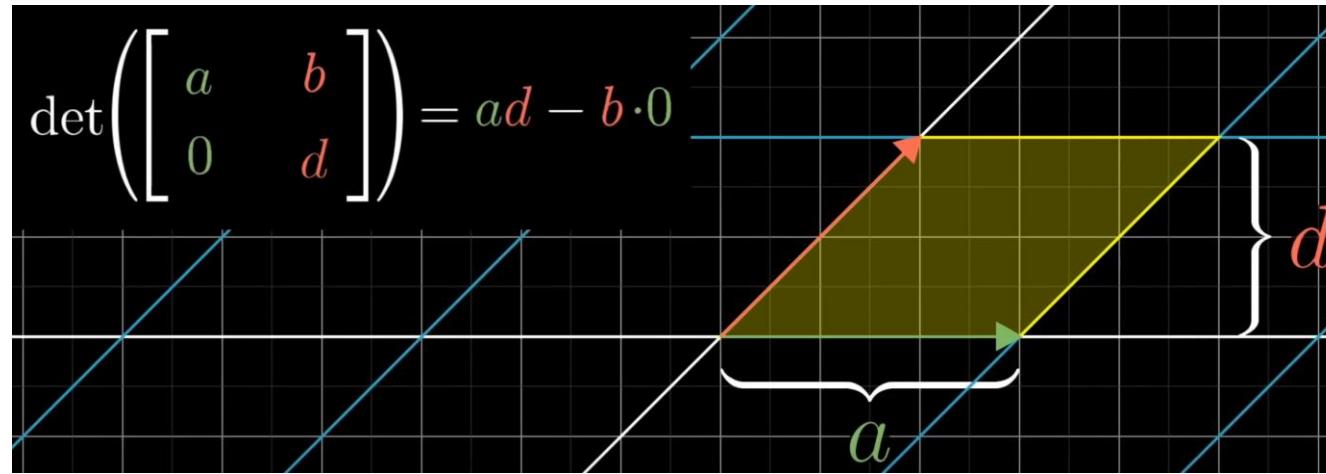


# Computing of Determinant in 2D

■  $\det\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$

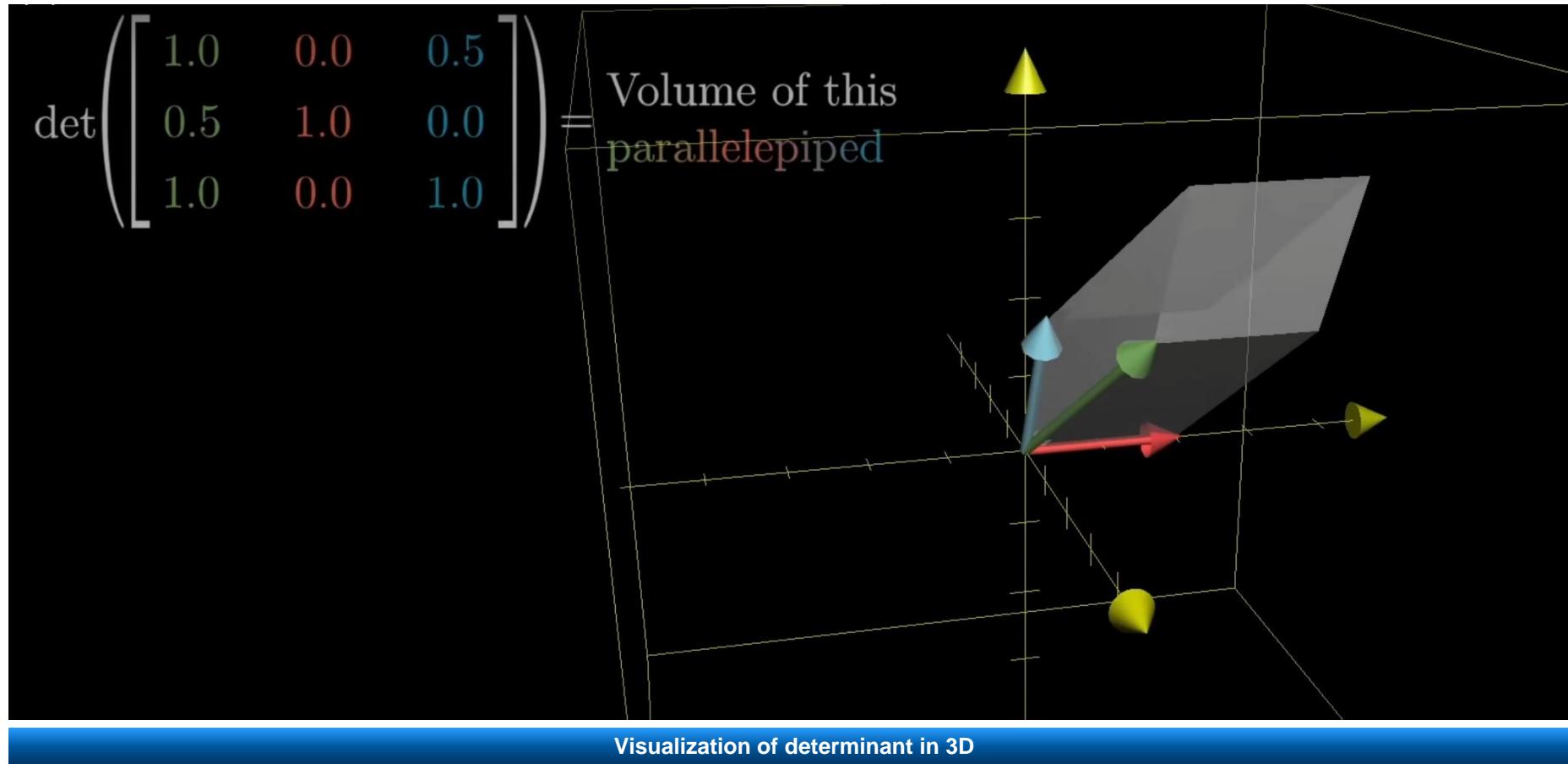
■ Proof of the formula

1. Assume that  $b$  and  $c$  are both 0. Then,  $a$  will represent how much  $\hat{I}$  is stretched in the **x direction**, and  $d$  represent how much  $\hat{J}$  is stretched in **y direction**. So, the output is the rectangle of  $a*d$ .
2. Assume that only one of  $b$  or  $c$  are 0. Then the output will be a parallelogram with a base of  $a$  and the height of  $d$ , which still has the area of  $ad$ .
3. Assume that both  $b$  and  $c$  are not 0. Then  $bc$  will affect how much this parallelogram is stretched or squished in the diagonal direction.



# Computing of Determinant in 3D

■  $\det\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = a \det\begin{bmatrix} e & f \\ h & i \end{bmatrix} - b \det\begin{bmatrix} d & f \\ g & i \end{bmatrix} + c \det\begin{bmatrix} d & e \\ g & h \end{bmatrix}$



# Determinant II

$\text{Det} = 0 \text{ iff } \exists \mathbf{z}$ .  
↳ Linear dependence



# Applications of Determinant

## ■ In geometric interpretation

- ▶ During matrix-vector multiplication, it is related to how much the matrix stretches vectors.
- ▶ During the transformation, a negative determinant means that one axis is rotated.

## ■ In data science-related applications

- ▶ Used algebraically.
- ▶ Crucial step in advanced topics.
  - Matrix inverse, eigen decomposition and singular value decomposition.

# Computing the Determinant

- Computing the determinant is time-consuming and tedious.
- Shortcut for computing the determinant of a  $2 \times 2$  matrix, which is shown in Eq 1..
  - ▶ For a real-valued matrix, the determinant will always be a real number.

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Eq 1. Computing the determinant of a  $2 \times 2$  matrix

## ■ Code Exercise of Determinant of matrix

- ▶ [Code Exercise \(06\\_01\)](#)

```
% Clear workspace, command window, and close all figures
clc; clear; close all;

% Define a 2x2 matrix
A = [3, 2; 0, 2];
B = [1, 3, 2; 3, 1, 5; 2, 4, 1];

% Calculate the determinant of matrix A,B
determinantA = det(A);
determinantB = det(B);

% Display the determinant
disp('The determinant of matrix A is:');
disp(determinantA);
disp('The determinant of matrix B is:');
disp(determinantB);
```

$\det(A)$

MATLAB code example of Determinant of matrix

# Problem of Computing the Determinant

## ■ The shortcut method for $2 \times 2$ matrix **doesn't scale up** to larger matrices.

- ▶ Eq 1. is a “shortcut” for  $3 \times 3$  matrices, but it isn’t really a shortcut.
- ▶ Eq 2. is a “shortcut” for  $4 \times 4$  matrices, also not shortcut.

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - ceg - bdi - afh$$

Eq 1. Computing the determinant of a  $3 \times 3$  matrix

$$\begin{vmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{vmatrix} = afkp - aflo - agjp + \dots - dgim + dgjm$$

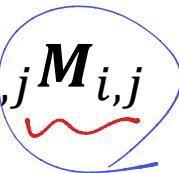
Eq 2. Computing the determinant of a  $4 \times 4$  matrix

# Generalization of Determinant

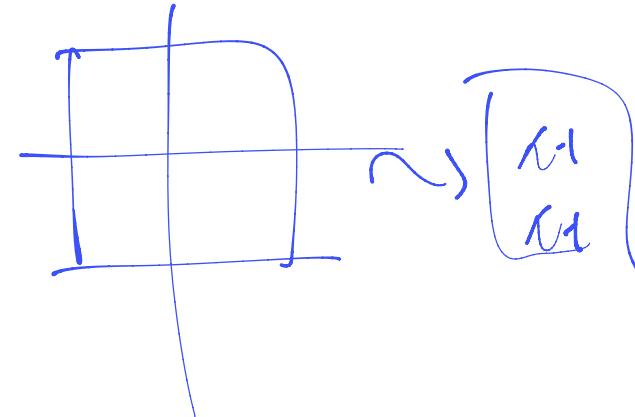
■ Let  $A$  be an  $N \times N$  matrix.

►  $M$  is the determinant of the Submatrix.

- Submatrix: constructed by removing the  $i$ th row and  $j$ th column of  $A$ .

$$\det(A) = \sum_{j=1}^N (-1)^{i+j} A_{i,j} M_{i,j}$$


Generalization of determinant



# Determinant with Linear Dependencies

- Determinants are **Zero** for any singular matrix.
- Any singular matrix has at least one column.
  - ▶ Expressed as a linear combination of other columns.
- All nonsingular (full-rank) matrices have a **Nonzero** determinant.
- In geometric meaning:  $\Delta = 0$ .
  - ▶ A matrix with  $\Delta = 0$  is a transformation, at least one dimension gets flattened to have surface area but no volume.

$$\begin{vmatrix} a & \lambda a \\ c & \lambda c \end{vmatrix} = ac\lambda - a\lambda c = 0$$

Eq 1. Reduced-rank matrix has a determinant of 0

# Using the Determinant to Find a Missing Matrix Element

- $a, b, c, \lambda$  are the elements in the matrix and  $\Delta$  is the determinant value.
- Assume that  $a, b, c$  and  $\Delta$  are known, and  $\lambda$  is some unknown quantity.
  - ▶ We can solve for  $\lambda$  in terms of the other quantities.
- Point
  - ▶ If we know the determinant of a matrix, we can solve for unknown variables inside the matrix.

$$\begin{vmatrix} a & b \\ c & \lambda \end{vmatrix} \Rightarrow a\lambda - bc = \Delta$$

Eq 1. Example of finding a missing matrix element

$$\begin{vmatrix} \lambda & 1 \\ 3 & \lambda \end{vmatrix} = 1 \Rightarrow \lambda^2 - 3 = 1 \Rightarrow \lambda^2 = 4, \quad \lambda = \pm 2$$

Eq 2. Numerical example to solve for unknown variable inside the matrix

# The Characteristic Polynomial of Matrix

- Combining **matrix shifting** with the determinant as shown in Eq 1..
- Why is it called a polynomial?
  - ▶ The shifted  $M \times M$  matrix has an  $\lambda^M$  term, therefore has  $M$  solutions.

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

Eq 1. The characteristic polynomial of the matrix

$$\begin{vmatrix} a - \lambda & b \\ c & d - \lambda \end{vmatrix} = 0 \Rightarrow \lambda^2 - (a + d)\lambda + (ad - bc) = 0$$

Eq 2. Example of the characteristic polynomial for  $2 \times 2$  matrices

# Numerical Example of the Characteristic Polynomial

■ Let's return to the  $2 \times 2$  case, this time using numbers instead of letters.

- ▶ Assume that it has a determinant of 0 after being shifted by some scalar  $\lambda$ .
- ▶ After some algebra, the two solutions are  $\lambda = \boxed{-2, 4}$ .

$$\det \left( \begin{bmatrix} 1 & 3 \\ 3 & 1 \end{bmatrix} - \lambda I \right) = 0$$

Eq 1. The characteristic polynomial of the matrix

*Eigenvalue.*

$$\begin{vmatrix} 1 - \lambda & 3 \\ 3 & 1 - \lambda \end{vmatrix} = 0 \Rightarrow (1 - \lambda)^2 - 9 = 0 \quad \lambda = \boxed{-2, 4}$$

Eq 2. Example of the characteristic polynomial for  $2 \times 2$  matrices

# Properties of the Characteristic Polynomial

## ■ Let's plug them back into the shifted matrix.

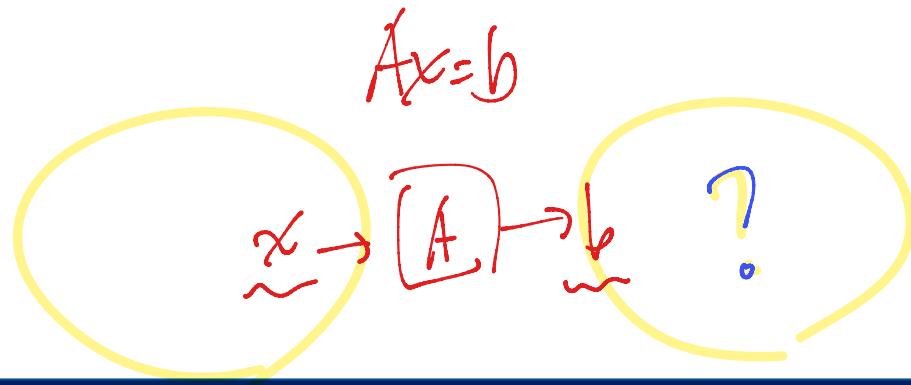
- ▶ Both matrices have nontrivial null spaces.
  - Some non-zeros vector  $y$  such that  $(A - \lambda I)y = 0$ .
  - In this case, any scaled version of  $[1 \ -1]$  and  $[1 \ 1]$ .

## ■ Every square matrix can be expressed as an equation.

- ▶ Directly links matrices to the fundamental theorem of algebra.
- ▶ The solutions to the characteristic polynomial set to  $\Delta = 0$  are the eigenvalues of the matrix.

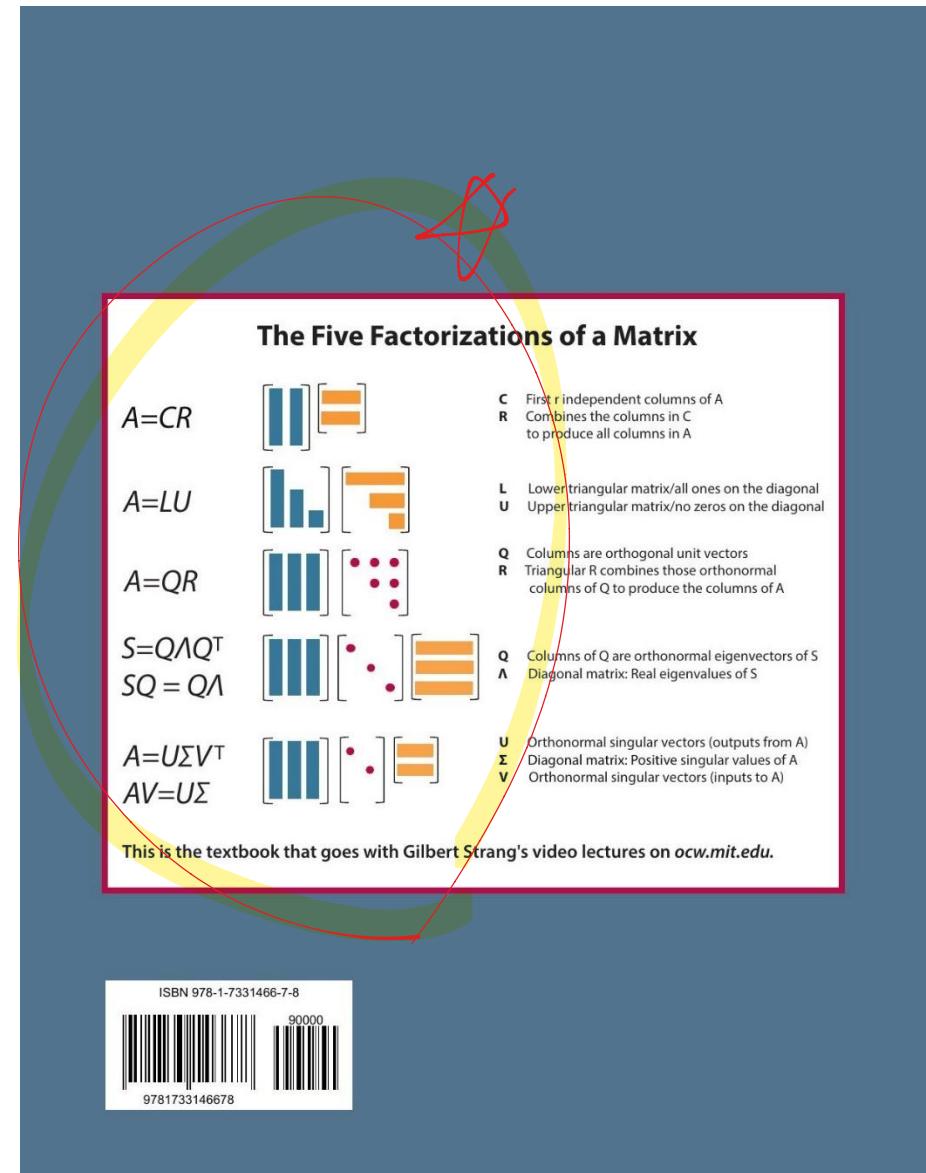
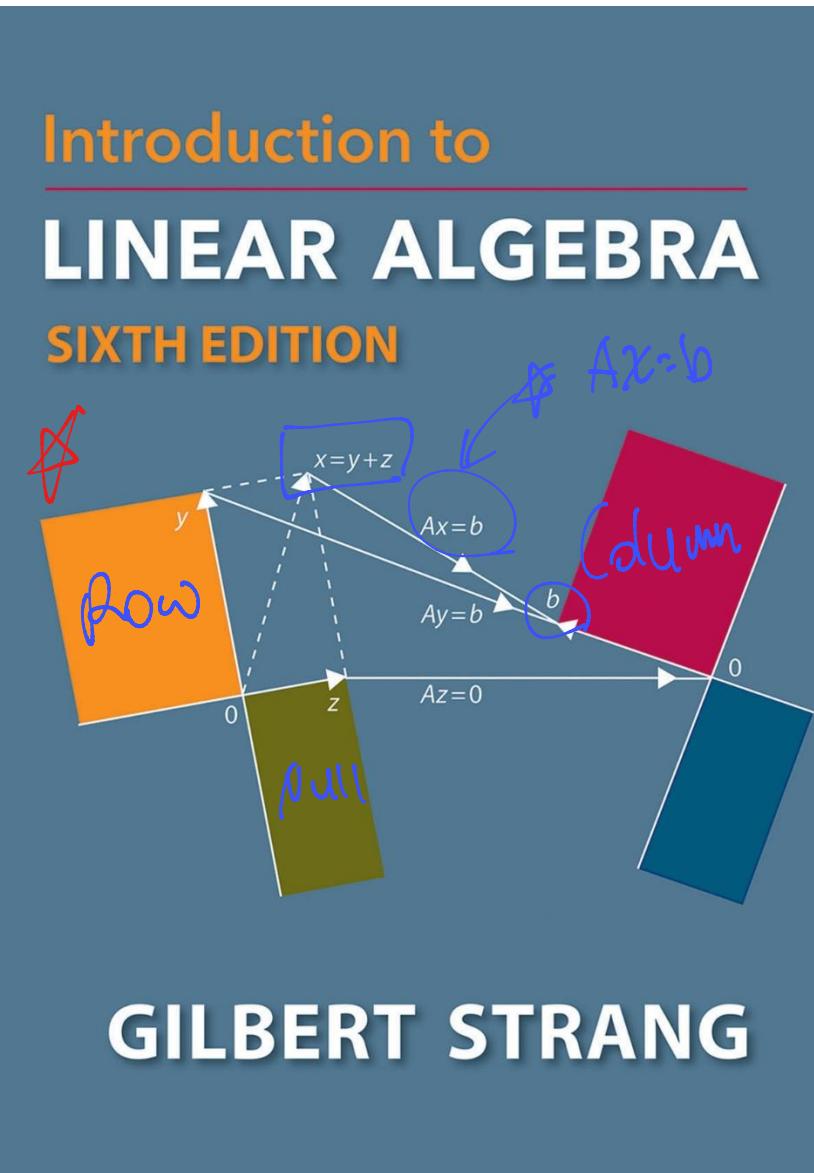
$$\lambda = -2 \Rightarrow \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix} \quad \lambda = 4 \Rightarrow \begin{bmatrix} -3 & 3 \\ 3 & -3 \end{bmatrix}$$

Eq 1. Shifted matrix



# Matrix spaces (column, row, nulls)

# Gilbert Strang – Linear Algebra



# Matrix Spaces

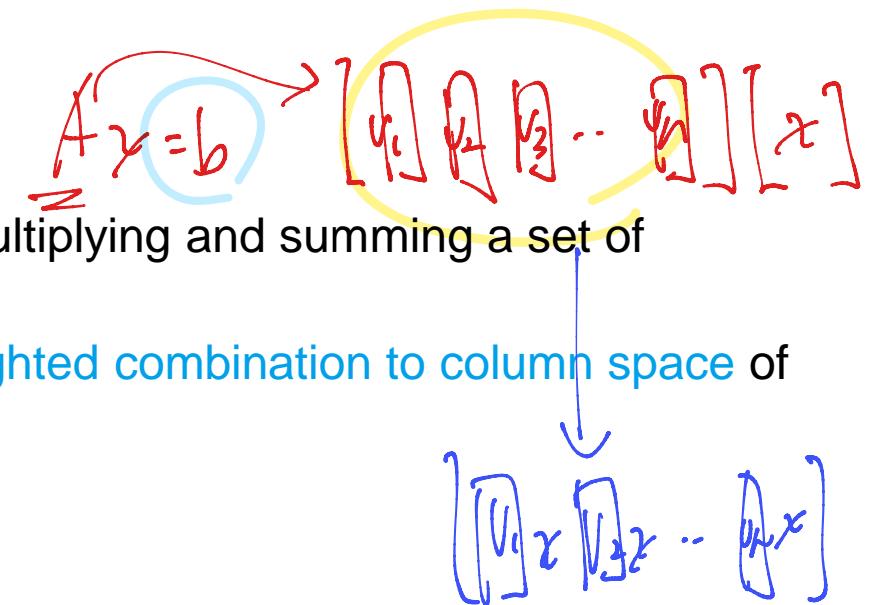
## ■ Concept of matrix spaces

- ▶ Linear transformation: linear weighted combinations of different features of a matrix.
  - Central to many topics in abstract and applied linear algebra.

# Column Space of Matrix

## ■ Remember in vector extension part...,

- ▶ A **linear weighted combination of vectors** involves scalar multiplying and summing a set of vectors.
- ▶ If two modifications to this concept will extend to **linear weighted combination to column space** of matrix.



## ■ Two modifications for **column space of matrix**

- ▶ Conceptualize a matrix as a set of column vectors.
- ▶ Consider infinity of real-valued scalars instead of working with a specific set of scalars.
  - Resulting infinite set of vectors is called **column space of a matrix**.

# Numerical Example of Column Space of Matrix

## ■ Expression of column space of matrix

- ▶  $C(A)$  indicates the column space of matrix  $A$ .
- ▶  $\lambda$  can be any possible real-valued number.

$$C \left( \begin{bmatrix} 1 \\ 3 \end{bmatrix} \right) = \lambda \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \lambda \in \mathbb{R}$$

Expression of column space

$$Ax \rightarrow b$$

~~but 2D doesn't have BL space~~

: Column Space

## ■ Mathematical meaning of above expression

- ▶ Column space is the set of all possible scaled versions of the column vector  $[1 \ 3]^T$ .
- ▶ Is the vector  $[2 \ 6]^T$  in the column space?
  - Yes
  - Can express that vector as the matrix times  $\lambda = 2$ .
- ▶ How about vector  $[1 \ 4]^T$ ?
  - No
  - Simply no scalar that can multiply the matrix to produce that vector.

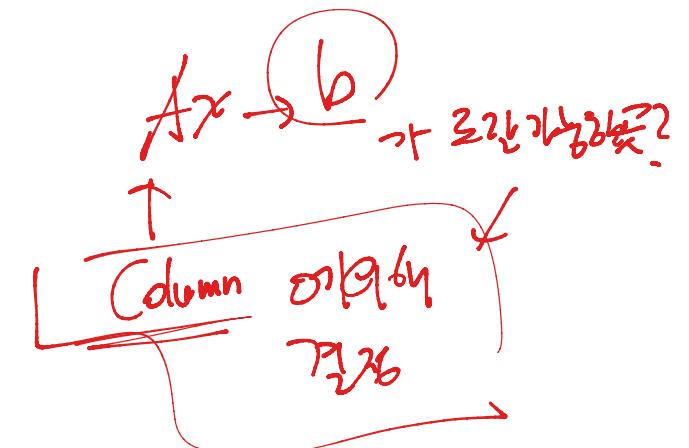
# Form of Column Space of Matrix with One Column

## ■ What does the column space look like?

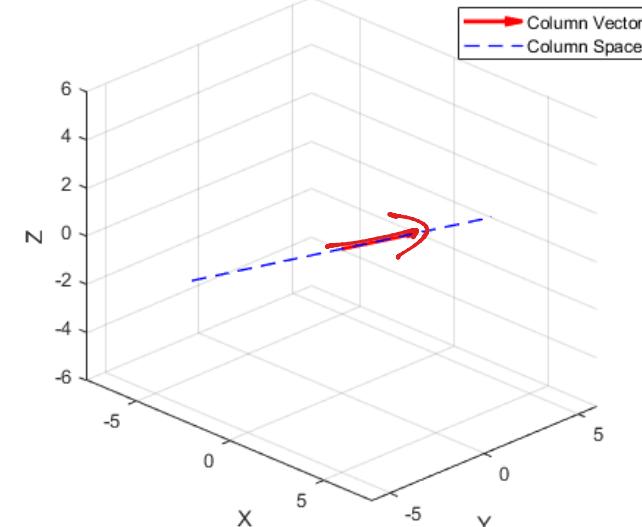
- ▶ Matrix with one column vector, column space is a line.
  - Passes through the origin, in the direction of the column vector.
  - Stretches out to infinity in both directions.

$$C(A) = C\left(\begin{bmatrix} 1 \\ 3 \end{bmatrix}\right) = \lambda \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \lambda \in \mathbb{R}$$

Example of Column space of a matrix that has only one column



Column vector and Column Space



Visualization of column space of a left matrix

# Example of Column Space of Matrix with Two Columns

## ■ Example of matrix with two columns

- ▶ Allow two distinct  $\lambda$ s.
  - Both real-valued numbers but can be different from each other.

$$C \left( \begin{bmatrix} 1 & 1 \\ 3 & 2 \end{bmatrix} \right) = \lambda_1 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \lambda_2 \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \lambda \in \mathbb{R}$$

Example of column space of a matrix with two column

$Ax \rightarrow b$  Space

- ▶ What is the set of all vectors that can be reached by some linear combination of these two column vectors ?
  - All vectors in  $\mathbb{R}^2$ .
  - For example, vector  $[-4 \quad 3]^T$  can be obtained by scaling the two columns.
    - In above matrix example, when  $\lambda_1, \lambda_2$  is 11 and -15, respectively.
  - These two columns can be appropriately weighted to reach any point in  $\mathbb{R}^2$ .
- ▶ Always two columns of matrix cover any point in  $\mathbb{R}^2$ ? **No!**

# Another Example of Column Space of Matrix with Two Columns

## ■ Example of another matrix with two columns

$$C \left( \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix} \right) = \lambda_1 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \lambda_2 \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \lambda \in \mathbb{R}$$

Example of column space of another matrix with two column

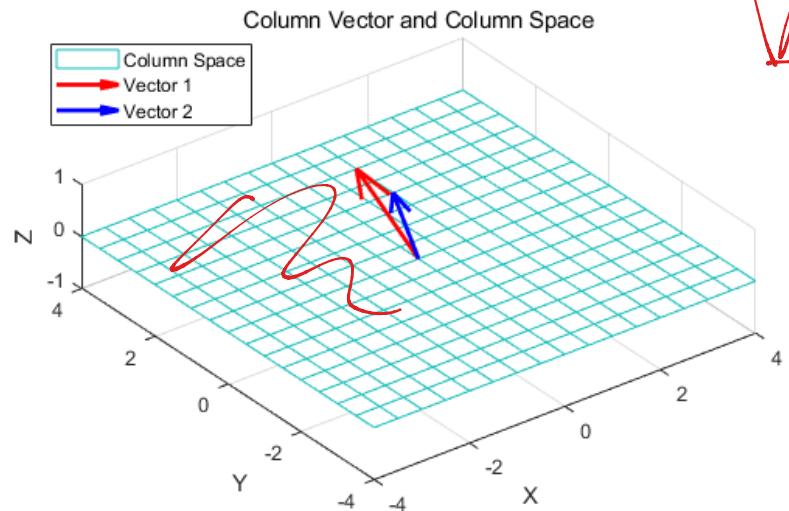
- ▶ Is it possible to reach any point in  $\mathbb{R}^2$  ?
  - No
  - Can't produce another vector(ex.  $[3 \ 5]^T$ ) by linear combination of the two columns.
  
- ▶ What is dimensionality of its column space ?
  - Two columns are **colinear**.
    - One is already a scaled version of the other.
  - Column space of this  $2 \times 2$  matrix is still just a **1D** subspace.
  
- ▶ Having  **$N$  columns** in a matrix does not guarantee that the column space will be  **$N$  dimension**.
  - Dimensionality of **columns space equals the number of columns** only if the **columns form a linearly independent set**.

# Visualization of Previous Two Examples of Column Spaces

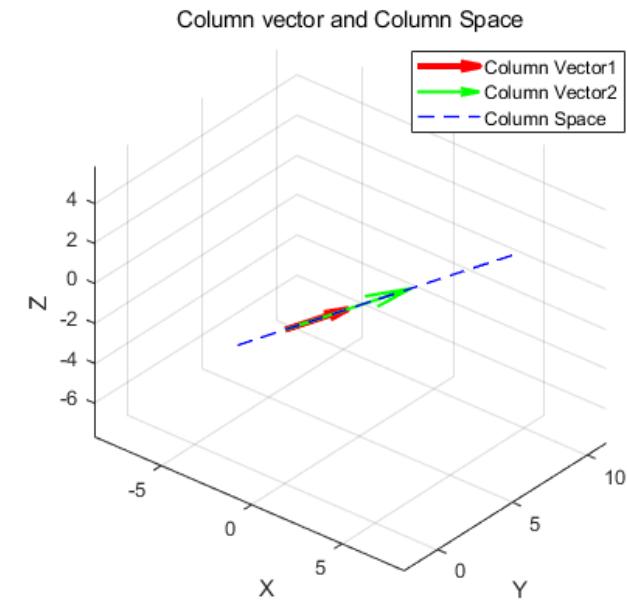
Asking  $\vec{v}_1, \vec{v}_2 \rightarrow$  b!.  $\rightarrow$  3D vector space.

$$C\left(\begin{bmatrix} 1 & 1 \\ 3 & 2 \end{bmatrix}\right) = \lambda_1 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \lambda_2 \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \lambda \in \mathbb{R}$$

$$C\left(\begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}\right) = \lambda_1 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \lambda_2 \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \lambda \in \mathbb{R}$$



Visualization of column space of first example matrix



Visualization of column space of second example matrix

# Example of Column Space of Matrix with Two Columns in $\mathbb{R}^3$

## ■ Example of matrix with two columns in $\mathbb{R}^3$

- ▶ Two columns in below matrix are linearly independent.
  - Can't express one as a scaled version of the other.
  - Column space of this matrix is  $2D$ , but it is  $2D$  plane embedded in  $\mathbb{R}^3$ .
- ▶ Column space of this matrix is an infinite  $2D$  plane, but it is merely an infinitesimal slice of  $3D$ .
  - Like an infinitely thin piece of paper that spans universe.

$$C \left( \begin{bmatrix} 3 & 0 \\ 5 & 2 \\ 1 & 2 \end{bmatrix} \right) = \lambda_1 \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix}, \lambda \in \mathbb{R}$$

Example of column space of a matrix with two columns and three rows

# Code Exercise of Column Space

## Code Exercise (06\_02)

```
% Clear workspace, command window, and close all figures
clc; clear; close all;

% Define two 3D vectors
v1 = [3; 5; 1];
v2 = [0; 2; 2];

% Calculate the normal vector (cross product of the two vectors)
normal = cross(v1, v2);

% Choose a point on the plane (for example, using v1)
point = v1;

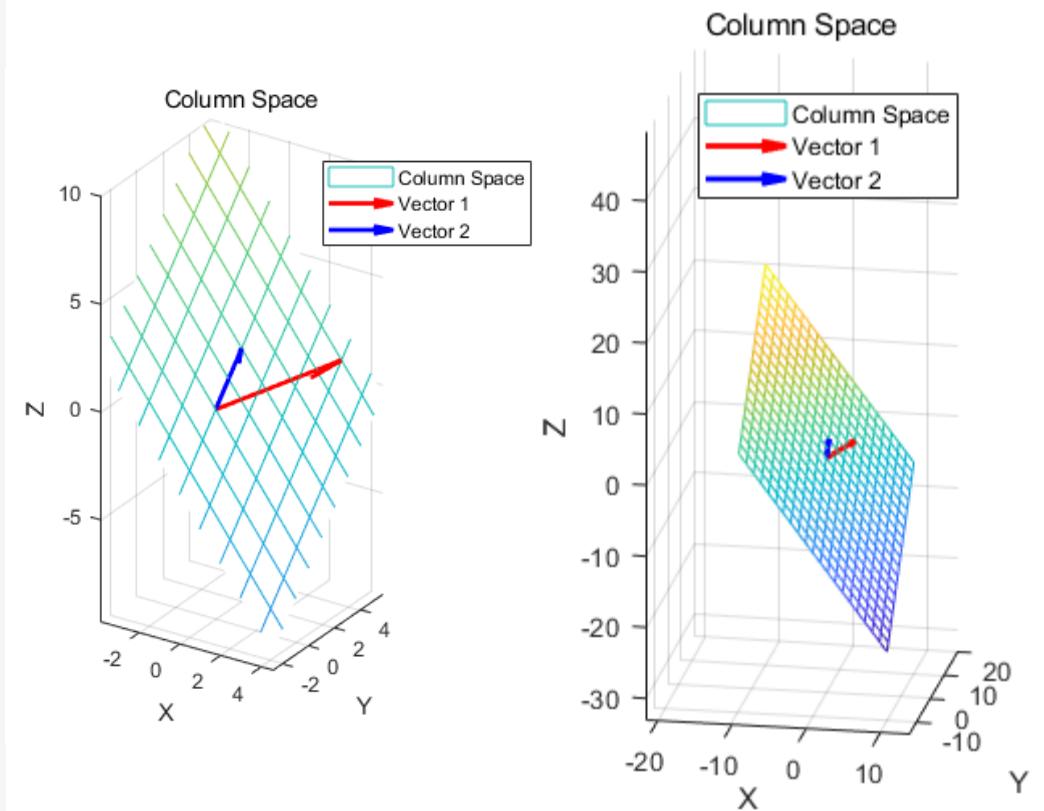
% Equation of the column space: ax + by + cz = d
% where a, b, c are components of the normal vector, and d is the constant term of the column space equation
a = normal(1);
b = normal(2);
c = normal(3);
d = -dot(normal, point);

% Code for visualization
[x, y] = meshgrid(-10:1:10, -10:1:10); % Create a grid to represent the column space
z = (-d - a*x - b*y) / c; % Calculate z values of the column space

% Draw the column space
figure;
mesh(x, y, z);
hold on;

% Draw the two vectors
quiver3(0, 0, 0, v1(1), v1(2), v1(3), 'r', 'LineWidth', 2, 'AutoScale', 'off', 'MaxHeadSize', 1);
quiver3(0, 0, 0, v2(1), v2(2), v2(3), 'b', 'LineWidth', 2, 'AutoScale', 'off', 'MaxHeadSize', 1);

xlabel('X'); ylabel('Y'); zlabel('Z');
title('Column Space');
legend('Column Space', 'Vector 1', 'Vector 2');
axis equal;
xlim([-20 10]); ylim([-10 20]); zlim([-30 40]);
grid on;
```



MATLAB code example of Column Space and results

# Row Space

## ■ Exact same concept of column space.

- ▶ Consider all possible **weighted combinations of the rows** instead of the columns.

## ■ Expression of row space

- ▶ Indicated as  $R(A)$ .

## ■ Properties of row space

- ▶ Row space of a matrix is the column space of the matrix transposed.
  - $R(A) = \boxed{C(A^T)}$
- ▶ Two matrix spaces are identical for symmetric matrices.
  - Because row space equals the column space of the matrix transpose.

# Reminder of Column Space

축공간(Column Space)

$$Ax=0$$

## ■ Column space

- ▶ Summarizing column space as the following equation.

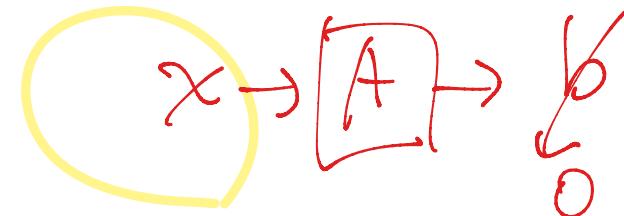
$$Ax = b$$

Column space equation

## ■ Can we find some set of coefficients in $x$ such that the weighted combination of columns in $A$ produce $b$ ?

- ▶ If yes,  $b \in C(A)$  and vector  $x$  tells how to weight the columns of  $A$  to get to  $b$ .

## ■ How about when $b = 0$ ?



# Null Space

## ■ Null space

Null space

- ▶ Summarizing null space as the following equation.

$$Ay = 0$$

Null space equation

## ■ Can we find some set of coefficients in $y$ such that the weighted combination of columns in $A$ produces the zeros vector 0?

- ▶ Set  $y = 0!$  (multiplying all columns by 0s will sum to the zeros vector.)
  - But it's a trivial solution and we will exclude it.
  - Therefore, question becomes different.

## ■ Can we find a set of weights – not all of which are 0 – that produces the zeros vector?

- ▶ Any vector  $y$  that can satisfy this equation is in null space of  $A$ .
  - Notation:  $N(A)$

# Example of Null Space

## ■ Find null vector $y$ when matrix is as below:

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

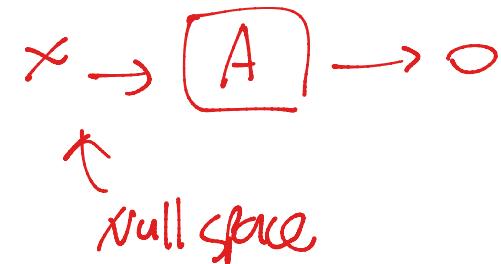
Example equation of null space

- ▶ Infinite number of vectors  $y$  that satisfy  $Ay = \mathbf{0}$  for that specific matrix  $A$ .
  - Answer can be vector  $[1 \ 1]$ ,  $[-1 \ -1]$  or  $[7.34 \ 7.34]$ .

## ■ Null space of this matrix can be expressed as:

$$N(A) = \lambda \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \lambda \in \mathbb{R}$$

Expression of null space of this matrix



# Another Example of Null Space

■ Find null vector  $y$  when matrix is as below:

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

Second example equation of null space

► Did you find the any vector  $y$  that satisfy  $Ay = \mathbf{0}$ ?

- Matrix  $A$  has no null space.

■ Null space of this matrix is the **empty set** and can be expressed as:

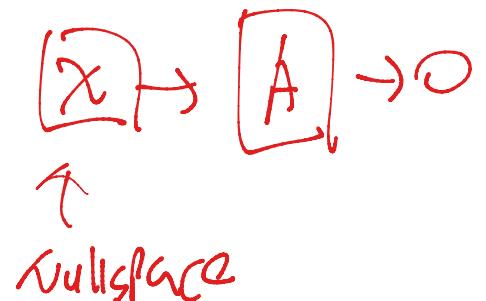
$$N(A) = \boxed{\emptyset}$$

Expression of null space of this matrix

# Mathematical Property of Null Space

## Difference between first example and second example

- ▶ First matrix
  - Contains columns that can be formed as scaled versions of other columns.
- ▶ Second matrix
  - Contains columns that form an **independent set**.



## What can we find from this properties?

- ▶ Tight relationship between **dimensionality of null space** and **linear independence of columns** in a matrix
  - This relationship is given by the rank-theorem, which we will learn later.
- ▶ Key point: **Null space is empty when the columns of matrix form a linearly independent set.**

$$A_1 = \begin{bmatrix} 1 & -1 \\ -2 & 2 \end{bmatrix}$$

First Example equation of null space

$$A_2 = \begin{bmatrix} 1 & -1 \\ -2 & 3 \end{bmatrix}$$

Second example equation of null space

# Compute Null Space of a Matrix in MATLAB

## ■ Code Exercise (06\_03)

```
% Clear workspace, command window, and close all figures  
clc; clear; close all;  
  
% Define matrices A and B  
A = [1 -1; -2 2];  
B = [1 -1; -2 3];  
  
% Calculate the null spaces of A and B  
nullA = null(A);  
nullB = null(B);  
  
% Display the null spaces  
disp('Null space of matrix A:');  
disp(nullA);  
disp('Null space of matrix B:');  
disp(nullB);
```

MATLAB code example of Null Space

## ■ MATLAB code and result

- ▶ Why did MATLAB choose 0.70710678 as the answer?
  - It's easier to chose 1.
  - MATLAB returned a Unit Vector.
- ▶ Advantages of unit vector
  - Convenient to work with.
  - Have several nice properties including numerical stability.

# Shape of Null Space

## ■ Row space is orthogonal to the null space.

▶ Definition of the null space:  $Ay = 0$

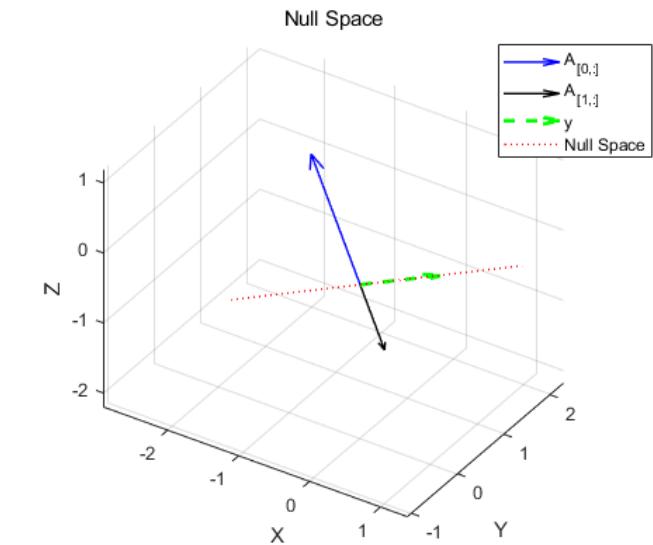
- Each row of the matrix ( $a_i$ ) leads to the expression  $a_i y = 0$ .
- Dot product between each row and null space vector is 0.

$$\begin{bmatrix} \vdots & \vdots & \vdots \\ a_1 & a_2 & a_3 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = 0$$

## ■ Every matrix has four associated subspaces.

- ▶ Column, row, null space.
- ▶ Fourth space is called right null space.
  - Null space of the rows.
  - Written as:  $N(A^T)$

$$x \rightarrow (A) \rightarrow b$$



Visualization of the null space of a matrix

## ■ Why all the fuss about null space?

- ▶ Null space is the keystone.
  - To find eigenvectors and singular vectors.
  - You'll learn later.

# Inverse matrix, column space, and null space

# Linear System of Equations

$$Ax = b$$

## ■ System of equations.

- ▶ List of variables
- ▶ List of equations

## ■ Among system of equation, “linear system of equations” has a special form equation.

- ▶ Each variable is that it's scaled by some constant.
- ▶ Scaled variables are added to each other.

$$\vec{x} = A\vec{x} + \vec{b}$$

$$\frac{d}{dx} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [A] \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} : \end{bmatrix} h$$

## ■ Examples of linear system of equations.

- ▶  $2x + 5y + 3z = -3$
- ▶  $4x + 0y + 8z = 0$
- ▶  $1x + 3y + 0z = 2$

## ■ Linear system of equations can be packaged together into a single vector equation.

$$\begin{bmatrix} 2 & 5 & 3 \\ 4 & 0 & 8 \\ 1 & 3 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 2 \end{bmatrix}$$

Coefficients      Variables      Constants  
 $A$                    $\vec{x}$                    $\vec{v}$

■ So, linear equation can be represented as  $A\vec{x} = \vec{v}$ .

# Meaning of $\vec{v}$

$$A\vec{x} = \vec{v}$$

■ Matrix  $A$  corresponds with some linear transformation. So, solving  $A\vec{x} = \vec{v}$  means we're looking for a vector  $\vec{x}$ , which, after applying the transformation, lands on  $\vec{v}$ .

## Example for solving $A\vec{x} = \vec{v}$ .

- ▶ Assume that there are two equations  $2x + 2y = -4$ , and  $1x + 3y = 0$ . Then, linear equation can be represented as  $\begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -4 \\ -1 \end{bmatrix}$ .

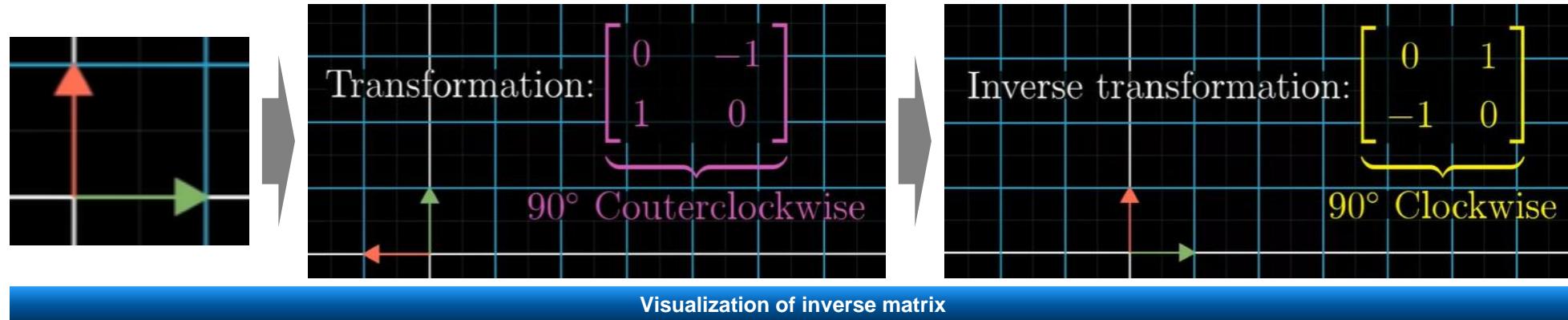
- ▶ When  $\text{Det}(A) \neq 0$  case which means, space does not get squished into a zero-area region.

- There will always be **only one vector** that lands on  $\vec{v}$ , and you can find it by playing the **transformation in reverse**.

Transformation in reverse corresponds to the “inverse of  $A^n$  ( $A^{-1} = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}^{-1}$ ).

## Example of inverse matrix.

1. If  $A$  is a **90-degree rotation**, then  $A^{-1}$  rotates **clockwise 90-degree**.
2. If  $A$  is a **rightward shear** that pushes  $\hat{J}$  to the right ,then  $A^{-1}$  will be a **leftward sheer** that pushes  $\hat{J}$  to the left.



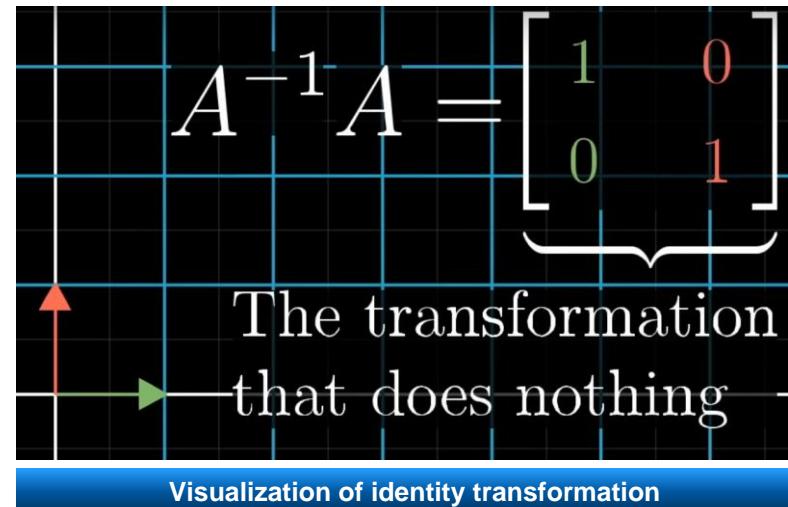
# Properties of Inverse Matrix

## ■ Properties of $A^{-1}$

- ▶ If you first **apply  $A$**  then follow it with the **transformation  $A^{-1}$** , you end up back where you started.
- ▶  $A^{-1}A$  corresponds to doing nothing, which is called "**identity transformation**". In this case,  $\hat{I}$  and  $\hat{J}$  are unmoved, so its matrix is equal as  $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ .

## ■ Therefore, to know the initial position of $\vec{x}$ in $A\vec{x} = \vec{v}$ , you can multiply $A^{-1}$ by both sides to create $\vec{x} = A^{-1}\vec{v}$ .

- ▶  $A^{-1}A\vec{x} = A^{-1}\vec{v}$ , then  $\vec{x} = \boxed{A^{-1}\vec{v}}$



# Formula of $\det(A) \neq 0$

- Two equations for two unknowns form a system of equations, and when these equations are the only ones that have a solution, then  $\det(A) \neq 0$ .
- The formula is also valid for linear systems for many unknowns.
  - ▶ If each unknown has a unique solution, the determinant is not zero.
- If  $\det(A) \neq 0$ ,  $A^{-1}$  exists.

One unique solution . . .

$$\overbrace{ax + cy = e}^{}$$

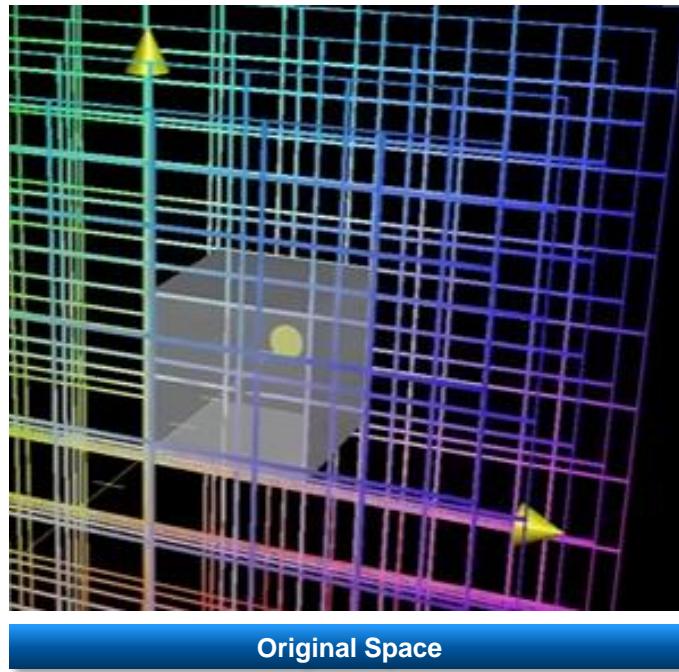
$$bx + dy = f$$

Formula of  $\det(A) \neq 0$

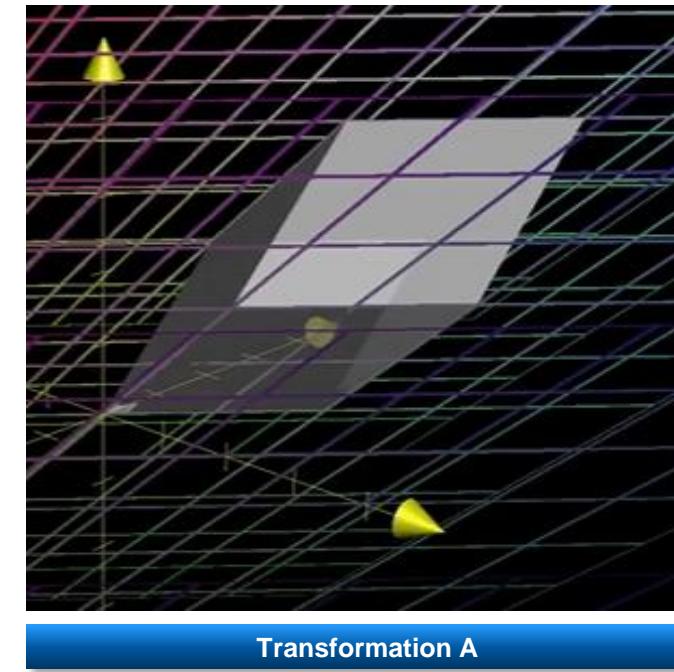
# Geometric Analysis of Inverse Matrix

## ■ System of equations can be interpreted geometrically.

- ▶ We have Transformation  $A$ , vector  $\vec{v}$ , vector  $\vec{x}$  that reaches  $\vec{v}$  through transformation.
- ▶ Applying a  $A$  to a particular matrix and then applying the  $A^{-1}$  can be reversed for the first time.

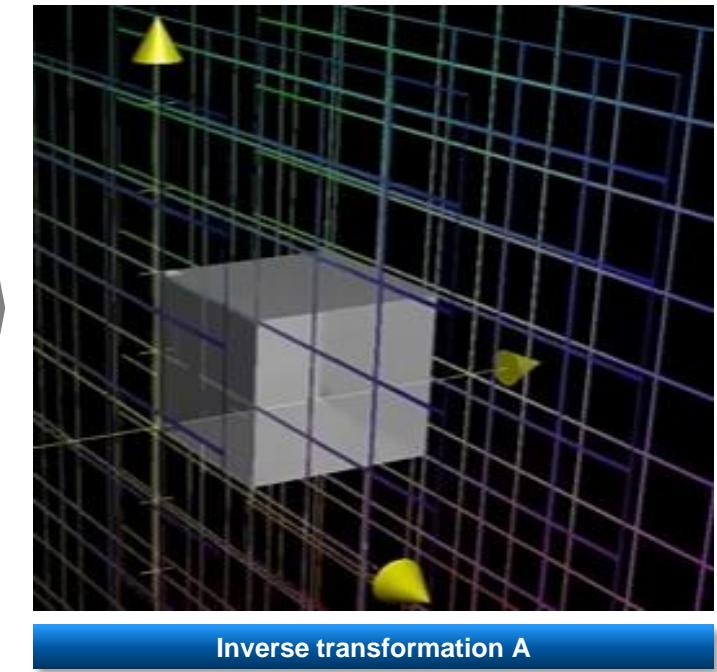


Original Space



Transformation A

$$A\vec{x} = \vec{v}$$



Inverse transformation A

$$A^T A\vec{x} = A^T \vec{v}$$

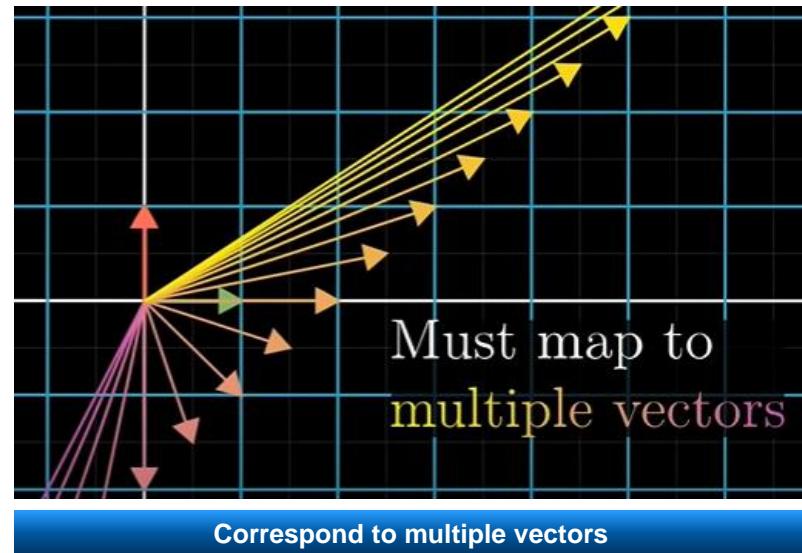
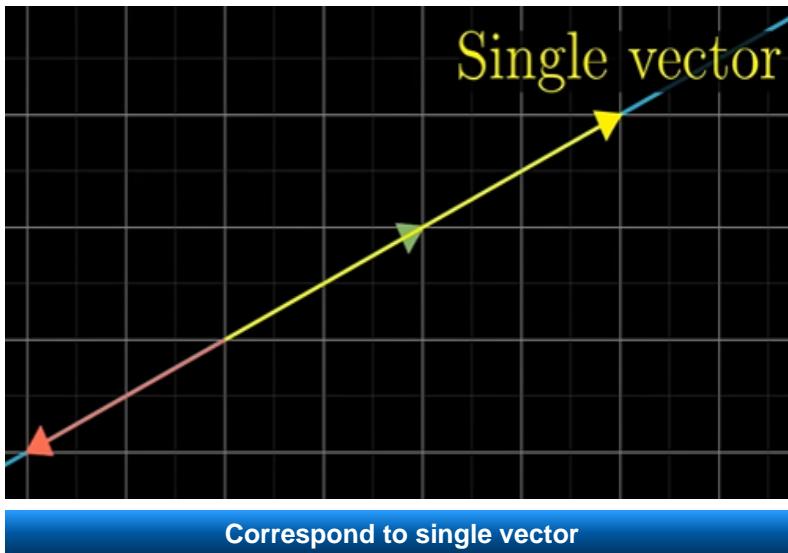
# $A\vec{x} = \vec{v}$ when $\det(A) = 0$

## ■ When the $\det(A) = 0$

- ▶ Transformation squishes space into a smaller dimension.
- ▶ There is no inverse matrix.

## ■ Function cannot unsquash a line to turn it into a plane.

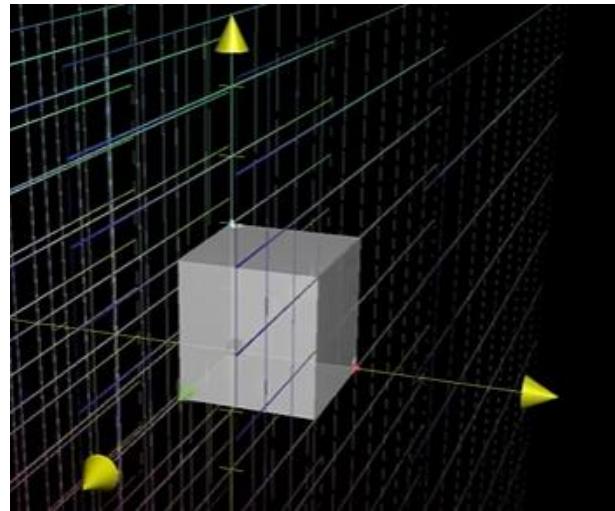
- ▶ Functions can only take a single output for a single input.
- ▶ However, if a straight line is stretched to a plane, one vector must correspond to multiple vectors.



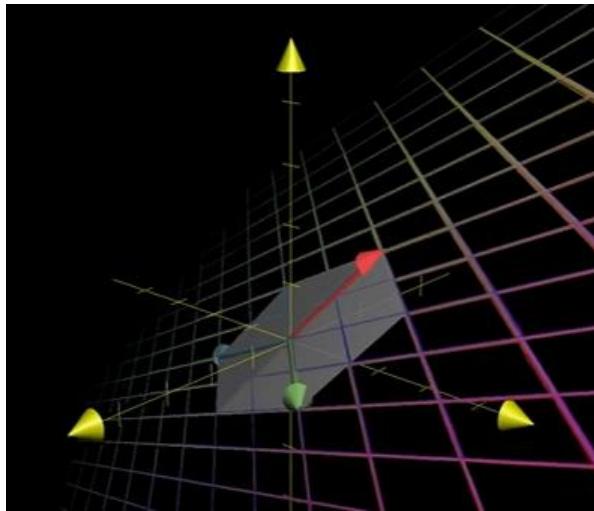
# Geometric Transformation for Three Equations

## ■ Three equations for three unknown variables

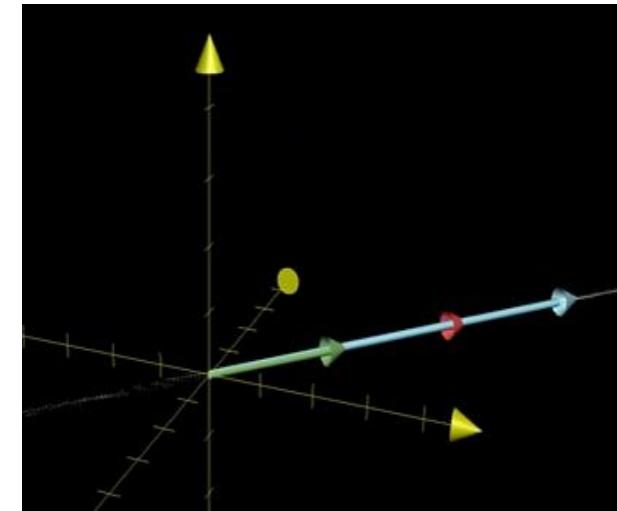
- If the transformation matrix squishes three dimensional space to a plane, straight line, or point, the reverse does not exist.



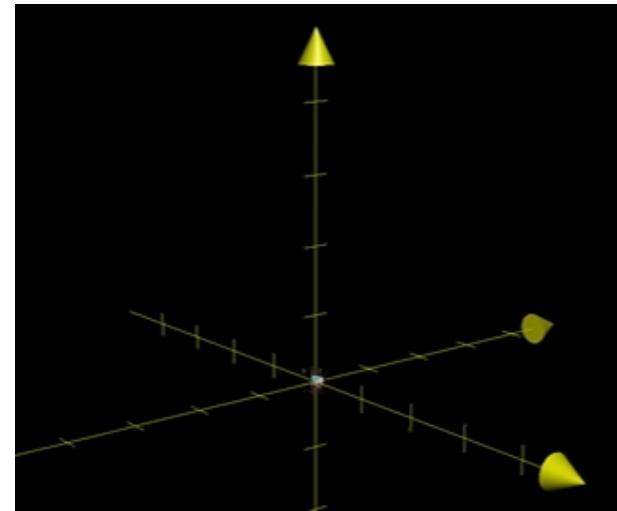
Cube in 3D



Transformation to plane in 3D



Transformation to straight line in 3D

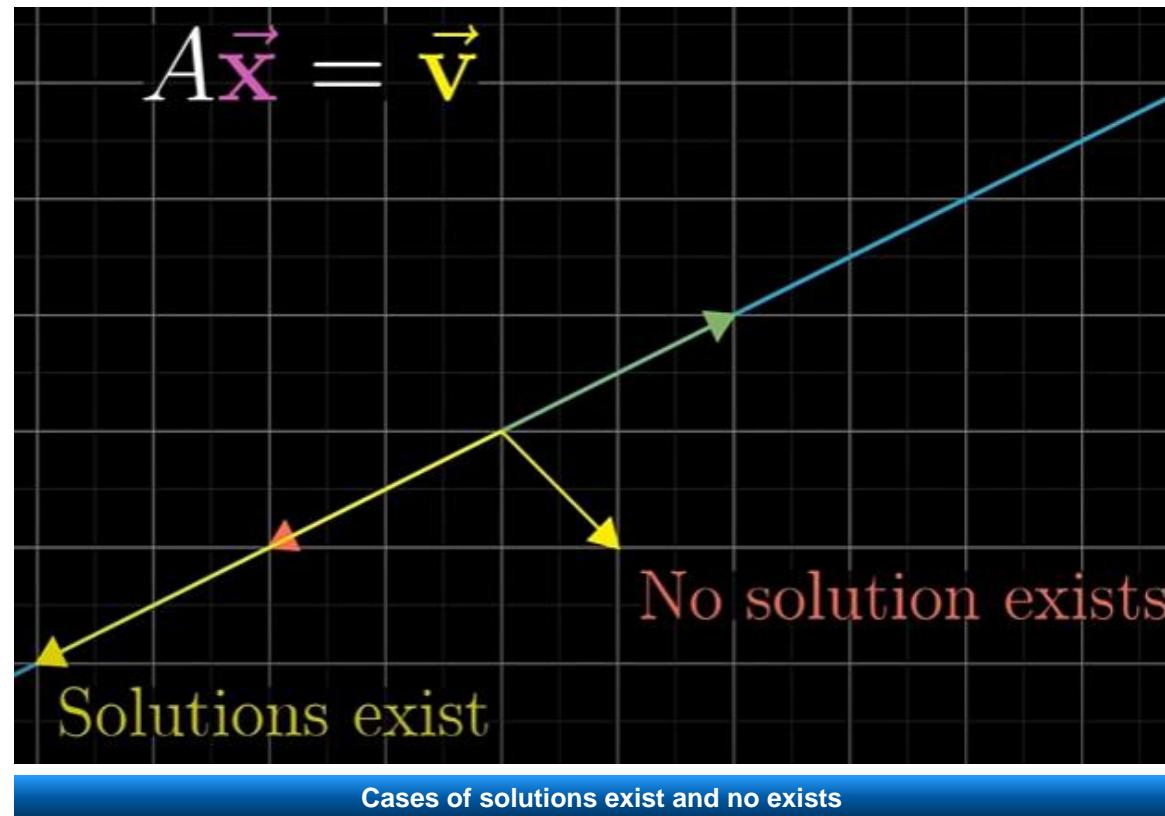


Transformation to point in 3D

# Cases Where $\det(A) = 0$ , but a Solution Exists

## ■ When the transformation matrix $A$ converts a spatial vector $\vec{x}$ into a line vector $\vec{x}$ ,

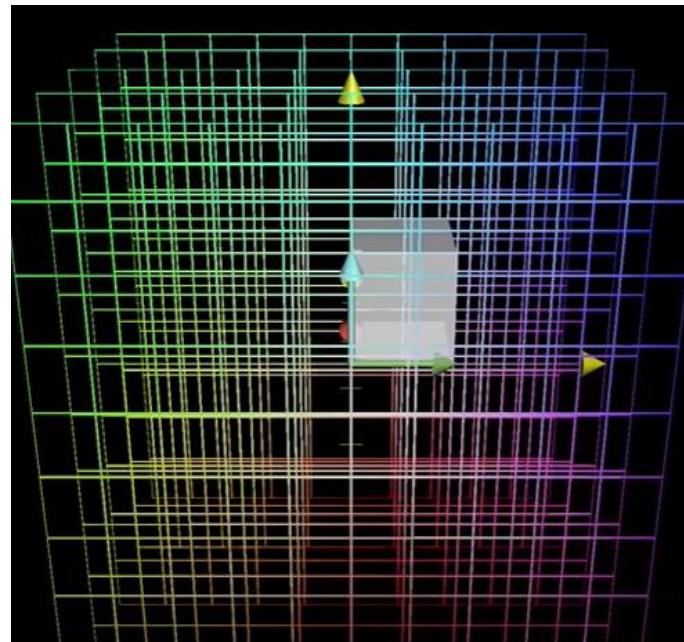
- ▶ If the vector  $\vec{v}$  exists on the transformed vector  $x$  after the transformation, there are infinitely many solutions.
- ▶ However, if the vector  $\vec{v}$  does not exist on the transformed vector  $\vec{x}$ , there are no solutions.



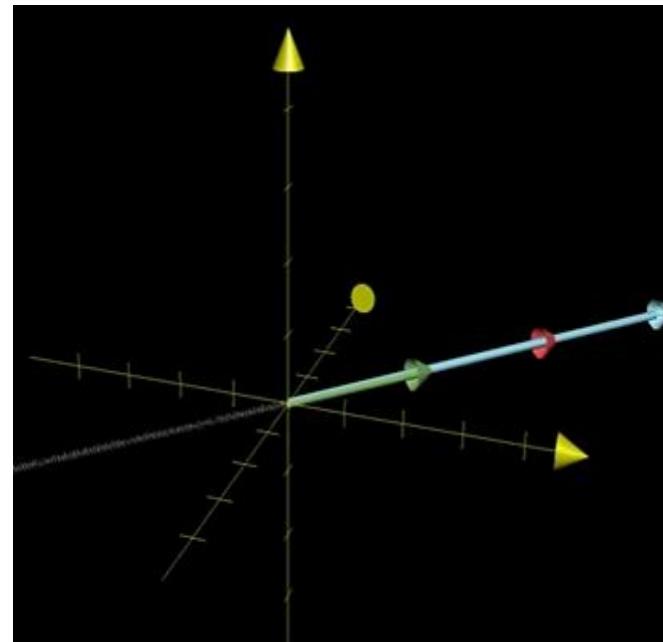
# Transformation of Space Vector in Case of $\det(A) = 0$

## ■ If $\det(A) = 0$ in 3D

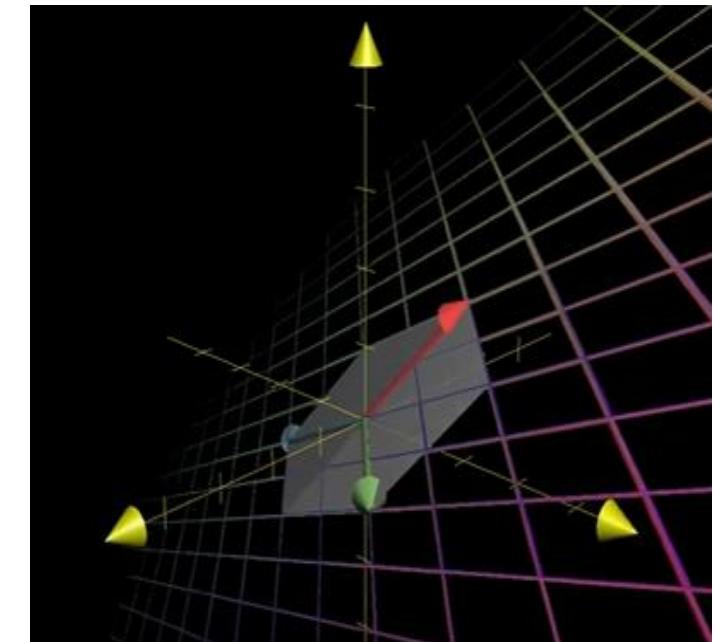
- ▶ 3D Space may be squished into a straight line.
- ▶ Or space can be squished to a plane.
- ▶ More specific expression is possible through "[rank](#)" rather than just  $\det(A) = 0$ .



Initial 3D space



Squished to straight line



Squished to plane

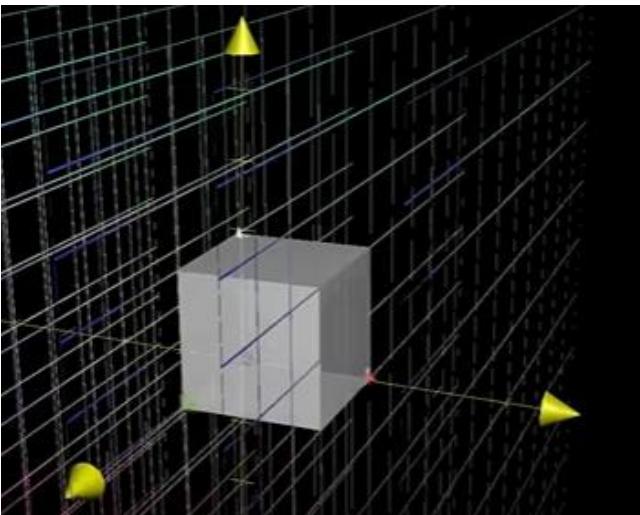
# Rank in Geometric Space

$$\boxed{Ax=b}$$

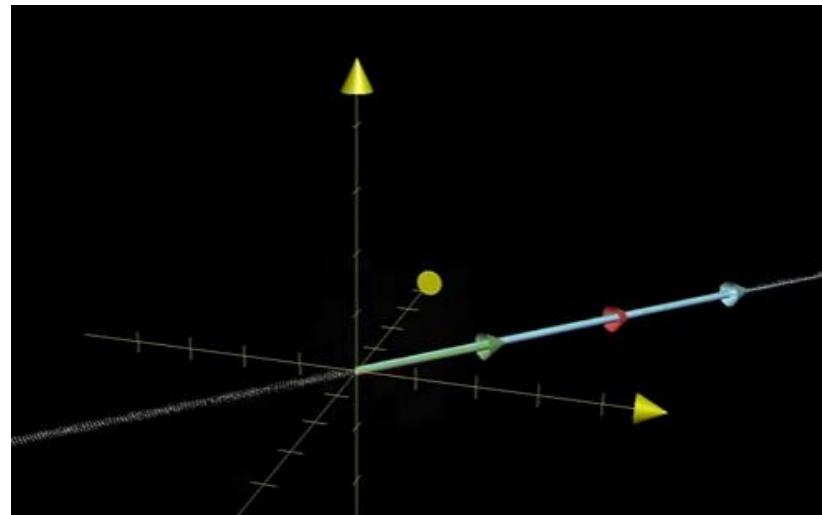
## ■ After applying transformation matrix $A$ ,

- ▶ If the output is **line**, that is, one-dimensional: **rank 1**
- ▶ If the output is **plane**, that is, two dimensional: **rank 2**

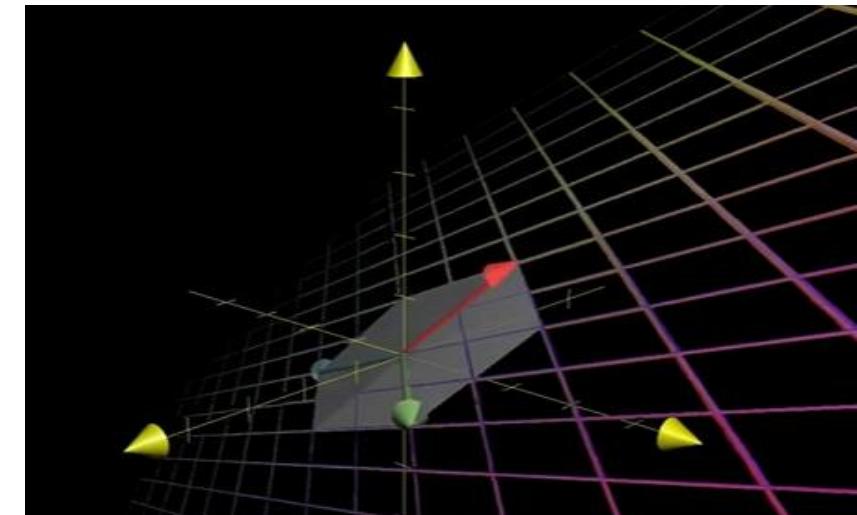
■ In other words, "rank" refers to the number of dimensions of the output after conversion.



Original 3D matrix



One-dimensional: rank 1



Two-dimensional: rank 2

# Number of Dimensions of the Output “Rank”

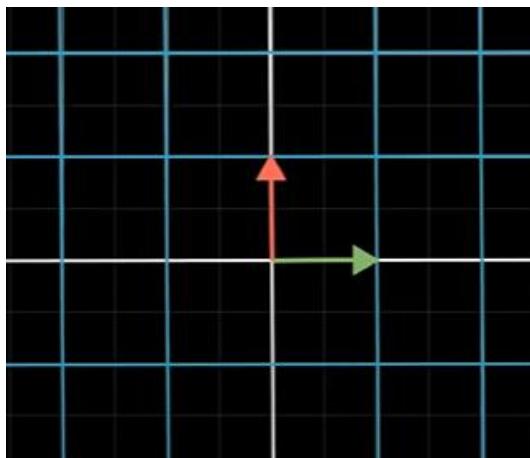
## ■ For a $2 \times 2$ matrix, the maximum rank is 2.

- ▶ In other words, a  $2 \times 2$  matrix has a base vector in rank 2 that creates all two-dimensional spaces.
- ▶  $2 \times 2$  matrix has a nonzero determinant.

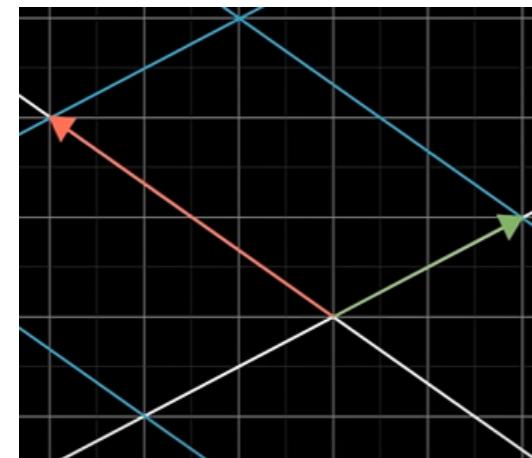
## ■ Also, for a $3 \times 3$ matrix, the maximum rank is 3.

- ▶ However, if  $\det(A) = 0$ , the rank may be reduced to 2 or 1.

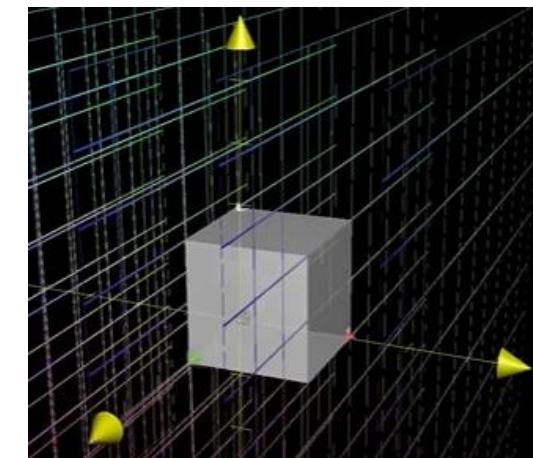
## ■ The set of all outputs for a matrix is called the “Column Space of the matrix”.



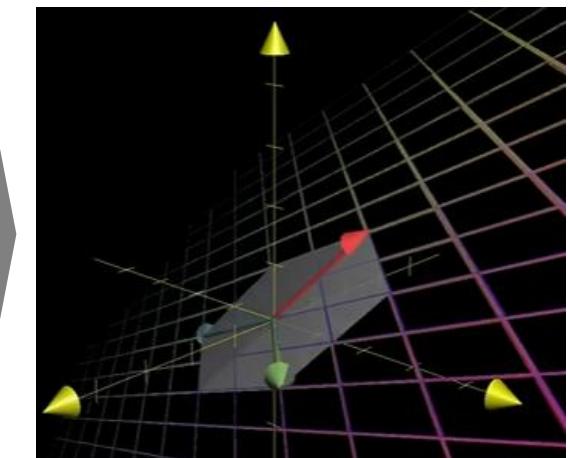
Original 2D matrix



Rank 2

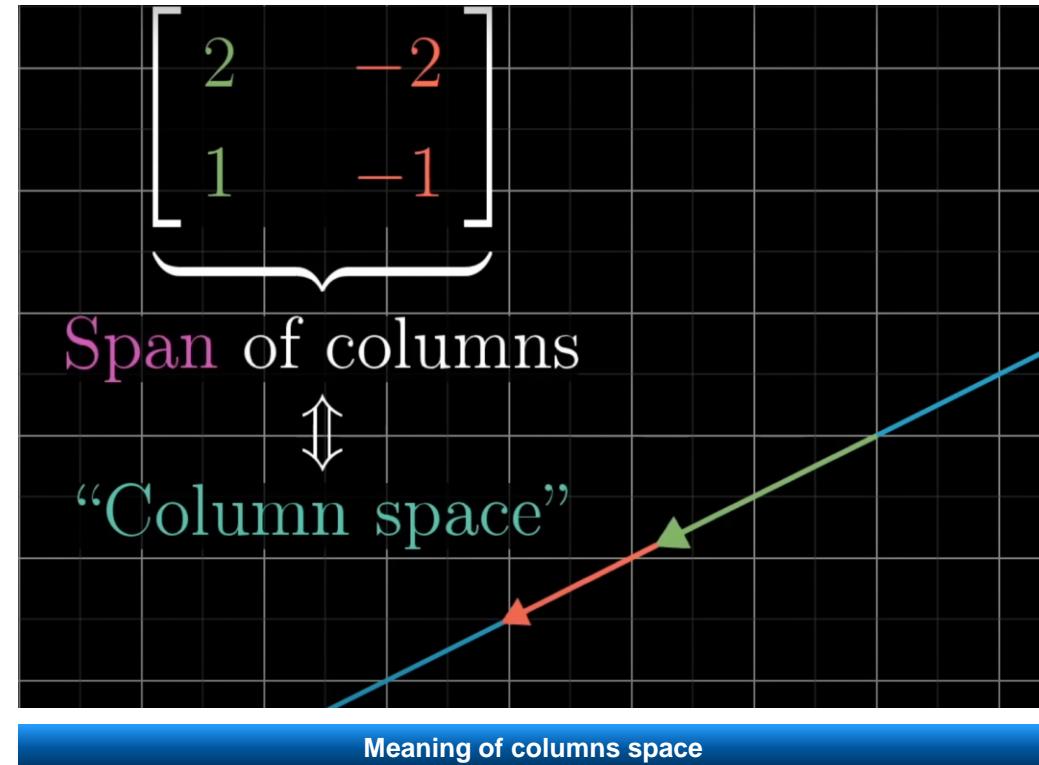


Original 3D matrix

Rank 2 when  $\det = 0$

# Column Space of Matrix

- The columns of a matrix indicate the positions of the basis vectors they reach.
- The span of transformed basis vectors represents all possible outputs.
- In other words, the column space is the *Span* of the columns of the matrix.
  - ▶ Therefore, the rank is equivalent to the number of dimensions of the column space.



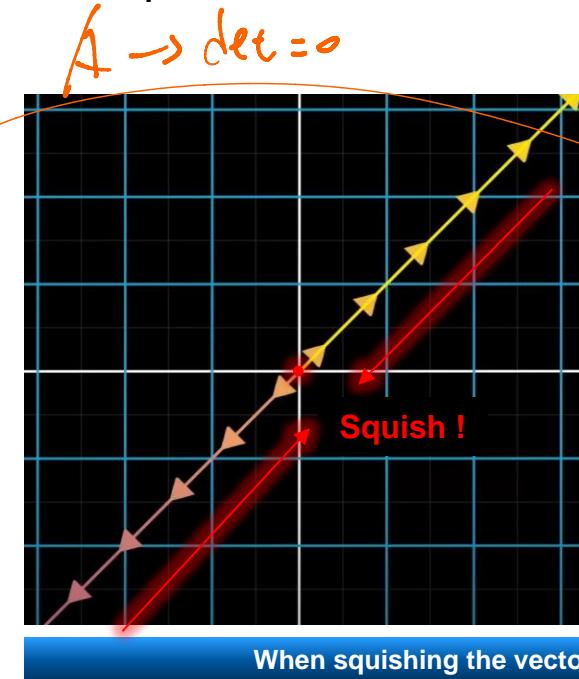
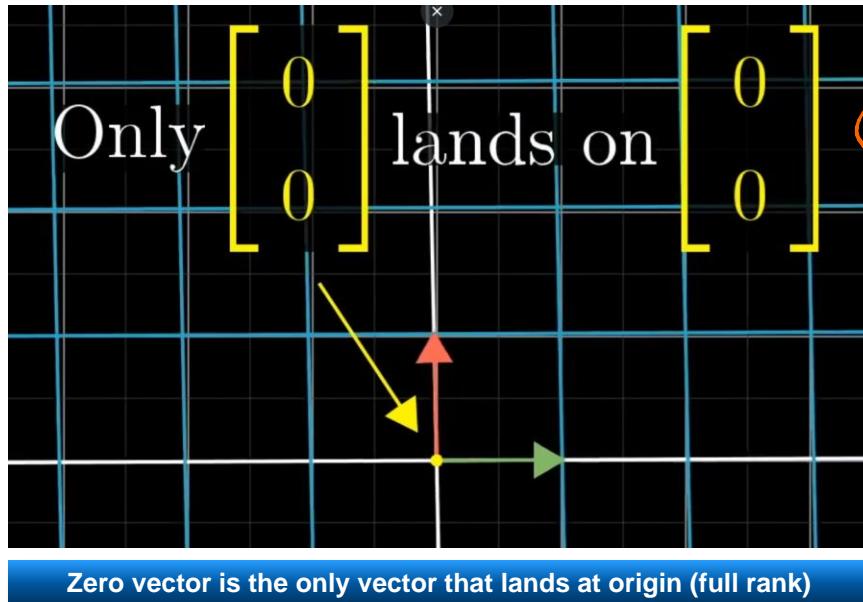
# Full Rank

■ A matrix is said to be of “full rank” when its rank is maximal.

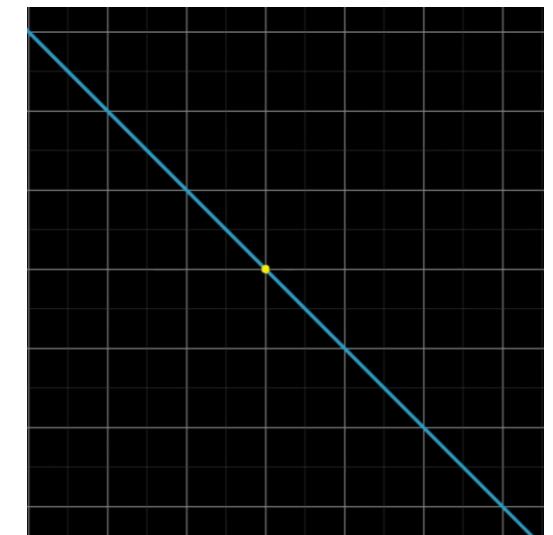
- ▶ Rank equals the number of columns in the matrix.

■ Since linear transformations must always keep the origin fixed, all column spaces necessarily include the zero vector.

- ▶ In a full rank transformation, the only vector that reaches the origin is the zero vector itself.
- ▶ For matrices that aren't full rank, which squish to a smaller dimension, there are a whole bunch of vectors that land on zero.



$$Ax = 0$$

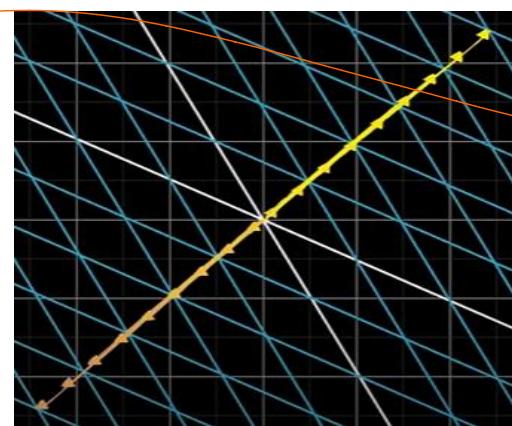
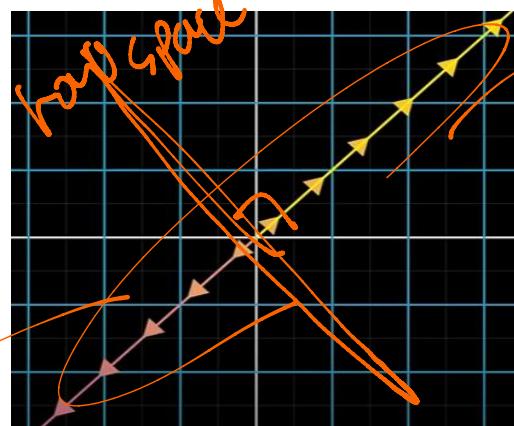


# Null Space of Matrix: Squishes 2D Space to Line

- For full rank, the only vector that reaches the origin by transformation is the zero vector itself.

$\downarrow$   
 $Ax = 0$   
 Full Rank

- In cases where the matrix is not of full rank, the space squishes to a lower dimension, resulting in a full line of vectors that can reach the origin.
- Case 1: 2D transformation squishes **2D space onto a line**.
  - Separated line in a different direction full of vectors that get squished onto the origin.
- Case 2: 3D transformation squishes 3D space to a plane.
- Case 3: 3D transformation squishes 3D space to a line.

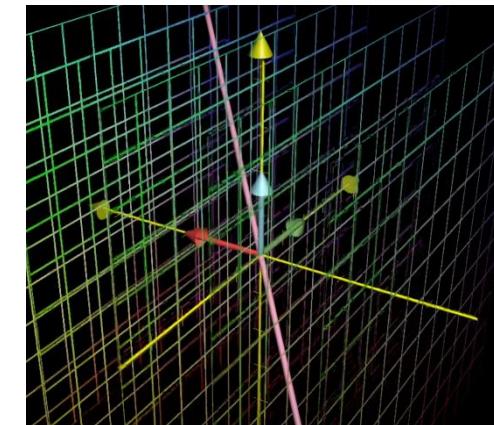
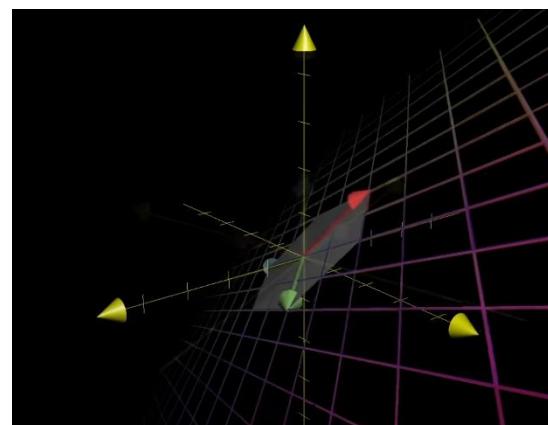
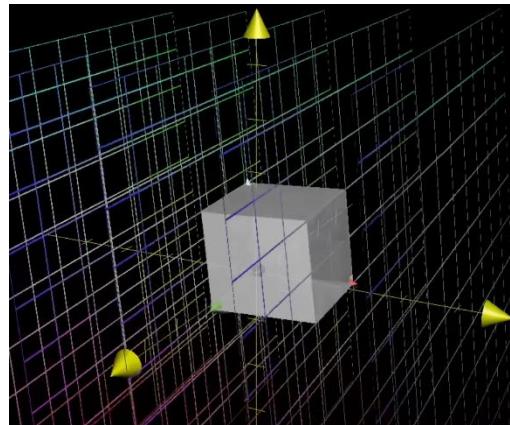


If 2D transformation squishes space to a line, a line in a different direction full of vectors that get squished onto the origin

# Null Space of Matrix: Squishes 3D Space to Plane

■ For full rank, the only vector that reaches the origin by transformation is the zero vector itself.

- ▶ In cases where the matrix is not of full rank, the space squishes to a lower dimension, resulting in a full line of vectors that can reach the origin.
- ▶ Case 1: 2D transformation squishes 2D space onto a line.
- ▶ Case 2: 3D transformation squishes **3D space** squishes to a **plane**.
  - ▶ There is also a full line of vectors that land on the origin.
- ▶ Case 3: 3D transformation squishes 3D space to a line.

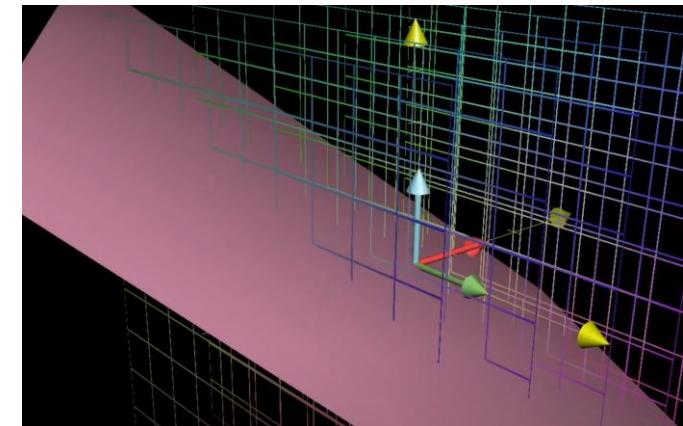
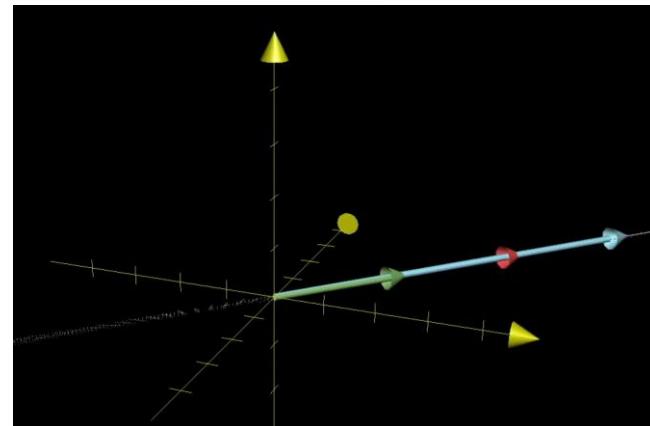
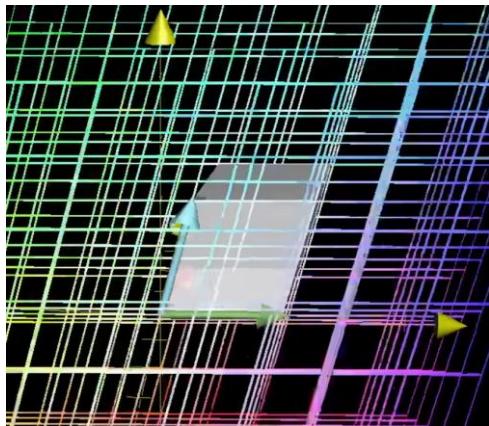


If 3D transformation squishes space to a plane, there is a full line of vectors that land on the origin

# Null Space of Matrix: Squishes 3D Space to Line

■ For full rank, the only vector that reaches the origin by transformation is the zero vector itself.

- ▶ In cases where the matrix is not of full rank, the space squishes to a lower dimension, resulting in a full line of vectors that can reach the origin.
- ▶ Case 1: 2D transformation squishes 2D space onto a line.
- ▶ Case 2: 3D transformation squishes 3D space to a plane.
- ▶ Case 3: 3D transformation squishes **3D space** to a **line**.
  - There exists a plane filled with vectors that reach the origin.



If 3D transformation squishes space to a line, there exists a plane filled with vectors that reach the origin

# Null Space of Matrix: Summary

## ■ For full rank, the only vector that reaches the origin by transformation is the zero vector itself.

- ▶ In cases where the matrix is not of full rank, the space squishes to a lower dimension, resulting in a full line of vectors that can reach the origin.
- ▶ Case 1: 2D transformation squishes 2D space onto a line.
- ▶ Case 2: 3D transformation squishes 3D space to a plane.
- ▶ Case 3: 3D transformation squishes 3D space to a line.

## ■ This set of vectors reaching the origin is called the “null space of the matrix”.

- ▶ Null space is the space of all vectors that reach a null, i.e. zero vector.
- ▶ In terms of simultaneous equations, this is the case where  $\vec{v}$  is zero vector, where zero space is all possible solutions of the equation.

$$Ax \rightarrow 0$$

# Code Exercise of Inverse Matrix

## Code Exercise (06\_04)

```
% Clear workspace, command window, and close all figures
clc; clear; close all;

% Define a square matrix
matrix = [1, 3; 2, 4];

% Define a 2D column vector
vector1 = [1; 2];

% Check if the matrix is singular (det(matrix) == 0 means it's singular)
if det(matrix) == 0
    disp('The matrix is singular and does not have an inverse.');
else
    % Calculate the inverse of the matrix
    inverseMatrix = inv(matrix);

    % Multiply the matrix by vector1 to get vector2
    vector2 = matrix * vector1;

    % Multiply vector2 by the inverse matrix to get vector3
    vector3 = inverseMatrix * vector2;

    % Print the matrices
    disp('Original Matrix:');
    disp(matrix);
    disp('Inverse Matrix:');
    disp(inverseMatrix);

    % Create the first figure for visualization of vector1 and vector2
    figure;
    hold on;
    grid on;
    axis equal;

```

```
% Visualize vector1
quiver(0, 0, vector1(1), vector1(2), 'b', 'LineWidth', 2,
'AutoScale', 'off', 'MaxHeadSize', 0.5);

% Visualize vector2
quiver(0, 0, vector2(1), vector2(2), 'r--', 'LineWidth',
2, 'AutoScale', 'off', 'MaxHeadSize', 0.5);

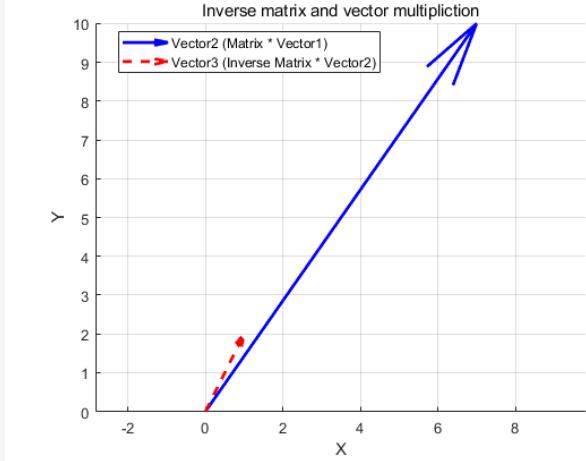
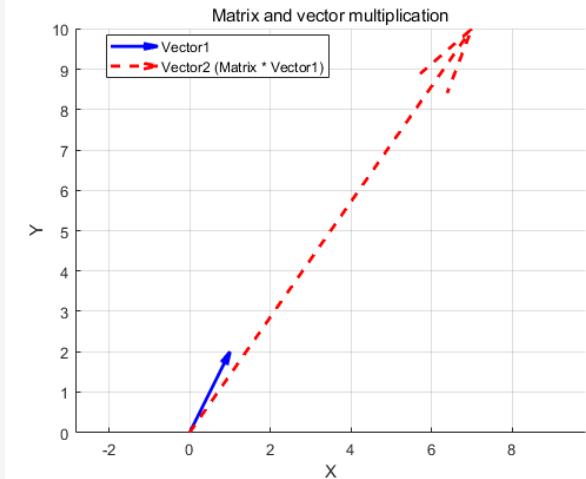
% Labels and title for the first figure
xlabel('X');
ylabel('Y');
title('Matrix and vector multiplication');
legend({'Vector1', 'Vector2 (Matrix * Vector1)'}, 'Location', 'best');

% Create the second figure for visualization of vector2 and vector3
figure;
hold on;
grid on;
axis equal;

% Visualize vector2 again for comparison
quiver(0, 0, vector2(1), vector2(2), 'b', 'LineWidth', 2,
'AutoScale', 'off', 'MaxHeadSize', 0.5);

% Visualize vector3
quiver(0, 0, vector3(1), vector3(2), 'r--', 'LineWidth',
2, 'AutoScale', 'off', 'MaxHeadSize', 0.5);

% Labels and title for the second figure
xlabel('X');
ylabel('Y');
title('Inverse matrix and vector multiplication');
legend({'Vector2 (Matrix * Vector1)', 'Vector3 (Inverse Matrix * Vector2)'}, 'Location', 'best');
end
```



MATLAB code example of Inverse matrix and results

# Rank



HANYANG UNIVERSITY



# Concept of Rank

## ■ Rank

- ▶ A **number** associated with a matrix.
- ▶ Related to **dimensionalities** of matrix subspaces.
- ▶ Has important implications for matrix operations.
  - **Inverting** matrices.
  - Determining the **number of solutions** to a system of equations.



# Properties of Rank

## ■ Rank is a **non-negative** integer.

- ▶ A matrix can have a rank of  $0, 1, 2, \dots$ , but not  $-2$  or  $3.14$ .

## ■ Every matrix has one **unique** rank.

- ▶ Matrix cannot simultaneously have multiple distinct ranks.
  - Also means that rank is a feature of the matrix, not of the rows or the columns.

## ■ Rank of a matrix is indicated: $r(A)$ or $\text{rank}(A)$ .

## ■ Maximum possible rank of a matrix

- ▶ **Smaller** of its row or column count.
- ▶ Maximum possible rank is  $\min\{M, N\}$ . 
- ▶ Maximum possible rank is called “**Full-rank**”
  - Rank  $r < \min\{M, N\}$  is variously called “**reduced-rank**,” “**rank-deficient**,” or “**singular**.”

## ■ Scalar multiplication does not affect the matrix rank.

- ▶ Exception of  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- ▶  $0$  transforms the matrix into the zero matrix with a rank of **0**.

# Several Equivalent Interpretations and Definitions of Matrix Rank

■ Largest number of columns (or rows) that form a **linearly independent set**.

■ **Dimensionality** of the column space.

- ▶ Same as the **dimensionality** of the row space.

■ Number of dimensions **containing information** in the matrix.

- ▶ Not the same as the total number of columns or rows in the matrix.
- ▶ Because of possible linear dependencies, number of nonzero singular values of the matrix.

■ Surprising that definition of rank is same for columns and rows.

- ▶ Even for non-square matrix ?

- Yes

# Example of Rank of Non-square Matrix

## ■ Column and row space of the matrix of right matrix.

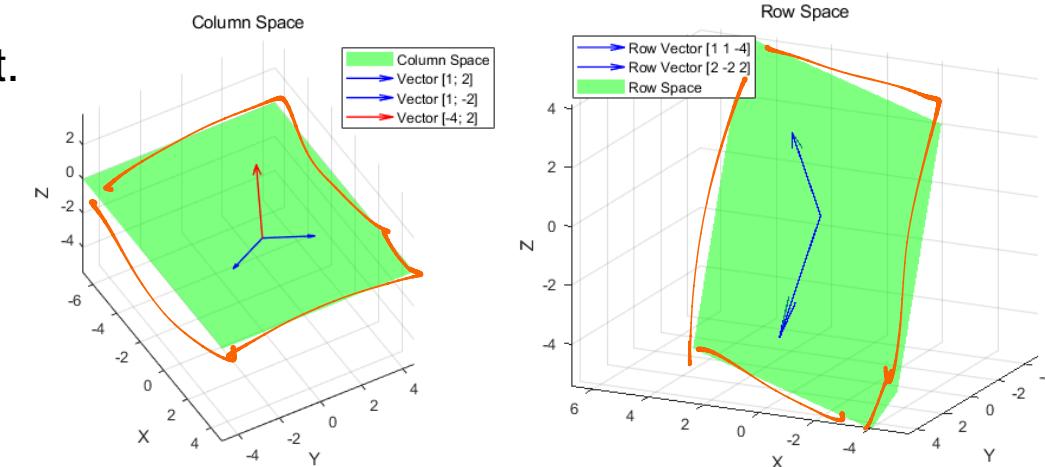
- ▶ Column space :  $\mathbb{R}^2$
- ▶ Row space :  $\mathbb{R}^3$
- ▶ Three columns do not form a linearly independent set.
  - Described as a **linear combination** of the other two.
- ▶ But they do span all  $\mathbb{R}^2$ .
  - The column space of the matrix is 2D.
  - Two rows do form a **linearly independent set**.
    - Subspace they span is a 2D plane in  $\mathbb{R}^3$ .

- ▶ Column space and row space of the matrix are different.
  - But **dimensionality** of those matrix spaces is the same.
    - Dimensionality is **rank** of the matrix.
  - Matrix has a rank of 2.

$Ax = b$

$$\begin{bmatrix} 1 & 1 & -4 \\ 2 & -2 & 2 \end{bmatrix} \text{Non-square matrix}$$

$$\begin{bmatrix} 1 & 1 & -4 \\ 2 & -2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$



Column space

Row space

# Guess Rank

## ■ Guess rank of matrix based on previous descriptions.

► We didn't learn yet, but let's try.

$$A = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 3 \\ 2 & 6 \\ 4 & 12 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 3.1 \\ 2 & 6 \\ 4 & 12 \end{bmatrix}$$

$r(A) = \boxed{1}$

$r(B) = \boxed{1}$

$r(C) = \boxed{2}$

$$D = \begin{bmatrix} 1 & 3 & 2 \\ 6 & 6 & 1 \\ 4 & 2 & 0 \end{bmatrix} \quad E = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$r(D) = \boxed{3}$

$r(E) = \boxed{1}$

$r(F) = \boxed{0}$

## ■ Focus on rank corresponds to the largest number of columns that can form a linearly independent set.

► Which also corresponds to the dimensionality of the column space of the matrix.

# Ranks of Special Matrices (I)

## ■ Vectors

- ▶ All of vectors have rank of  $|1|$
- ▶ Only exception is the zeros vector.

## ■ Zeros matrices

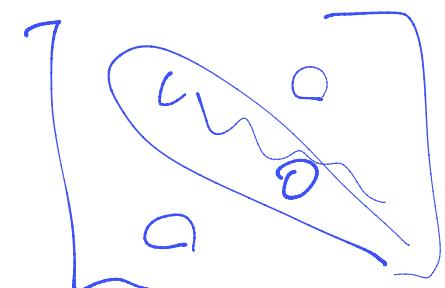
- ▶ Zeros matrix of any size (including zeros vector) has a rank of **0**.

## ■ Identity matrices

- ▶ Rank of identity matrix equals the number of rows.
  - Equal the number of columns.
- ▶  $r(I_N) = N$ .
- ▶ A special case of a diagonal matrix.

## ■ Diagonal matrices

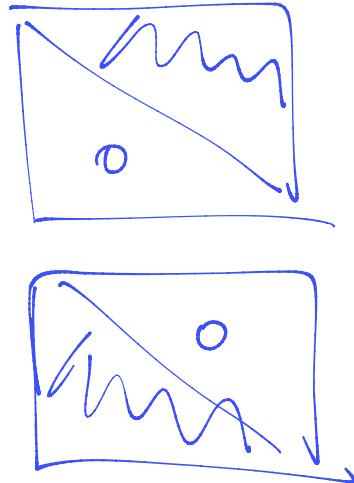
- ▶ Rank of diagonal matrix equals the number of nonzero diagonal elements.
- ▶ Each row contains maximum one nonzero element.
  - Impossible to create a nonzero number through weighted combinations of zeros.



# Ranks of Special Matrices (II)

## ■ Triangular matrices

- ▶ Full rank only if there are nonzero values in all diagonal elements.
- ▶ With at least one zero in the diagonal will be reduced rank.



## ■ Random matrices

- ▶ Impossible to know a priori.
  - Depends on **distribution** of numbers from which elements in the matrix were drawn.
  - Depends on **probability** of drawing each number.
  - Example of  $2 \times 2$  matrix populated with either 0s or 1s
    - In some case, it can have a rank of 0 if the individual elements all equal 0.
    - Another case, it can have a rank of 2 if identity matrix is randomly selected.
- ▶ Way to create random matrices with guaranteed maximum possible rank.
  - In MATLAB, you can create random matrix using randn() function

# Ranks of Special Matrices (III)

## ■ Rank – 1 matrices

- ▶ Definition: Matrix that has a **rank of 1**.
- ▶ Only one column's (or row's) worth of information in the matrix.
- ▶ All other columns (or rows) are simply linear multiples.

$$\begin{bmatrix} -2 & -4 & -4 \\ -1 & -2 & -2 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 0 & 0 \\ 2 & 1 \\ 4 & 2 \end{bmatrix}, \begin{bmatrix} 12 & 4 & 4 & 12 & 4 \\ 6 & 2 & 2 & 6 & 2 \\ 9 & 3 & 3 & 9 & 3 \end{bmatrix}$$

Example of rank – 1 matrices

- ▶ Regardless of the size, **each column (or row) is scaled copy of the first column (or row)**.
- ▶ Method to create *rank – 1* matrix
  - Taking the outer product between two nonzero vectors.
    - Third matrix above is the outer product of  $[4 \ 2 \ 3]^T$  and  $[3 \ 1 \ 1 \ 3 \ 1]$ .
- ▶ *Rank – 1* matrices are important in **eigen decomposition** and **singular value decomposition**.

# Rank of Added and Multiplied Matrices

■ If you know the ranks of matrices  $A$  and  $B$ , do you automatically know the *rank* of  $A + B$  or  $AB$ ?

▶ Answer:

▶ But here are the **rules of rank**.

- $\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B)$
- $\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}$

▶ Recommend memorizing the following rules:

- Cannot know **exact rank of a summed or product matrix**.
  - Based on knowing ranks of individual matrices (except zeros matrix).
  - Instead, individual matrices provide upper bounds for the rank of the summed or product matrix.
- Rank of a summed matrix **could be greater** than ranks of individual matrices.
- Rank of a product matrix **cannot be greater** than the largest rank of the multiplying matrices.

# Rank of Shifted Matrices

## ■ Shifted matrices have full rank.

- ▶ Goals of shifting a square matrix
  - To increase its rank from  $r < M$  to  $r = M$
- ▶ Obvious example: shifting zeros matrix by the identity matrix.
  - Rank of resulting sum  $\mathbf{0} + \mathbf{I}$  is a full rank matrix.

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

$\mathbf{0} + \mathbf{I}$  is a full rank matrix

# Example of Shifting Matrices

## ■ Leftmost matrix has a rank of 2.

- ▶ Before add  $0.1 * I$  in Leftmost matrix.
  - Third column equals the second minus the first.
- ▶ After add  $0.01 * I$  in Leftmost matrix.
  - Impossible to produce third column.
    - By some linear combination of the first two.
- ▶ Information in the matrix has **hardly changed**.
  - Pearson correlation between the elements in the original and shifted matrix:  $\rho = 0.99999997$ .
  - **It's significant implications.**
    - Ex)  $rank - 2$  matrix cannot be inverted whereas shifted matrix can.

$$\begin{bmatrix} 1 & 3 & 2 \\ 5 & 7 & 2 \\ 2 & 2 & 0 \end{bmatrix} + .01 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.01 & 3 & 2 \\ 5 & 7.01 & 2 \\ 2 & 2 & .01 \end{bmatrix}$$

Example of shifting matrix

# Code Exercise of Rank

## ■ Code Exercise (06\_05)

```
% Clear workspace, command window, and close all figures  
clc; clear; close all;  
  
% Define the 1x3 vector  
vector = [1 2 4];  
  
% Define the 2x3 matrices  
matrix1 = [1, 2, 4; 3, 6, 12];  
matrix2 = [1, 2, 4; 3.1, 6, 12];  
  
% Define the 3x3 identical (I think you meant identity) matrix and  
% 3x3 zero matrix  
identityMatrix = eye(3);  
zeroMatrix = zeros(3, 3);  
  
% Calculate the ranks  
vector_rank = rank(vector);  
matrix1_rank = rank(matrix1);  
matrix2_rank = rank(matrix2);  
identity_rank = rank(identityMatrix);  
zero_rank = rank(zeroMatrix);  
  
% Calculate and display the rank of each  
disp('Rank of the 1x3 vector: ');  
disp(vector_rank);  
disp('Rank of the first 2x3 matrix: ');  
disp(matrix1_rank);  
disp('Rank of the second 2x3 matrix: ');  
disp(matrix2_rank);  
disp('Rank of the 3x3 identity matrix: ');  
disp(identity_rank);  
disp('Rank of the 3x3 zero matrix: ');  
disp(zero_rank);
```

MATLAB code example of Null Space

# Rank Applications



HANYANG UNIVERSITY



# Augmenting Matrices

## ■ Meaning: to add extra columns to the right-hand side of the matrix.

- ▶ “**Base**”  $M \times N$  matrix and “**extra**”  $M \times K$  matrix.
  - Augmented matrix size:  $M \times (N + K)$ .
  - **Valid only two matrices have same number of rows.**
    - They can have different number of **Columns**

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

Example of augmenting matrices

## ■ Problem: whether a vector is in the column space of a matrix .

- ▶ Mathematically written as  $v \in C(A)$ .
- ▶ Use augmenting matrices to solve.

# Solve $v \in C(A)$

## Algorithm for determining whether a vector is in the column space of a matrix.

1. Augment the matrix **with the vector**.
  - Call original matrix  $A$  and the augmented matrix  $\tilde{A}$ .
2. **Compute the ranks of** the two matrices.
3. Compare the two ranks.
  - One of two possible outcomes.
  - $rank(A) = rank(\tilde{A})$ 
    - Vector  $v$  is **in the column space of matrix  $A$** .
  - $rank(A) < rank(\tilde{A})$ 
    - Vector  $v$  is **not in the column space of matrix  $A$** .

$\tilde{A}$  tilda

# Reason of Solving $v \in C(A)$

## ■ Repeat the result of solving $v \in C(A)$ .

▶  $\text{rank}(A) = \text{rank}(\tilde{A})$

- $v \in C(A)$

▶  $\text{rank}(A) < \text{rank}(\tilde{A})$

- $v \notin C(A)$

## ■ Reason of the result.

▶ If  $v \in C(A)$

- $v$  can be expressed as some linear weighted combination of the columns of  $A$ .
- Columns of augmented matrix  $\tilde{A}$  form a linearly.
- In terms of span, vector  $v$  is redundant in  $\tilde{A}$ .
  - Rank stays the same.

▶ If  $v \notin C(A)$

- $v$  cannot be expressed as some linear weighted combination of the columns of  $A$ .
- $v$  has added new information into  $\tilde{A}$ .
  - Rank will be increased by 1

# Matrix norms



HANYANG UNIVERSITY



# Matrix Norms

## ■ No ‘*the* matrix norm’.

- ▶ **Multiple** distinct norms that can be computed from a matrix.
- ▶ Somewhat similar to [vector norms](#).
  - Each matrix norm provides one number that characterizes a matrix as vector norm characterizes a vector.
  - Indicate using [double-vertical lines](#) as shown in below.

$$\|A\|$$

Norm of A matrix



# Different Meaning of Different Matrix Norms

## ■ Different matrix norms have different meanings.

- ▶ Myriad of matrix norms can be broadly divided into 2 families.
  - Element-wise (also sometimes called entry-wise)
    - Computed based on the individual elements of the matrix.
    - Can be interpreted to reflect the magnitudes of the elements in the matrix.
  - Induced
    - A measure of how much that transformation scales.
      - How much stretches or shrinks that vector.

## ■ We will study element-wise norms!

# Element-wise Norm

## ■ Euclidean norm

- ▶ A direct extension of the vector norm to matrices.
- ▶ Also called **Frobenius norm**.
  - Computed as the square root of the sum of all matrix elements squared as below Eq 1..
- ▶ In Eq 1.,
  - $i$  and  $j$  correspond to the  $M$  rows and  $N$  columns.
  - Sub-scripted  $A_F$  indicating the Frobenius norm

$$\|A\|_F = \sqrt{\sum_{i=1}^M \sum_{j=1}^N a_{ij}^2}$$

Eq 1. Frobenius norm

# Another Notation of Frobenius Norm

## ■ Frobenius norm

- ▶ Also called *l2 norm*.
- ▶ *l2 norm* gets its name from the general formula.
  - Element-wise  $p$ -norms

$$\|A\|_p = \left( \sum_{i=1}^M \sum_{j=1}^N |a_{ij}|^p \right)^{1/p}$$

General formula for element-wise  $p$ - norms

- Get Frobenius norm when  $p = 2$

# Applications of Matrix Norm: Matrix Distance

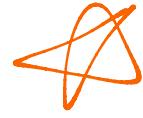
## ■ Computing measure of matrix distance.

- ▶ Distance between a matrix and itself is 0.
- ▶ How about **distance between two distinct matrices?**
  - Distance increases as the numerical values in those matrices **become increasingly dissimilar**.

## ■ Computation of Frobenius matrix

- ▶ Simply by replacing matrix  $A$  with matrix  $C = A - B$ .
  - This equation will be come out later.

# Applications of Matrix Norm: **Regularization**



## ■ Matrix norms have several application in machine learning and statistical analysis.

### ■ **Regularization** in machine learning

- ▶ Aims to improve model fitting and to increase generalization of models to unseen data

### ■ Basic idea of regularization

- ▶ Add a matrix norm as a cost function to minimization algorithm.
- ▶ Prevent model parameters from becoming too large.
  - $l_2$  regularization also called “[ridge regression](#)”.
- ▶ Prevent encouraging sparse solutions.
  - $l_1$  regularization also called “[lasso regression](#)”.

$$J = W + \dots$$

### ■ Modern deep learning architectures rely on matrix norms.

- ▶ To achieve impressive performance at solving computer vision problems.

# Code Exercise of Matrix Norm

## ■ Code Exercise (06\_06)

```
% Clear workspace, command window, and close all figures  
clc; clear; close all;  
  
% Define a 2x2 matrix  
matrix = [1, 2; 3, 4];  
  
% Calculate the norm of the matrix  
frobeniusNorm = norm(matrix, 'fro');  
L2Norm = norm(matrix(:, 2), 2);  
  
% Display the Frobenius norm  
disp('Frobenius norm of the matrix: ');  
disp(frobeniusNorm);  
disp('L2 norm of the matrix');  
disp(L2Norm);
```

MATLAB code example of Matrix Norm



# Matrix Trace

## ■ Definition of *trace* in matrix

- ▶ Sum of its diagonal elements.

## ■ Notation of *trace*

- ▶ Indicated as  $tr(A)$ .

## ■ Exists only for square matrices.

- ▶ Both of the below matrices have same trace of 14

$$\begin{bmatrix} 4 & 5 & 6 \\ 0 & 1 & 4 \\ 9 & 9 & 9 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 8 & 0 \\ 1 & 2 & 6 \end{bmatrix}$$

Both of matrices have the same trace

# Matrix Trace and Frobenius Norm

## ■ Properties of matrix trace

- ▶ Trace of a matrix equals the sum of its eigenvalues.
  - A measure of the “volume” of its eigenspace.
- ▶ Frobenius norm can be calculated with trace of matrix
  - Square root of the trace times its transpose as shown in below Eq 1..
- ▶ Why Eq 1. works?
  - Each diagonal element of the matrix  $A^T A$  is defined by dot product of each row with itself.

$$\|A\|_F = \sqrt{\sum_{i=1}^M \sum_{j=1}^N a_{ij}^2} = \sqrt{tr(A^T A)}$$

Eq 1. Calculation of Frobenius norm

# Code Exercise of Matrix Trace

## ■ Code Exercise (06\_07)

```
% Clear workspace, command window, and close all figures  
clc; clear; close all;  
  
% Define 3x3 matrices  
matrix1 = [1, 2, 3; 4, 5, 6; 7, 8, 9];  
matrix2 = [2, 3, 0; -2, 7, 4; -4, -1, 6];  
  
% Calculate the trace of the matrix  
tra1 = trace(matrix1);  
tra2 = trace(matrix2);  
  
% Display the trace of matrices  
disp('Matrix1');  
disp(matrix1);  
disp('Matrix2');  
disp(matrix2);  
disp('Trace of the matrix1: ');  
disp(tra1);  
disp('Trace of the matrix1: ');  
disp(tra2);
```

MATLAB code example of Matrix Trace

# Summary



HANYANG UNIVERSITY



# Summary

## ■ Two kinds of matrix norms

- ▶ Element-wise: reflects the magnitudes of the elements in the matrix.
  - Called the Frobenius norm (Euclidean norm or the  $l_2$  norm)
- ▶ Included: reflects the geometric-transformative effect of the matrix on vectors.

## ■ The trace of a matrix is the sum of the diagonal elements.

## ■ Four matrix spaces: column, row, null, left-null

- ▶ The set of linear weighted combinations of different features of the matrix.

## ■ The column space of the matrix

- ▶ All linear weighted combinations of the columns in the matrix, written as  $C(\mathbf{A})$ .

## ■ If some vector $b$ is in the column space of a matrix

- ▶ Some vector  $x$  such that  $\mathbf{Ax} = \mathbf{b}$ .

## ■ The row space of the matrix

- ▶ The set of linear weighted combinations of the rows of the matrix, written as  $R(\mathbf{A})$  or  $C(\mathbf{A}^T)$ .

## ■ The null space of the matrix

- ▶ The set of vectors that linearly combines the columns to produce the zeros vector.

# Summary

## ■ Rank

- ▶ Nonnegative integer associated with a matrix
- ▶ The largest number of columns (or rows) that can form a linearly independent set.
- ▶ Reduced-rank or singular
  - Matrices with a rank smaller than maximum possible
- ▶ Full-rank
  - Shifting a square matrix by adding a constant to the diagonal
- ▶ One application: determine whether a vector is in the column space of a matrix
  - Comparing the rank of the matrix to the rank of the vector-augmented matrix.

## ■ Determinant

- ▶ A number associated with a square matrix.
- ▶ Zero for all reduced-rank matrices
- ▶ Nonzero for all full-rank matrices

## ■ Characteristic polynomial

- ▶ Transform a square matrix, shifted by  $\lambda$ , into an equation that equals the determinant.

# Code exercises

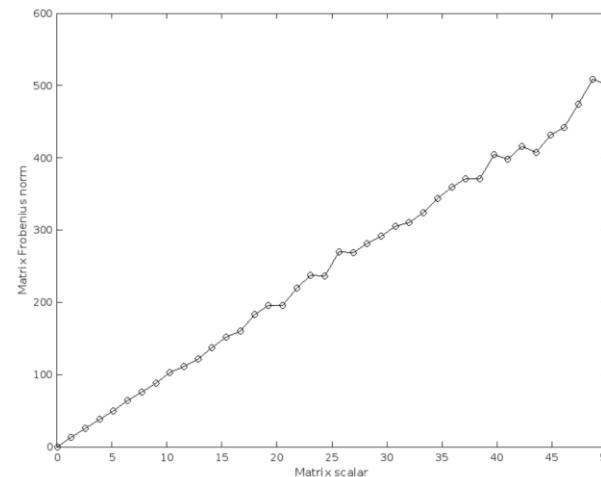


HANYANG UNIVERSITY



# Frobenius Norm Exercise

■ The norm of a matrix is related to the scale of the numerical values in the matrix. In this exercise, you will create an experiment to demonstrate this. In each of 10 experiment iterations, create a  $10 \times 10$  random numbers matrix and compute its Frobenius norm. Then repeat this experiment 40 times, each time scalar multiplying the matrix by a different scalar that ranges between 0 and 50. The result of the experiment will be a  $40 \times 10$  matrix of norms. Figure shows the resulting norms, averaged over the 10 experiment iterations. This experiment also illustrates two additional properties of matrix norms: they are strictly nonnegative and can equal 0 only for the zeros matrix.



Result of figure

# Matrix size and ranks

- I will now show you how to create random matrices with arbitrary rank (subject to the constraints about matrix sizes, etc.). To create an  $M \times N$  matrix with rank r, multiply a random  $M \times r$  matrix with an  $r \times N$  matrix.

# Matrix, Transpose Matrix Rank and Size

- Interestingly, the matrices  $A$ ,  $A^T$ ,  $A^T A$ , and  $AA^T$  all have the same rank. Write code to demonstrate this, using random matrices of various sizes, shapes (square, tall, wide), and ranks.



**THANK YOU  
FOR YOUR ATTENTION**



HANYANG UNIVERSITY

