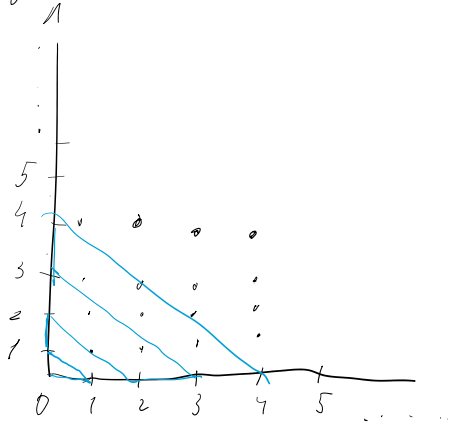
Module 2

Monday, September 11, 2023 3:41 PM

- Proof that $\sqrt{2}$ is actually a number

For a $\triangle$ with sides as 1 will have a hypothenus that's $\sqrt{2}$

- Algebraic real numbers: numbers that are the solutions to a polynomial
- Counting and Comparing Infinities
  - Cardinality: if there is 1-1 mapping between 2 sets, they are of the same size - have the same cardinality
  - The sets of all rational numbers and integers are equal:



  - If an infinite set of numbers can be lined up with natural numbers then this set is a countable infinity
    - IR Rational and Rational number sets are uncountable

In-Class Exercise 7: Show that the cardinality of $[0, 1]$ is the same as that of any $[a, b]$, for example $[0, 10]$.

(I'd could to be the same there needs to be a 1-1 mapping, so $f(x)$ - the mapping function should have a single solution for any input from $[0,1]$ that's mapped into $[a,b]$.

Let $f(x) = a + (b-a) \cdot x$ - linear func that maps the set $[0,1]$ to $[a,b]$

Let $f(x) = a + (b-a) \cdot x$ - linear func that maps the set $[0,1]$ to $[a,b]$

$\Rightarrow$ $f(x_1) = f(x_2)$ should only be the case when $x_1 = x_2$

$\Rightarrow$ $a + (b-a) \cdot x_1 = a + (b-a) x_2$

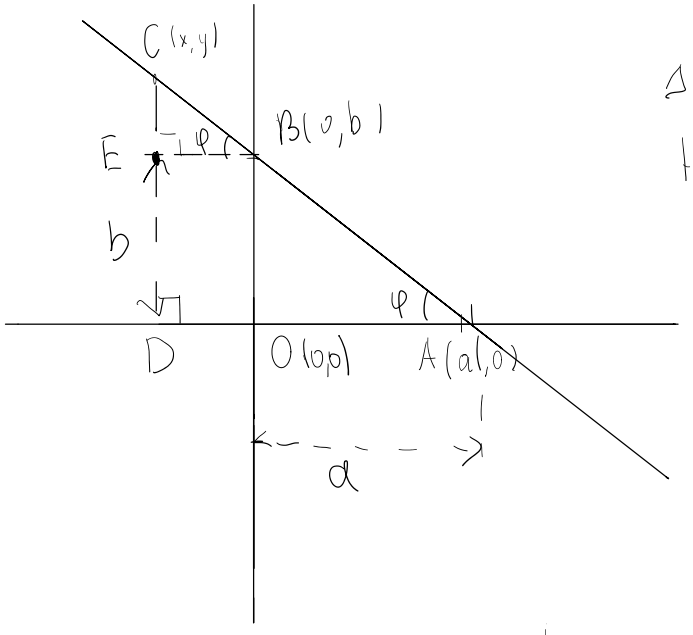
$\Rightarrow$ $(b-a) x_1 = (b-a) x_2$.

$\Rightarrow$ $x_1 = x_2$, so cond of any $[0,1]$

where $A_i \in \mathbb{R}$ is = card of $[a,b]$ where $B_i \in \mathbb{R}$

In-Class Exercise 9: Why is $ax + by + c = 0$ the equation of a line? Can you find an argument using simple geometry (similar triangles)?

A line can be represented by the location of its initial (a, b) and terminal points (x, y) on a Cartesian coordinate system.



$\triangle CEB$ and $\triangle CDA$ are similar

$\tan\varphi = \frac{CE}{EB} = \frac{CD}{DA} = \frac{CD - ED}{DA - OA} = \frac{y-b}{x} = m$

$m = \frac{y-b}{x}$

$mx = y - b$

$y = mx + b$ Proven!

Now, let $y = mx + b$

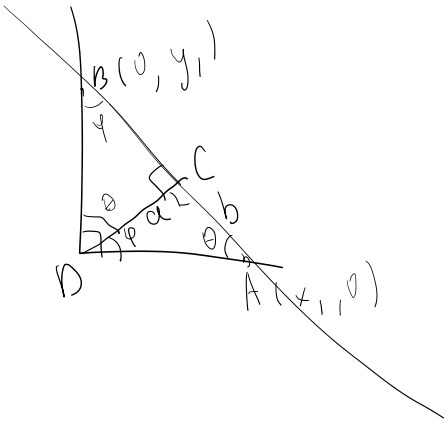
$\Rightarrow b y = b m x + b^2$

$\Rightarrow b mx - by + b^2 = 0$

$\Rightarrow b m = a \quad -b = b \quad c = b^2$

$\Rightarrow bm = a, \; -b = b, \; c = b^2$

Proof 2.



$\tan\theta = \frac{CD}{CA} = \frac{BD}{DA}$ due to similarity of the Δs

$\Rightarrow \frac{a}{b} = \frac{y}{x}$

$\Rightarrow ax = yb$

$\Rightarrow ax - by = 0$

$\Rightarrow a = a_1, \; b = b, \; c = 0$ both eq. are similar

Polynomials

- Revise Combinatorics?
- Bernstein Polynomial

Trig functions

- Any well-behaved function can be represented using the linear combination of sin and cos

$$f(t) = a_0 + \sum_{k=1}^{\infty} a_k \sin(2\pi kt/T) + \sum_{k=1}^{\infty} b_k \cos(2\pi kt/T)$$

-

$$e^z \triangleq 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \dots \text{ (analogous to the Taylor series for the real function } e^x)$$

$$\begin{aligned} e^{i\theta} &= 1 + i\theta + \frac{(i\theta)^2}{2!} + \frac{(i\theta)^3}{3!} + \dots \\ &= (\text{series for } \cos(\theta)) + i\,(\text{series for } \sin(\theta)) \\ &= \cos(\theta) + i\sin(\theta) \end{aligned}$$

- Proving Number Theory using Complex Numbers

- The cardinality of the plane is the same as the line.

In-Class Exercise 8: Why is this true?

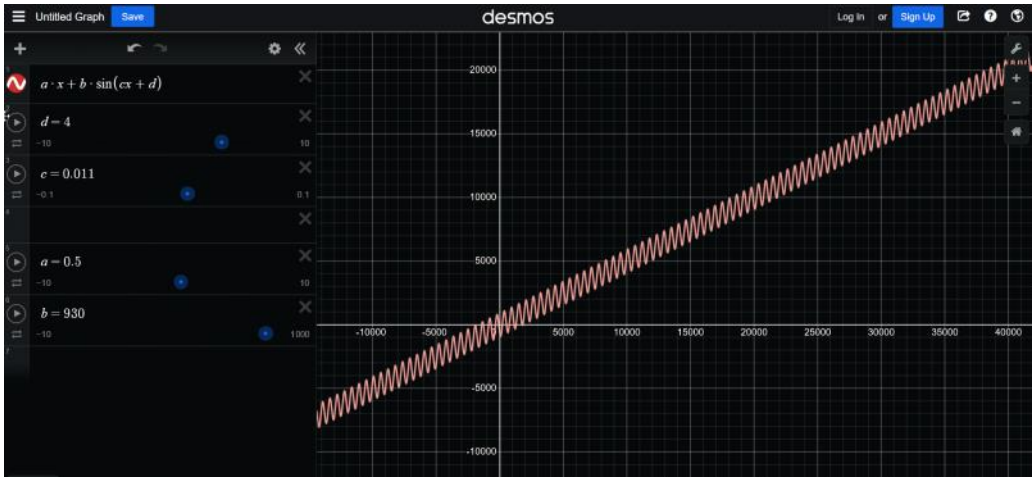
For the Card of 2 sets to be the same there needs to exist a [illegible]

For the Cardinality of 2 sets to be the same there needs to exist a function that'll perform 1-1 unique mapping from set A to set B

If we can prove the existence of such a function, then we can prove that the cardinalities of set A and B are the same. Let's take a line (called X) which will inevitably have all of its points in |R and a plane (XY) with all its points in the set |R x |R. For example, we can create a mapping with the following function:

$f(x) = a_0 x + a_1 \cdot \sin(a_2 x + a_3)$

In this case, we have a continuous function that'll map every point in X to a different point XY. Similarly, every point in XY will map to a unique point in X. Since we have established that this function f is a bijection, the cardinalities of these sets are the same



In-Class Exercise 10: Consider the two-variable function $f(x, y) = (x + y)^n$. Show that $(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$

We now that (x + y)^n is n factors of (x+y). In this representation, we know that the unique terms in the equation will be sum(i=0, n){x^{i}*y^{n-i}}. Now, the problem is to find how many times each of these terms will occur.

We know that $(x+y)^n = (x+y)(x+y)\cdots(x+y)$ n times

$\Rightarrow (x+y)^n = f(x,y) = [x^k, y^{n-k}]_{k=0}^{n}$

Unique terms in $(x+y)^n$

$\Rightarrow$ Number of ways we can choose k such that k satisfies $[x^k, y^{n-k}]_{k=0}^{n} : \binom{n}{k}$

Binom. coef provides the num of terms for each value of k from n terms

$\Rightarrow$ Sum of all terms will be $\sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$

Sum of all copies

$\sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$

summing all terms with the same power

summing all copies of all unique terms

- Replace y with $1 - x$ and define

$$b_{n,k}(x) = \binom{n}{k} x^k (1-x)^{n-k}$$

In-Class Exercise 11: Why is $b_{n,k}(x)$ a polynomial?

Polynomial is an equation of the following form: $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x_1 + a_0$

$b_{n,k}(x) = \binom{n}{k} x^k (1-x)^{n-k}$

$\Rightarrow \binom{n}{k}$ is a constant — Binomial coefficient

$\Rightarrow x^k$ is a polynomial

$\Rightarrow (1-x)^{n-k}$ is a polynomial

$\Rightarrow x^k (1-x)^{n-k}$ is a polynomial — Product of Polynomials is a polynomial.

$\Rightarrow \binom{n}{k} \cdot x^k \cdot (1-x)^{n-k}$ is a polynomial — Product of a constant and a polynomial is a polynomial

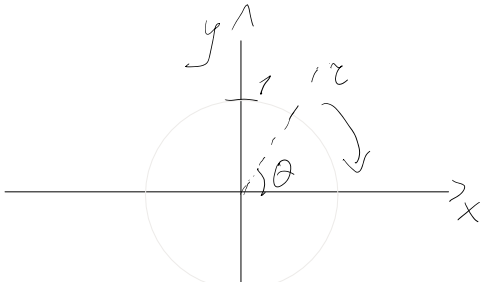
---

In-Class Exercise 12: Let's explore the last point.

- Note: 2π is approximately 6.29. Why is it OK to define a value for $\sin(17.5)$ or $\sin(-156.32)$? How are these values defined?
- Download PlotSin.java, Function.java, and SimplePlotPanel.java. Then compile and execute `PlotSin.java` to plot the function *sin* and *cos* in the range *[0,20]*. How many times does the function "repeat" in this range?

First of all, sin(θ) is a periodic function and one period is equal to 2π. This means that every ~6.29 radians, the values will repeat themselves. This means that any value of θ can be represented as multiples of 2π and the remainder.

Let θ be the angle between the radius of an arbitrarily sized circle on a Cartesian coordinate system and the positive direction of the X axis (or the Y axis value of the coordinate)



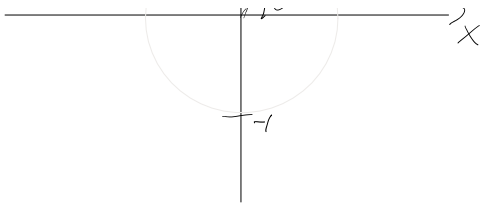
$\Rightarrow \sin(\theta) = \sin(\theta + k \cdot 2\pi)$ — Due to periodicity

$\Rightarrow \sin(\theta) = \sin(\theta + k \cdot 2\pi)$ Due to periodicity

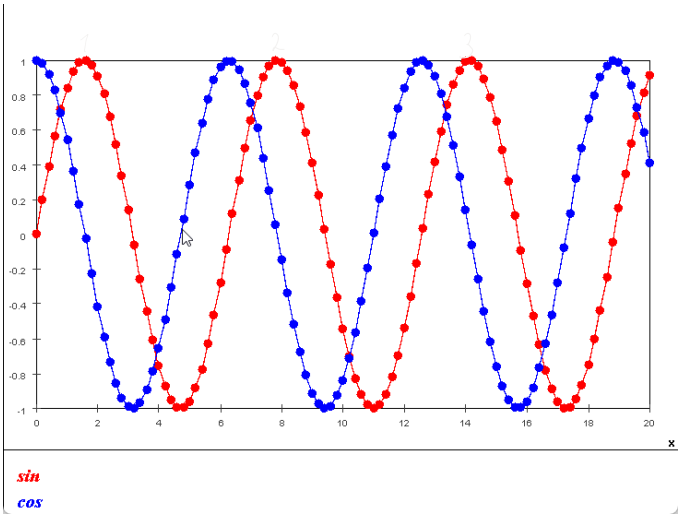
$\Rightarrow \sin(-\theta) = -\sin(\theta)$ Due to the chosen direction

$\Rightarrow -\infty < \theta < \infty$

Any value of θ is acceptable as it can be decomposed into the fixed domain of $[0, 2\pi]$ using above rules



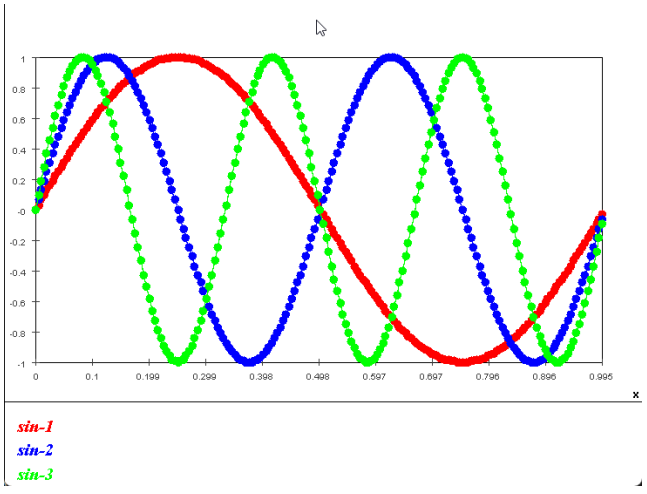
We can see that within the given range the function repeats itself 3 times



- Let's examine the function $\sin(2\pi\omega t)$.

In-Class Exercise 13: Compile and execute PlotSin2.java to plot $\sin(2\pi\omega t)$ in the range $[0, 1]$ with $\omega = 1, 2, 3$. What is the relationship between the three curves?

Despite all of the curves representing a sinusoid, their frequencies differ a lot. For example, the curve sin-3 repeats itself 3 times as frequently as the sin-1 curve. This is the concept of wavelength (or the period) and denotes a shorter wavelength with increased frequency



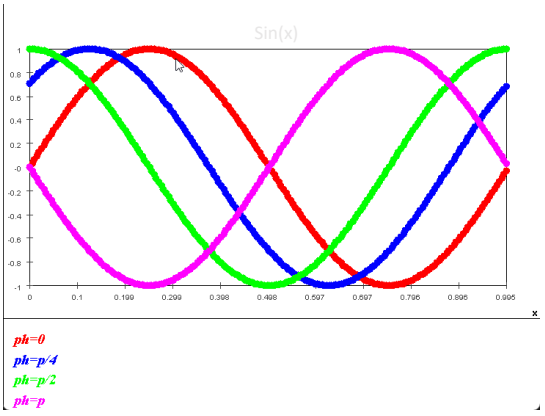
- You can shift a sin wave by adding a fixed constant to the argument: $\sin(2\pi\omega t + \phi)$

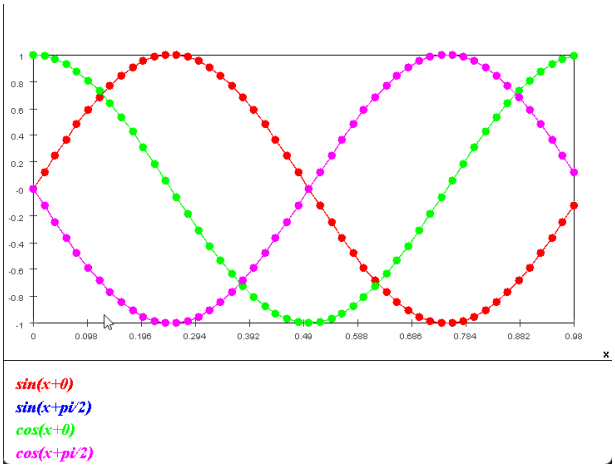
In-Class Exercise 14: Modify PlotSin3.java to plot $\sin(2\pi t + \phi)$ in the range $[0, 1]$. Plot four separate curves with $\phi = 0, \frac{\pi}{4}, \frac{\pi}{2}, \pi$. What can you say about $\sin(2\pi t + \frac{\pi}{2})$? Similarly, plot $\cos(2\pi t + \frac{\pi}{2})$ and comment on the function.

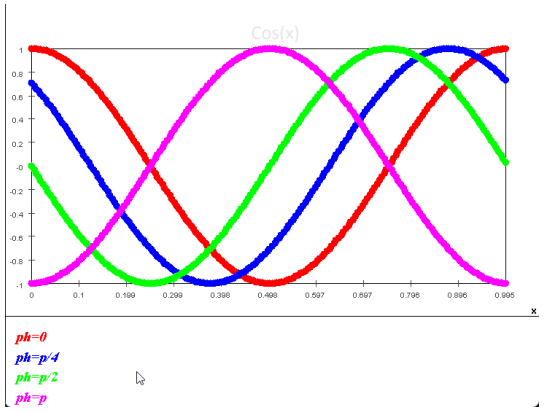
Here we can see the effects of phase shift on the sin(x) function. When it comes to the curve shifted by π/2 it is equivalent to cos of the same x. This is more evident if we look at the example of a circle from before.

$\cos(x) = \sin(x + \pi/2)$

We can confirm this by checking out the cosine curves of the same phase shift values. In the below plot, we cannot see the sin(x+pi/2) as it is overlapping with cos(x)



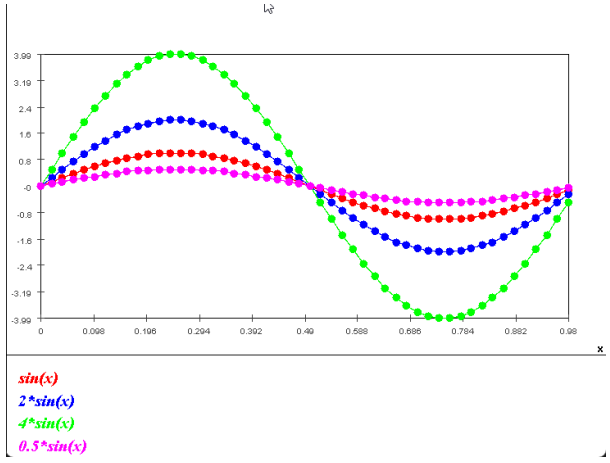




- First, note that multiplying a sinusoid by a constant, e.g., $\alpha \sin(t)$ does not change the period.

In-Class Exercise 15: Why?

If we pay attention to the form of the function, it is evident the constant alpha will only stretch the result of the function sin(t). This will only affect the final result and will have no influence on the frequency or phase as we are not modifying the angle. In the illustration to the right, we can see that no matter what the constant (the amplitude) we choose is, the frequency will stay the same and there will be no phase shift. Basically the rate at which the function repeats itself remains unchanged



Complex numbers

- Once arithmetic has been defined, we can define *functions* on complex numbers
  - ▷ e.g., for a complex number z, define $f(z) = z^2$.
- What about $\sin(z)$?

In-Class Exercise 16: Does it make sense to use a right-angled triangle to define $\sin(z)$ for a complex number z?

Complex numbers have two components: a real part and an imaginary part. It is true that complex numbers can also be represented geometrically on the complex plane, with the real part on the one axis and the imaginary part on a second axis that is perpendicular to the first. However, this geometric representation extends beyond the simple geometric concept of length and angle present in right-angled triangles. Geometry in general deals with real lengths and angles since it was created on the basis of physical analogy to the shapes present around us in the real and observable world. However, geometry is not the only way we can use to represent this function. We can also use Euler's formula.

In-Class Exercise 17: As a little practice exercise with one dimensional arrays, write some code to determine whether a string is a *double-palindrome*. Download ArrayExamples.java, which explains what a double-palindrome is, and write your code in Palindrome.java.

The code on the right is my approach to solving the problem. Here the conditions are basically checked one-y-one. I have implemented the looping logic across just the 1st half of the array since we can easily obtain the indices for both the 2nd half and the entire array easily through simple arithmetic. The results are displayed at the bottom as well. I modified the ArrayExamples.java file so that the results are more self-explanatory

Palindrome.java 1 | ArrayExamples.java 7

mod2 > Palindrome.java > Palindrome > isDoublePalindrome(char[])

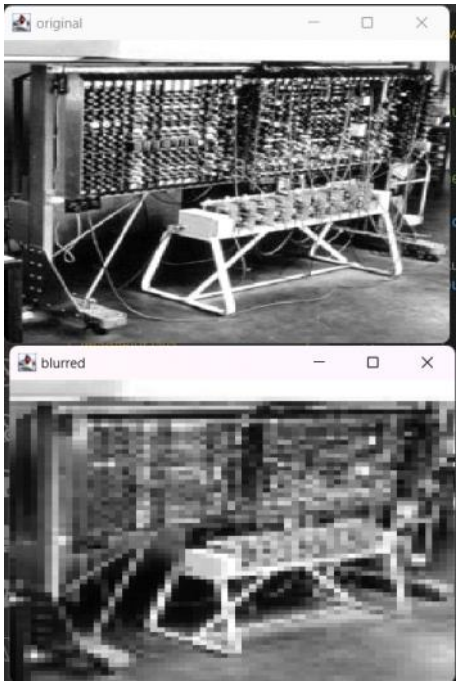
```java
public class Palindrome {

    public static boolean isDoublePalindrome (char[] digits)
    {
        // A double palindrome is defined as follows:
        // (1) The string has even length.
        // (2) The string is a palindrome.
        // (3) Each half of the string is also a palindrome.

        // check if the string has even length
        if (digits.length % 2 != 0) {
            return false;
        }

        int lenArr = digits.length;
        int halfLenArr = digits.length / 2;

        for (int i = 0; i < halfLenArr; i++) {
            /*We do not need to traverse the whole array, we can just traverse half of it
             * and mirror the indices to check from the other end of the array
             */
            // check if the string is a palindrome as a whole
            if (digits[i] != digits[lenArr - 1 - i]) {
                return false;
            }

            // check if any of the halves violate the palindrome rule
            if (digits[i] != digits[halfLenArr - 1 - i] || digits[halfLenArr + i] != digits[lenArr - 1 - i]) {
                return false;
            }
        }
        // if none of the above conditions are violated, then the string is a double palindrome
        return true;
    }
}
```

PROBLEMS 63 | DEBUG CONSOLE | PORTS | AZURE | COMMENTS | OUTPUT | TERMINAL | GL | GL

```
(base) C:\Users\togru\ADA\cla\linAlg\mod2>java ArrayExamples
>> Passed test: 3333
[INFO] isDoublePalindrome(3333) returned true
>> Passed test: 24422442
[INFO] isDoublePalindrome(24422442) returned true
>> Passed test: 123321123321
[INFO] isDoublePalindrome(123321123321) returned true
>> Passed test: 1232112321
[INFO] isDoublePalindrome(1232112321) returned true
>> Passed test: 123242321
[INFO] isDoublePalindrome(123242321) returned false
>> Passed test: 3344
[INFO] isDoublePalindrome(3344) returned false
>> Passed test: 121212
[INFO] isDoublePalindrome(121212) returned false
>> Passed test: 12344321
[INFO] isDoublePalindrome(12344321) returned false
>> Passed test: 12211212
[INFO] isDoublePalindrome(12211212) returned false
```

In-Class Exercise 18: Write code to *blur* an image. Start by writing pseudocode to solve the problem. Then, download ImageBlur.java, which contains instructions, and where you will write your code. You will also need ImageTool.java. And as a test image, you can use ace.jpg.

In the blur method, the input image is traversed in blocks of a specified size, blurSize, with non-square dimensions being carefully handled by checking the boundaries of the image. The average pixel intensity for each block is calculated, and this average value is assigned to all pixels within the block, effectively blurring the image. Finally, the newly generated blurred image, accommodating any non-square dimensions, is returned.




```java
public class ImageBlur {

    Run | Debug
    public static void main(String[] argv) {
        ImageTool imTool = new ImageTool();
        int[][] pixels = imTool.imageFileToGreyPixels(fileName:"ace.jpg");
        imTool.showImage(pixels, title:"original");
        // Each block of k x k pixels has the same color.
        int k = 4;
        int[][] blurredPixels = blur(pixels, k);
        imTool.showImage(blurredPixels, title:"blurred");
    }

    static int[][] blur(int[][] pixels, int blurSize) {
        // Get the dimensions of the image
        int height = pixels.length;
        int width = pixels[0].length;

        // Initialize a copy of the original array with the same dimensions
        int[][] blurredPixels = new int[height][width];

        // Iterate through the image in blocks of size blurSize x blurSize
        for (int i = 0; i < height; i += blurSize) {
            for (int j = 0; j < width; j += blurSize) {
                int sum = 0;
                int count = 0;

                // Calculate the average intensity in the current block
                for (int ki = 0; ki < blurSize && i + ki < height; ki++) {
                    for (int kj = 0; kj < blurSize && j + kj < width; kj++) {
                        sum += pixels[i + ki][j + kj];
                        count++;
                    }
                }
                int avg = sum / count;

                // Set the intensity of each pixel in the current block to the average
                for (int ki = 0; ki < blurSize && i + ki < height; ki++) {
                    for (int kj = 0; kj < blurSize && j + kj < width; kj++) {
                        blurredPixels[i + ki][j + kj] = avg;
                    }
                }
            }
        }

        // Return the blurred image
        return blurredPixels;
    }
}
```

In-Class Exercise 19: Let's examine the idea of computing the "distance" between two arrays.

- A distance measure should take two arrays and return a non-negative number.
- Propose a "distance" calculation for 1D arrays.
- Propose an extension of that idea to 2D arrays.

Do the arrays have to be of the same length?

To do this, we can assign a geometric meaning to arrays. This way, we can calculate the distance between the two arrays as the sum of the distances between each corresponding point. This can be done by aligning the elements of the arrays on a 1D (line) axis. To prevent the negative numbers from distorting the actual concept of distance we can use either the Manhattan or Euclidian distance measure.

$A = [a_1, a_2, a_3, \dots]_n \qquad B = [b_1, b_2, b_3, \dots]_m$

Euclidian: $D_E(A,B) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$

Manhattan: $D_M(A,B) = \sum_{i=1}^{n} |a_i - b_i|$

We can extend this to 2D simply by treating each column as a 1D vector and summing the distances between those vectors. $A, B \in \mathbb{R}^{m \times n}$

$$D_E(A,B) = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} (a_{ij} - b_{ij})^2}$$

Ideally, to compute the Euclidean distance, the arrays should be of the same length/dimensions, as the formula involves pairwise differences between corresponding elements of the arrays. However, in practice, there might be cases where arrays of different lengths need to be compared. In such cases, we can use padding. Shorter array can be padded with zeros to match the length/dimension of the longer array.

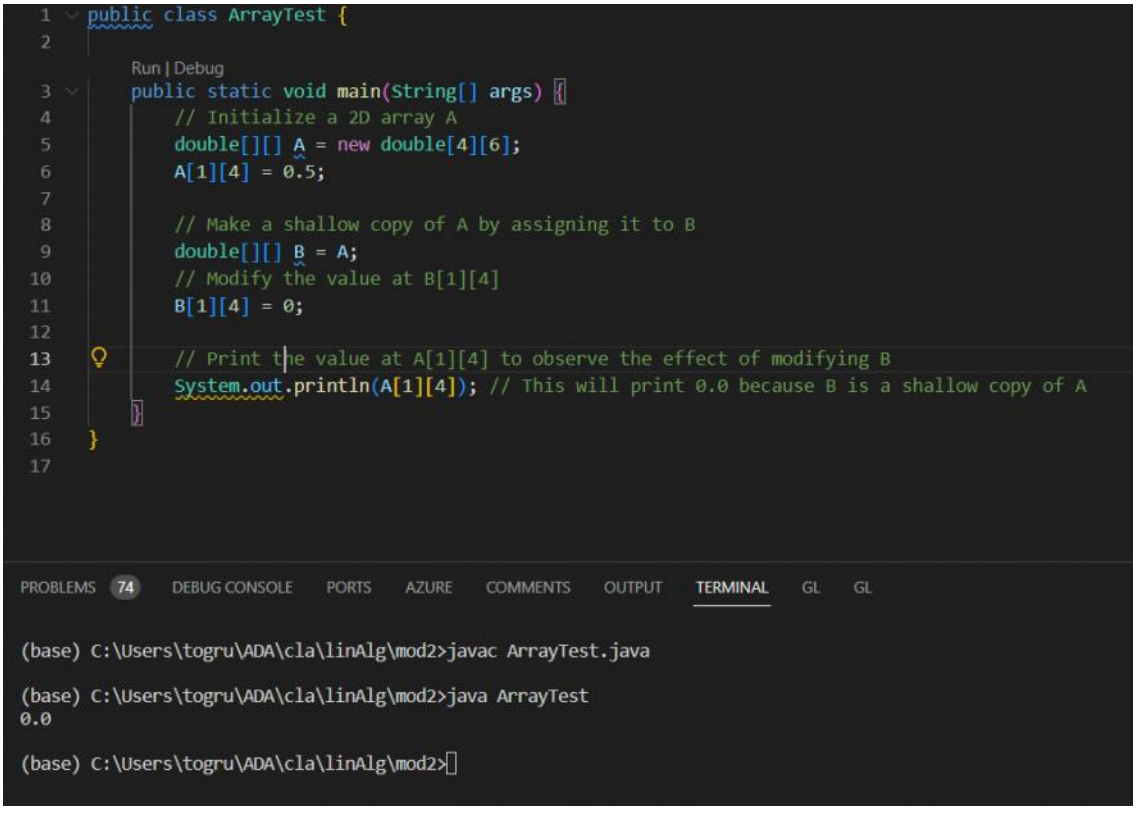
In-Class Exercise 20: How is a 2D array stored in memory? Consider this code snippet:

```
double[][] A = new double [4][6];
A[1][4] = 0.5;
double[][] B = A;
B[1][4] = 0;
System.out.println (A[1][4]);       // What gets printed out?
```

What does it mean to make a "deep copy" of an array?

In computer memory, a 2D array is stored as an array of arrays, often referred to as a "jagged array" because it allows each row to have a different number of columns, though in languages like Java, 2D arrays are typically rectangular.
When you declare a 2D array double[][] A = new double[4][6];, it's somewhat like creating 4 separate arrays (each of length 6) and then another array holding references to these 4 arrays. And B is not a deep copy of A. When the statement double[][] B = A; is executed, B becomes a reference to the same object in memory that A is pointing to which means it is a shallow copy

Making a "**deep copy**" of an array means creating a new array object and then copying each element from the original array to the new one



Module 3

Sunday, September 17, 2023 7:39 PM

In-Class Exercise 1: Pick three other applications and propose natural groupings of numbers for those applications.

1. Weather Forecasting

- Atmospheric Pressure Data: Readings at various altitudes
- Temperature Data: Readings at different times and locations
- Humidity Data: Humidity levels at different times and places
- Wind Speeds: Speed and direction at different altitudes

2. Natural Language Processing (NLP)

- Word Embeddings: Vectors that represent the semantic meaning of words.
- Document Vectors: Average or weighted sum of word embeddings in a document.
- TF-IDF Scores: Term Frequency-Inverse Document Frequency scores for words in a corpus

3. Time Series Analysis

- Historical Data: Past observations of a variable over time.
- Frequency Components: Fourier coefficients representing cyclical behavior.
- Predicted Values: Forecasted observations based on the model.

In-Class Exercise 3: Why is it true that the orientation (angle), except for flips, is unchanged by scalar multiplication?

Given a vector A and a scalar α, the scalar multiple αA can be represented as:

$$\alpha A = (\alpha x, \alpha y)$$

The orientation or angle θ of the vector A with respect to the x-axis is given by:

$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

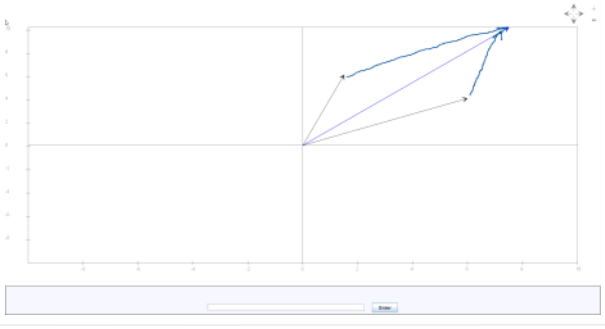
angle of A wrt x axis

$$\Rightarrow \theta = \tan^{-1}\left(\frac{\alpha y}{\alpha x}\right)$$

after scalar Mult.

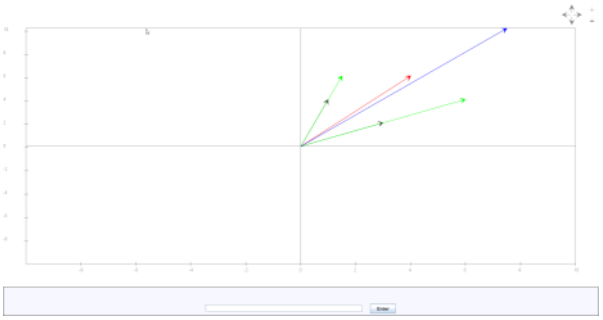
$$\Rightarrow \theta = \tan^{-1}\left(\frac{y}{x}\right) = \tan^{-1}\left(\frac{\alpha y}{\alpha x}\right)$$

In-Class Exercise 4: Download LinCombExample.java and DrawTool.java. Then, examine the code in LinCombExample to see how vectors and arrows are drawn. Then, draw the remaining arrows to complete the parallelogram.

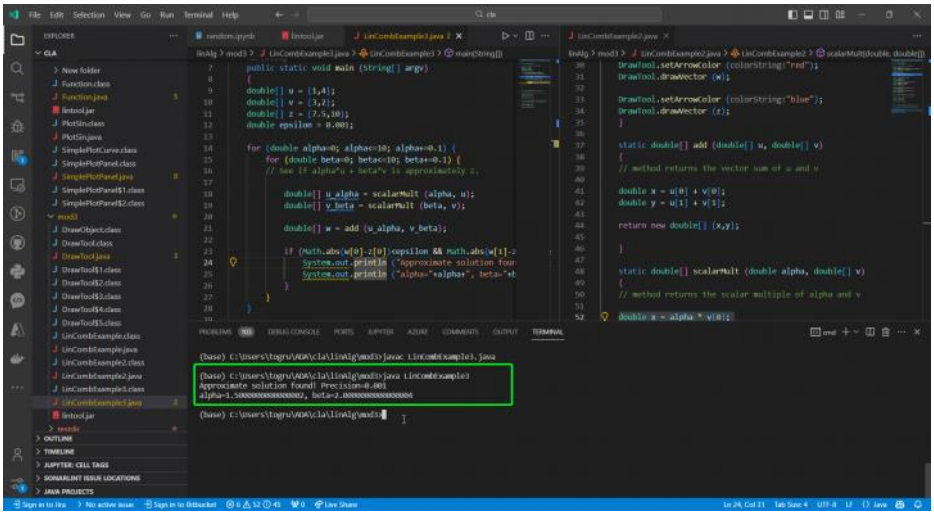


In-Class Exercise 5: Download LinCombExample2.java (you already have DrawTool). Write code to implement vector addition and scalar multiplication, and test with the example in main().

Here, red arrow indicates (u + v) while the green arrows indicate alpha * u and beta * v



In-Class Exercise 6: Download LinCombExample3.java. The double-for loop tries to systematically search over possible values of α,β to see if some combination will work. Write code to see if the linear combination αu+βv is approximately equal to z.



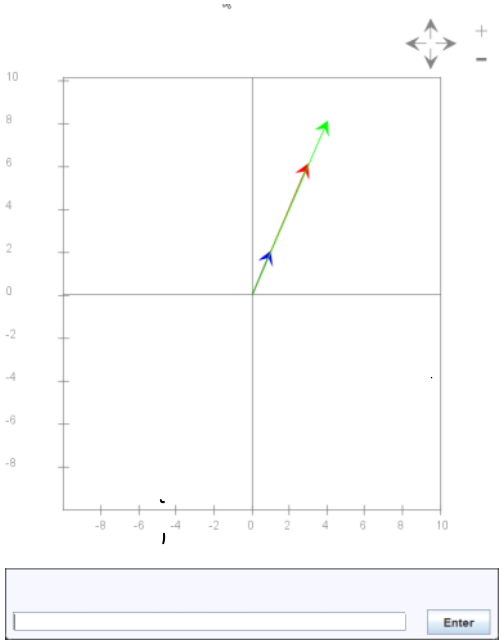
In-Class Exercise 9: Now consider **u**=(1,2),**v**=(3,6)and **z**=(4,8). Write code in LinCombExample5.java to draw these. Is there a unique solution? Explain both geometrically and algebraically.

As we can see below, all of the vectors are on the same line and are linearly dependent on each other.
Geometrically, the vectors u and v are collinear -> they lie along the same line. This means that their linear combinations will span only a one-dimensional subspace (a line in this case). The vector z also lies along this line. Therefore, there are infinitely many ways to represent z as a linear combination of u and v.

$$\begin{array}{c} \\ R_1 \\ R_2 \end{array} \begin{array}{c} \begin{array}{cc} \alpha & \beta \end{array} \\ \left[\begin{array}{cc|c} 1 & 3 & 4 \\ 2 & 6 & 8 \end{array}\right] \end{array}$$

$2R_1 - R_2 \Rightarrow \boxed{0 = 0}$ tautology

This means we have infinitely many solutions



In-Class Exercise 2: Explain the following:

- Go back to the example with two coordinate systems, and verify by hand that addition produces the same result, relative to the new origin.
- What would be the case if a second set of coordinate axes had the same origin but was rotated slightly (say, by 60 degrees)?
- By the rules above, we aren't allowed to change the orientation of an arrow when it's moved for addition. Why is that? What would go wrong with the theory if we allowed the direction to change?

1. New Coordinate System at [1, 2]
   a. Vector 1: [7 3]
   b. Vector 2: [2 6]
   c. Sum: [9 9]
   d. Origin of new coordinate system: [1 2]
   e. Vector 1 in new coordinate system: [6 1]
   f. Vector 2 in new coordinate system: [1 4]
   g. Sum in new coordinate system: [8 7]
2. What would be the case if a second set of coordinate axes had the same origin but was rotated slightly (say, by 60 degrees)?
   a. the vectors themselves would not change, as vectors are geometric entities independent of any coordinate system. However, the coordinates of the vectors in this new rotated coordinate system would differ from those in the original system
   b. Let's say you have two vectors AA and BB in the original coordinate system, and A'A' and B'B' represent the coordinates of these vectors in the rotated system.
   In general, you can transform coordinates from one system to another using a rotation matrix RR. If θ is the angle of rotation, then the 2x2 rotation matrix R for a counter-clockwise rotation in 2D is given by:

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

   Thus, the new coordinates would be:

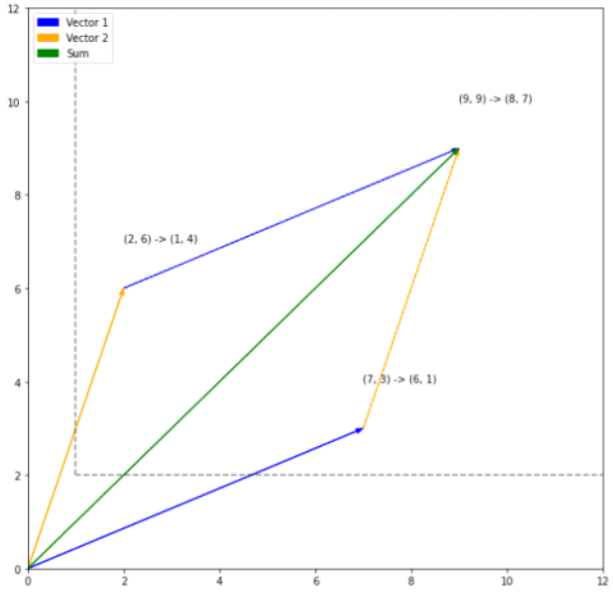
$$A' = RA$$

$$B' = RB$$

   a. Interpretation:

      i. Direction: The vectors A and B will appear rotated by 60 degrees in this new system.
      ii. Magnitude: The lengths of the vectors will remain unchanged because a rotation doesn't affect vector magnitudes.
      iii. Vector Addition: The vector C=A+B will also rotate by the same angle when represented in the new coordinate system, and its coordinates would be C'=RC

1. By the rules above, we aren't allowed to change the orientation of an arrow when it's moved for addition. Why is that? What would go wrong with the theory if we allowed the direction to change?
   a. Since a vector also holds the concept of direction in addition to the concept of magnitude, changing its orientation would result in the terminal points of the vector being displaced, hence, creating a totally different vector
   b. Commutative and Associative Laws: In vector arithmetic, A+B=B+A and (A+B)+C=A+(B+C). Changing the orientation of a vector when adding would violate these laws, undermining the



mathematical structure we rely on

c. Also, when a vector is translated it is always linearly dependent on the original vector as the equation of the line on which it lies needs to transformed solely linearly

---

In-Class Exercise 7: Solve the above problem by hand: are there values of α, β such that $\alpha\mathbf{u} + \beta\mathbf{v} = \mathbf{z}$ when $\mathbf{u} = (1, 4)$, $\mathbf{v} = (3, 2)$ and $\mathbf{z} = (7.5, 10)$?

$x: \alpha \cdot 1 + \beta \cdot 3 = 7.5$

$y: \alpha \cdot 4 + \beta \cdot 2 = 10$

$\Rightarrow \left[\begin{array}{cc|c} \alpha & \beta & \\ 1 & 3 & 7.5 \\ 4 & 2 & 10 \end{array}\right] \begin{array}{l} \\ R_1 \\ R_2 \end{array}$

$4R_1 - R_2 \Rightarrow 10\beta = 20$

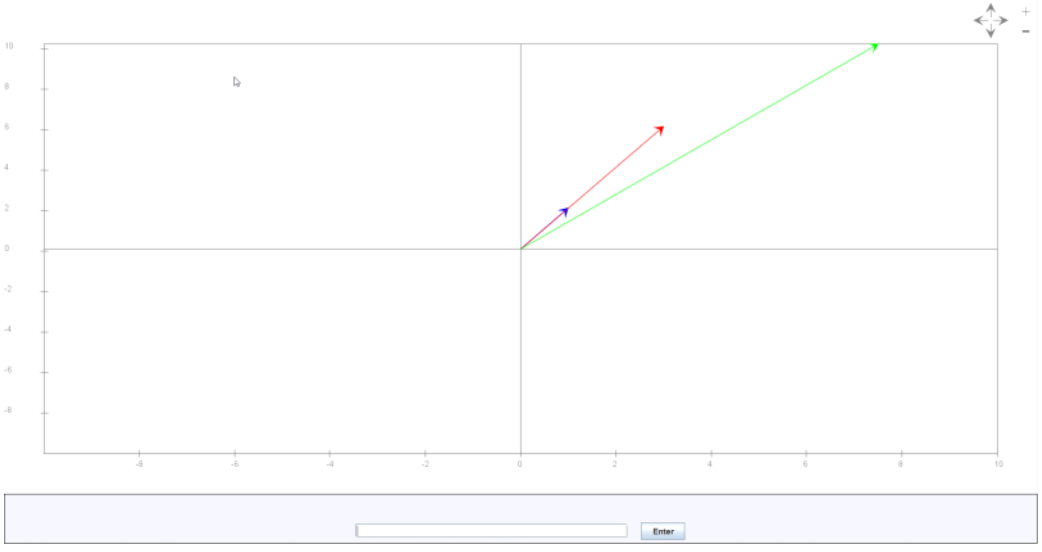
$\beta = 2$

$\alpha = 1.5$

In-Class Exercise 8: Suppose $\mathbf{u} = (1, 2)$, $\mathbf{v} = (3, 6)$ and $\mathbf{z} = (7.5, 10)$. Download LinCombExample4.java and draw all three vectors. Then:

- Use the drawing to explain why no linear combination of $\mathbf{u}$ and $\mathbf{v}$ will work.
- Solve by hand to confirm the explanation algebraically.

From the below image we can see that the vectors u and v (in blue and red respectively) simply linear transformations of each other. Since they are linearly dependent they are on the same line an can be defined by the same equation of the line. Thus, it is impossible to get to any other point on the plane XY aside from the points that fall onto the same line as u and v.



$v = 3 \cdot u$

$\Rightarrow$ we can use u for the comparison

[illegible] to be a sol. the following systems of eq. needs to have a sol.

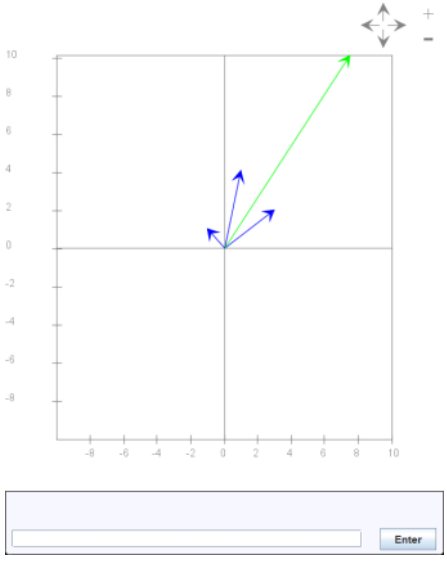
For there to be a sol. the following systems of eq. needs to have a sol.

$$\begin{matrix} R_1 \\ R_2 \end{bmatrix} \left[\begin{array}{cc|c} 1 & 3 & 7.5 \\ 2 & 6 & 10 \end{array}\right]$$

$R_2 - 2R_1 \Rightarrow \quad 0 = -5 \rightarrow$ contradiction, so no solution.

In-Class Exercise 10: Finally, consider $\mathbf{u} = (1, 4), \mathbf{v} = (3, 2), \mathbf{w} = (-1, 1)$ and $\mathbf{z} = (7.5, 10)$. We will ask whether there is a linear combination $\alpha\mathbf{u} + \beta\mathbf{v} + \gamma\mathbf{w}$ such that $\alpha\mathbf{u} + \beta\mathbf{v} + \gamma\mathbf{w} = \mathbf{z}$. Write out the equations and explain algebraically whether there is a solution. The code in LinCombExample6.java draws all four vectors. Can a linear combination of any two of $\mathbf{u}, \mathbf{v}, \mathbf{w}$ suffice to produce $\mathbf{z}$?

Hypothesis: Considering that all 3 vectors are on 2D, and that they do not appear to have collinearity, there should be such a combination of 2 vectors that we can obtain the vector z. This is a logical conclusion from the fact that 2 equations (thus, 2 vectors) should generally be enough to prove existence or impossibility of a solution.



$$\begin{matrix} & \alpha & \beta & \gamma \\ R_1 \\ R_2 \end{matrix} \left[\begin{array}{ccc|c} 1 & 3 & -1 & 7.5 \\ 4 & 2 & 1 & 10 \end{array}\right]$$

$$R_1 + R_2 \Rightarrow \left[\begin{array}{cc|c} 5 & 5 & 17.5 \\ 4 & 2 & 10 \end{array}\right]$$

$$\Rightarrow \left[\begin{array}{cc|c} 1 & 1 & 3.5 \\ 2 & 1 & 5 \end{array}\right]$$

$R_2 - R_1 \Rightarrow \alpha = 1.5$

$\beta = 2$

$\gamma = 0$

in fact, it's actually possible to get vect. z through comb of just u and v

$\gamma = 0$

in fact, it's actually possible to get vect. z through comb of just u and v

In-Class Exercise 11: The code in LinComb3DExample.java displays the vectors in the above example. (You need to have the Draw3D jar in your CLASSPATH). Write down the equations for α, β, γ and solve by hand.

$$\alpha(1,3,1) + \beta(4,1,0) + \gamma(3,2,6) = (1,7,8)$$

$$\begin{matrix} R_1 \\ R_2 \\ R_3 \end{matrix}\left[\begin{array}{ccc|c} 1 & 4 & 3 & 1 \\ 3 & 1 & 2 & 7 \\ 1 & 0 & 6 & 8 \end{array}\right]$$

$$R_1 = 2R_1 - R_3 \Rightarrow \begin{matrix} & \alpha & \beta & \gamma \\ \end{matrix}\left[\begin{array}{ccc|c} 1 & 8 & 0 & -6 \\ 3 & 1 & 2 & 7 \\ 1 & 0 & 6 & 8 \end{array}\right]$$

$$R_2 = 3R_2 - R_3 = \left[\begin{array}{ccc|c} 1 & 8 & 0 & -6 \\ 8 & 3 & 0 & 13 \\ 1 & 0 & 6 & 8 \end{array}\right]$$

$$\begin{matrix} R_1 \\ R_2 \end{matrix}\left[\begin{array}{cc|c} 1 & 8 & -6 \\ 8 & 3 & 13 \end{array}\right]$$

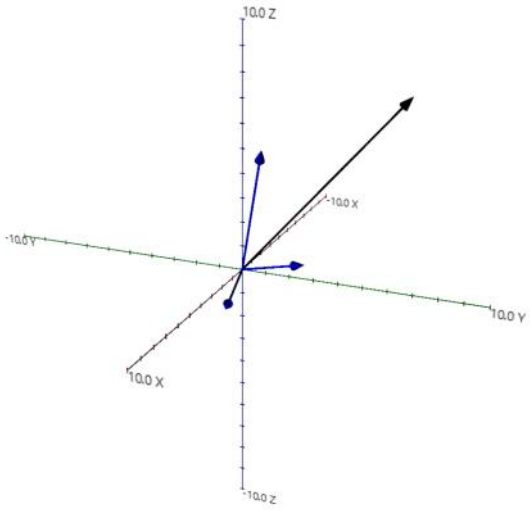
$$R_1 = 8R_1 - R_2 \Rightarrow \begin{matrix} \alpha & \beta \\ \end{matrix}\left[\begin{array}{cc|c} 0 & 61 & -61 \\ 8 & 3 & 13 \end{array}\right]$$

$\beta = -1$

$8\alpha - 3 = 13 \Rightarrow \alpha = 2$

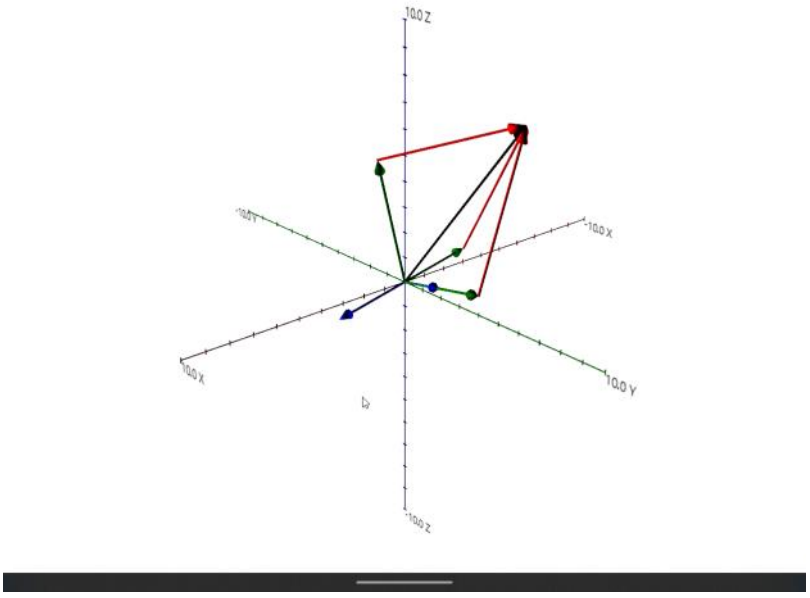
$1 \cdot \alpha + 6\gamma = 8$
$2 + 6\gamma = 8 \Rightarrow \gamma = 1$

$\alpha = 2, \beta = -1, \gamma = 1$



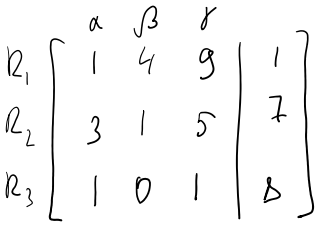
In-Class Exercise 12: Add code to LinComb3DExample2.java to display the 3D version of the 2D parallelogram. That is, draw arrows that show the stretched vectors when scaling by α, β, γ respectively. Then draw arrows from the tips of the stretched vectors to the resultant vector **z**.

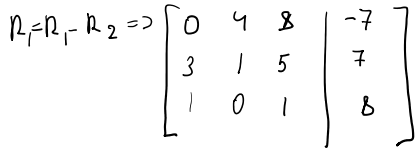
```java
// The three vectors u,v,w:

d3.setDrawColor (Color.BLUE);
d3.drawVector (1,3,1);
d3.drawVector (4,1,0);
d3.drawVector (3,2,6);

// Example of using drawArrow(). The vector z:
d3.setDrawColor (Color.BLACK);
d3.drawArrow (0,0,0, 1,7,8);

// Stretched vectors:
double alpha = 2;
double beta = -1;
double gamma = 1;

// Use drawArrow to draw the added stretch.
d3.setDrawColor (Color.GREEN);
d3.drawArrow (0, 0, 0, alpha*1, alpha*3, alpha*1);
d3.drawArrow (0, 0, 0, beta*4, beta*1, beta*0);
d3.drawArrow (0, 0, 0, gamma*3, gamma*2, gamma*6);

// Use drawArrow to draw arrows from the tips of
// the stretched vectors to the final vector z=(1,7,8).
d3.setDrawColor (Color.RED);
d3.drawArrow (alpha*1, alpha*3, alpha*1, 1,7,8);
d3.drawArrow (beta*4, beta*1, beta*0, 1,7,8);
d3.drawArrow (gamma*3, gamma*2, gamma*6, 1,7,8);

}
```

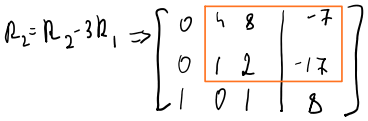


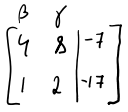
In-Class Exercise 13: Suppose now that $\mathbf{u} = (1, 3, 1), \mathbf{v} = (4, 1, 0), \mathbf{w} = (9, 5, 1)$ and $\mathbf{z} = (1, 7, 8)$. We seek α, β, γ such that $\alpha\mathbf{u} + \beta\mathbf{v} + \gamma\mathbf{w} = \mathbf{z}$. Write down the equations for α, β, γ and solve by hand. The code in LinComb3DExample3.java displays the four vectors. Explain geometrically why this is consistent with your solution.

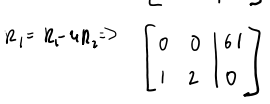
Keeping in mind that scalar multiplication serves solely to stretch a vector, since it is not possible to change the angle of the vector. We can somehow see that these three vectors may be on a single plane. We'll test this later



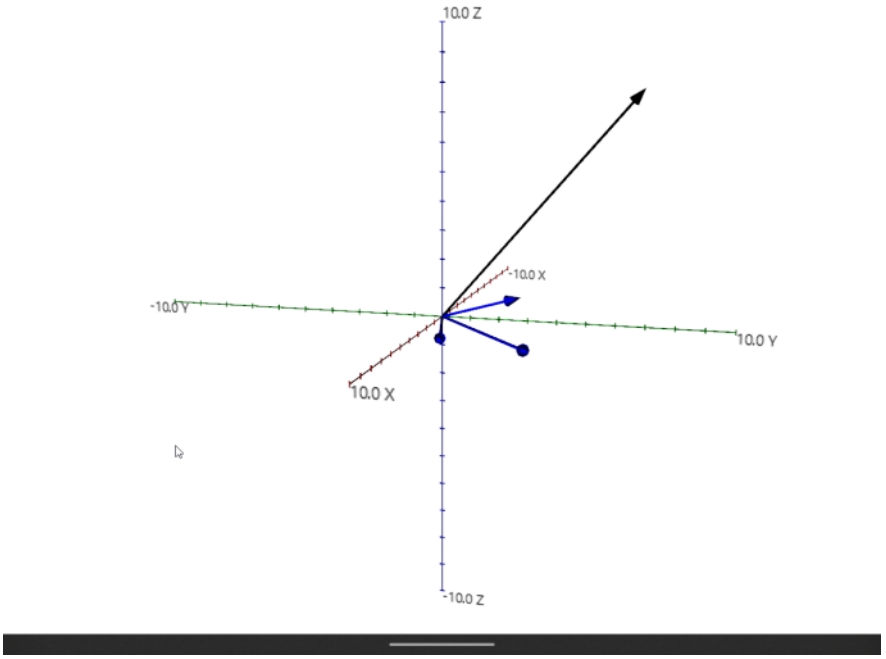








$0 \cdot \beta + 0 \cdot \gamma = 61 \rightarrow$ No Solution



Let's see if these three vectors are on the same plane. To do this, we can check the determinant of the transformation matrix created by the 3 vectors. The determinant of a matrix formed by three vectors as its columns results in a scalar value that indicates the volume of the parallelepiped they can form in 3D. If the determinant is zero, the volume of the parallelepiped is zero, which means that the vectors are linearly dependent and lie on the same plane. Whilst a non-zero determinant means that the vectors

are linearly independent and span a three-dimensional space, not confined to a single plane.
In conclusion, if the Determinant will be 0, then we will have proved that it is indeed geometrically impossible to construct vector z out of the 3 vectors

$$M = \begin{bmatrix} 1 & 4 & 9 \\ 3 & 1 & 5 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \text{Det}(M) &= 1\cdot(1\cdot 1 - 0\cdot 5) - \\ &\quad 4\cdot(3\cdot 1 - 5\cdot 1) + \\ &\quad 9\cdot(3\cdot 0 - 1\cdot 1) = 1 + 8 - 9 = 0 \end{aligned}$$

In-Class Exercise 14: Apply the matrix $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ to the vector $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. What are the dimensions (number of rows, number of columns) of the resulting product?

$$A_{2\times 2} \times X_{2\times 1} = M_{2\times 1}$$

$$A \times X = \begin{bmatrix} a_{11}\cdot x_1 + a_{12}\cdot x_2 \\ a_{21}\cdot x_1 + a_{22}\cdot x_2 \end{bmatrix}$$

In-Class Exercise 15: Apply the matrix $\mathbf{A} = \begin{bmatrix} 2 & -3 \\ 0 & 1 \end{bmatrix}$ to the vector $\mathbf{x} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ to get the vector $\mathbf{y}$. Then apply $\mathbf{B} = \begin{bmatrix} 1 & 2 \\ 0 & -3 \end{bmatrix}$ to $\mathbf{y}$ to get $\mathbf{z}$. What are $\mathbf{y}$ and $\mathbf{z}$? Show your calculations.

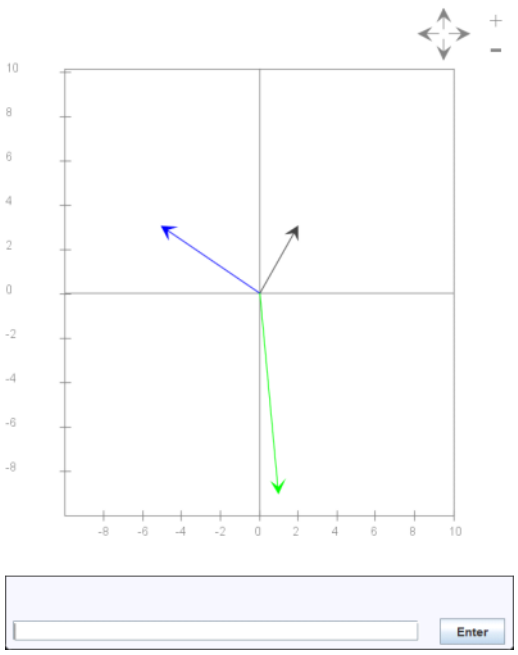
$$\begin{bmatrix} 2 & -3 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -5 \\ 3 \end{bmatrix} = y$$

$$B \times y = \begin{bmatrix} 1 & 2 \\ 0 & -3 \end{bmatrix} \times \begin{bmatrix} -5 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ -9 \end{bmatrix} = z$$

In-Class Exercise 16: Download MatrixVectorExample.java and write code to perform the matrix-vector product. Then confirm your calculations above from the displayed vectors.

```java
DrawTool.drawMiddleAxes (drawThem:true);
DrawTool.drawVector (x);
DrawTool.setArrowColor (colorString:"blue");
DrawTool.drawVector (y);
DrawTool.setArrowColor (colorString:"green");
DrawTool.drawVector (z);
}


static double[] matrixVectorMult (double[][] A, double[] v)
{
    // Get the number of rows of matrix A
    int rowsA = A.length;

    // Check if the matrix A and vector v can be multiplied
    if (A[0].length != v.length) {
        throw new IllegalArgumentException(s:"Number of columns of A must be equal to the number of rows of v");
    }

    // Compute the product of A and v and return the result.
    double[] result = new double[rowsA];
    for (int i = 0; i < rowsA; i++) {
        result[i] = 0;
        for (int j = 0; j < v.length; j++) {
            result[i] += A[i][j] * v[j];
        }
    }

    return result;
}
```

```
(base) C:\Users\togru\ADA\cla\linAlg\mod3>javac --module-path %PATH_TO_FX% --add-modules javafx.controls MatrixVectorExample.java

(base) C:\Users\togru\ADA\cla\linAlg\mod3>java --module-path %PATH_TO_FX% --add-modules javafx.controls MatrixVectorExample
Vector: -5.000  3.000
Vector:  1.000 -9.000

(base) C:\Users\togru\ADA\cla\linAlg\mod3>
```



In-Class Exercise 17: Compute the vector obtained by multiplying $\mathbf{x}$ above by the matrix $\mathbf{C} = \begin{bmatrix} 2 & -1 \\ 0 & -3 \end{bmatrix}$. Compare with the value of $\mathbf{z}$ calculated earlier.

$$C \times x = \begin{bmatrix} 2 & -1 \\ 0 & -3 \end{bmatrix} \times \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ -9 \end{bmatrix} = z$$

In-Class Exercise 18: Compute the vector obtained by multiplying $\mathbf{z}$ above by the matrix $\mathbf{C}_2 = \begin{bmatrix} 1.0/2 & -1.0/6 \\ 0 & -1.0/3 \end{bmatrix}$. Compare with the value of $\mathbf{x}$ calculated earlier.

$$C_2 \times z = \begin{bmatrix} 1/2 & -1/6 \\ 0 & -1/3 \end{bmatrix} \times \begin{bmatrix} 1 \\ -9 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} = x$$

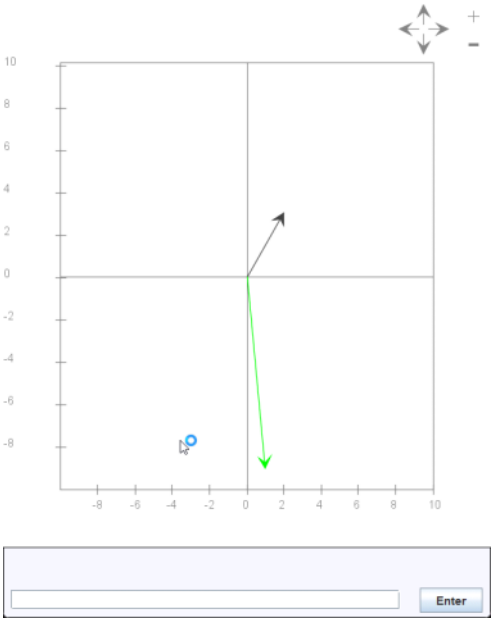
In-Class Exercise 19: Earlier, you computed the product of $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ with the vector $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, which produces a result vector. Multiply this result vector by $\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$. (The result is admittedly not pretty.)

$$B \times (A \times X) = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \times \begin{bmatrix} a_{11} \cdot x_1 + a_{12} \cdot x_2 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 \end{bmatrix} = \begin{bmatrix} b_{11}(a_{11} x_1 + a_{12} x_2) + b_{12}(a_{21} \cdot x_1 + a_{22} x_2) \\ b_{21}(a_{11} x_1 + a_{12} x_2) + b_{22}(a_{21} \cdot x_1 + a_{22} x_2) \end{bmatrix}$$

In-Class Exercise 20: Download MatrixVectorExample2.java and write matrix multiplication code to make the example work. Confirm that you obtained the same matrix in Exercise 17, and the same resulting vector.

The results fully align with ex17

```
Matrix (2x2):
  2.000 -1.000
  0.000 -3.000
Vector:  1.000 -9.000
```




```
MatrixVectorExample.java 3, U
MatrixVectorExample2.java 7, U
linAlg > mod3 > MatrixVectorExample2.java > MatrixVectorExample2 > print(double[][])

    static double[][] matrixMult (double[][] A, double[][] B)
    {
        /* Compute the product of A and B using the matrixVectorMult() method
         * and return the result. */

        // Get the number of rows for matrix A and columns for matrix B
        int rowsA = A.length;
        int colsB = B[0].length;

        // Check if the matrices can be multiplied
        if (A[0].length != B.length) {
            throw new IllegalArgumentException(s:"Number of columns of A must be equal to number of rows of B");
        }

        // Initialize the result matrix C
        double[][] C = new double[rowsA][colsB];

        // Compute each column of the result matrix C using matrixVectorMult method
        for (int j = 0; j < colsB; j++) {
            // Extract the j-th column from matrix B
            double[] columnB = new double[B.length];
            for (int i = 0; i < B.length; i++) {
                columnB[i] = B[i][j];
            }

            // Multiply matrix A with the extracted column vector
            double[] columnC = matrixVectorMult(A, columnB);

            // Assign the computed column vector to the result matrix C
            for (int i = 0; i < rowsA; i++) {
                C[i][j] = columnC[i];
            }
        }

        // Return the result matrix C
        return C;

PROBLEMS 141  DEBUG CONSOLE  PORTS  AZURE  COMMENTS  OUTPUT  TERMINAL  GL  GL

(base) C:\Users\togru\ADA\cla\linAlg\mod3>javac --module-path %PATH_TO_FX% --add-modules javafx.controls MatrixVectorExample2.java

(base) C:\Users\togru\ADA\cla\linAlg\mod3>java --module-path %PATH_TO_FX% --add-modules javafx.controls MatrixVectorExample2
Matrix (2x2):
  2.000 -1.000
  0.000 -3.000
Vector:  1.000 -9.000
```

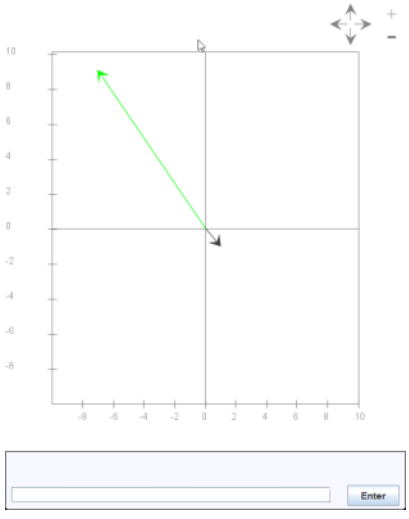
In-Class Exercise 21: Consider the matrices

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

Let $\mathbf{x} = (1, -1, 2)$ and define $\mathbf{y} = \mathbf{Ax}$ and $\mathbf{z} = \mathbf{By}$. Calculate each of $\mathbf{y}, \mathbf{z}$ and the matrix product $\mathbf{C} = \mathbf{BA}$. Confirm by calculating $\mathbf{Cx}$.

For this I used the earlier script for the purposes of saving time. The modified code is available in the file MatrixVectorExample2ex21. The results are given below and are consistent

```
(base) C:\Users\togru\ADA\cla\linAlg\mod3>java --m
Matrix A:
Matrix (3x3):
  3.000  2.000  1.000
 -2.000  3.000  5.000
  0.000  0.000  3.000
Matrix B:
Matrix (3x3):
 -4.000  1.000  0.000
  1.000  0.000  1.000
  3.000 -2.000  1.000
Matrix C=BA:
Matrix (3x3):
 -14.000 -5.000  1.000
   3.000  2.000  4.000
  13.000  0.000 -4.000
Vector y = Ax:
Vector:  3.000  5.000  6.000
Vector z = By:
Vector: -7.000  9.000  5.000
Vector Cx = Cx:
Vector: -7.000  9.000  5.000
```



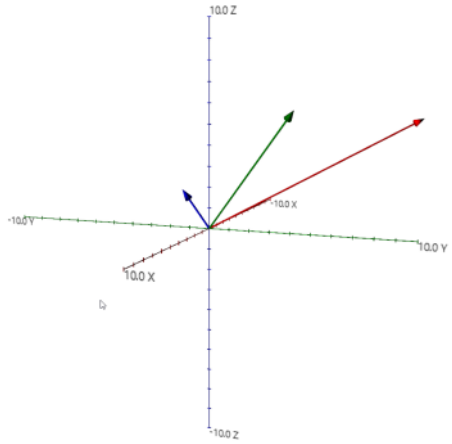

```
{
double[][] A = {
    {3, 2, 1},
    {-2, 3, 5},
    {0, 0, 3}
    };
System.out.println(x:"Matrix A:");
print(A);

double[][] B = {
    {-4, 1, 0},
    {1, 0, 1},
    {3, -2, 1}
    };

System.out.println(x:"Matrix B:");
print(B);

double[] x = {1, -1, 2};

double[][] C = matrixMult (B, A);
System.out.println(x:"Matrix C=BA:");
print (C);

double[] y = matrixVectorMult (A, x);
System.out.println(x:"Vector y = Ax:");
print (y);

double[] z = matrixVectorMult (B, y);
System.out.println(x:"Vector z = By:");
print (z);

double[] Cx = matrixVectorMult (C, x);
```

```
System.out.println(x:"Vector z = By:");
print (z);

double[] Cx = matrixVectorMult (C, x);
System.out.println(x:"Vector Cx = Cx:");
print (Cx);
```

In-Class Exercise 22: Download MatrixVector3DExample.java, then fill in the code (from previous exercises) to perform matrix multiplication and matrix-vector multiplication. Confirm that the results match the calculations in the previous exercise.

Matrix-matrix and matrix-vector multiplication methods were exactly the same as the above exercises. So I will not be attaching it in this section separately.

```
(base) C:\Users\togru\ADA\cla\linA
Vector:  3.000  5.000  6.000
Vector: -7.000  9.000  5.000
Matrix (3x3):
 -14.000 -5.000  1.000
   3.000  2.000  4.000
  13.000  0.000 -4.000
Vector: -7.000  9.000  5.000
```



In-Class Exercise 23: Suppose $\mathbf{A}$ and $\mathbf{B}$ are two $n \times n$ matrices. Prove or disprove: $\mathbf{AB} = \mathbf{BA}$.

Using the coed from the previous exercise, we get the results illustrated on the right. This means that AB != BA

```
(base) C:\Users\togru\ADA\cla\linAlg\mod3>java --module-path %P
Matrix A:
Matrix (3x3):
  3.000  2.000  1.000
 -2.000  3.000  5.000
  0.000  0.000  3.000
Matrix B:
Matrix (3x3):
 -4.000  1.000  0.000
  1.000  0.000  1.000
  3.000 -2.000  1.000
Matrix AB:
Matrix (3x3):
 -7.000  1.000  3.000
 26.000 -12.000  8.000
  9.000 -6.000  3.000
Matrix BA:
Matrix (3x3):
 -14.000 -5.000  1.000
  3.000  2.000  4.000
 13.000  0.000 -4.000
```

In-Class Exercise 24: What is the result of multiplying $\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 2 & 0 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 0 & -3 \end{bmatrix}$? What is $\mathbf{Ax}$ when $\mathbf{x} = (2, 3, 4)$?

```
(base) C:\Users\togru\ADA\cla\linAlg\mod3>java --module-path %
Matrix A:
Matrix (2x3):
 0.000  1.000  0.000
-1.000  2.000  0.000
Matrix B:
Matrix (3x2):
  1.000  1.000
  2.000  2.000
  0.000 -3.000
Matrix AB:
Matrix (2x2):
```

```
public static void main (String[] argv)
{
double[][] A = {
    {0, 1, 0},
    {-1, 2, 0}
    };
System.out.println(x:"Matrix A:");
print(A);

double[][] B = {
    {1, 1},
    {2, 2},
    {0, -3}
    };

System.out.println(x:"Matrix B:");
```

```
  1.000  1.000
  2.000  2.000
  0.000 -3.000
Matrix AB:
Matrix (2x2):
  2.000  2.000
  3.000  3.000
Vector x:
Vector:  2.000  3.000  4.000
Vector y = Ax:
Vector:  3.000  4.000
```

```
    {0, -3}
    };

System.out.println(x:"Matrix B:");
print(B);

double[] x = {2, 3, 4};

double[][] C = matrixMult (A, B);
System.out.println(x:"Matrix AB:");
print (C);

// double[][] D = matrixMult (B, A);
// System.out.println("Matrix BA:");
// print (D);
System.out.println(x:"Vector x:");
print (x);

double[] y = matrixVectorMult (A, x);
System.out.println(x:"Vector y = Ax:");
print (y);
```

Module 4

Monday, September 25, 2023 3:46 PM

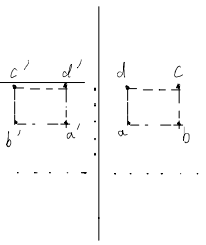
In-Class Exercise 1: On a piece of paper, draw the points $(2,3), (5,3), (5,5)$ and $(2,5)$ and join the dots to get a shape. Now, treating each of these tuples as a 2D vector, multiply each separately by the matrix $\mathbf{A} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ to get four new vectors. Draw the "points" (heads) corresponding to these vectors. What is the geometric relationship between the two shapes?

Matrix A will simply negate the sign of the x-axis values of all the vectors. The shape constructed by the "points" is a rectangle

$a = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $b = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$ $c = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$ $d = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$

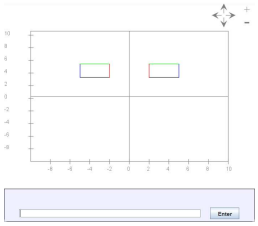
$Aa = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$ $Ab = \begin{bmatrix} -5 \\ 3 \end{bmatrix}$ $Ac = \begin{bmatrix} -5 \\ 5 \end{bmatrix}$ $Ad = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$

a' b' c' d'



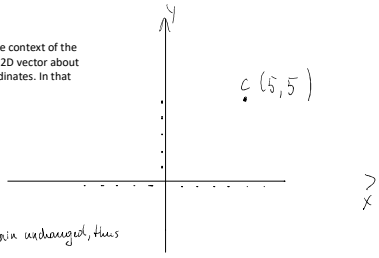
In-Class Exercise 2: Download GeomTransExample.java and MatrixTool.java, and add your matrix-vector multiplication code from earlier to `MatrixTool`. Then, confirm your calculations above. You will also need DrawTool.java.

As can be seen, the results are the same. Matrix A simply serves to reflect the original coordinates about the y axis.



In-Class Exercise 3: What matrix would resulting in reflecting a shape about the x-axis?

Every shape can be represented as the location of its corners. In the context of the previous ex., that is if we are looking for a matrix that will reflect a 2D vector about the x-axis then we need a matrix that'll only negate the y axis coordinates. In that case, this transformation matrix will be:



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

$Mc = c'$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 5 \\ -5 \end{bmatrix}$ We need the x-axis values to remain unchanged, thus

$M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ is the simplest matrix that will reflect a 2D shape about the x axis

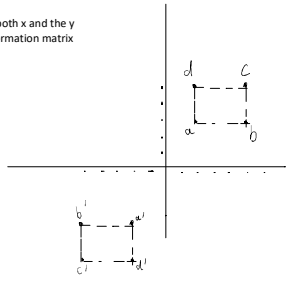
However, here we are considering the simplest matrix that will do the task in a single linear transformation in its reduced orthonormal form. There is an infinite number of matrices that can eventually reach the end goal with many rotations/reflections. This can be proven by solving the above given equation

In-Class Exercise 4: On paper, draw the same rectangle (the points $(2,3), (5,3), (5,5), (2,5)$) and reflect the rectangle about the origin. Use guesswork to derive the matrix that will achieve this transformation. Apply the matrix (call this matrix **B**) to the four corners to verify.

For reflection about the origin, we need both to reverse the signs for both x and the y coordinates. This will result in the shapes on the right. And the transformation matrix will be as shown below

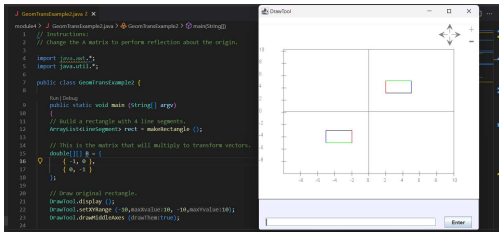
$B = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

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



In-Class Exercise 5: Download GeomTransExample2.java, and modify the matrix **B** in the code to achieve reflection about the origin, and confirm your calculations above.

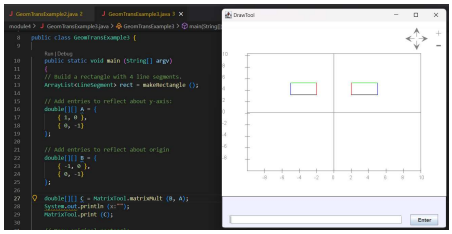
Observing the transformation matrix in the code and the resultant shapes, we can see that this is the right matrix for reflection about the origin



In-Class Exercise 6: Download GeomTransExample3.java, and insert the entries from the **A** and **B** matrices from above. Work out the product **BA** by hand and apply to the four corners of the rectangle to confirm what the program produces. Argue that the resulting matrix is intuitively what one would expect.

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

This is basically reflection about the y-axis. This is equivalent to reflecting about the origin and then about the x-axis.



$$Ca = \begin{bmatrix} -2 \\ 3 \end{bmatrix} \quad Cb = \begin{bmatrix} -5 \\ 3 \end{bmatrix}$$

$$Cc = \begin{bmatrix} -5 \\ 5 \end{bmatrix} \quad Cd = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$$

- Question 1: Can we multiply out all the matrices and apply the single resulting matrix to the vector and get the same result?
- Question 2: Is it permissible to do something like this?
  - Compute the product $\mathbf{B} = \mathbf{A}_2\mathbf{A}_3$ and substitute this:

$$\mathbf{A}_5(\mathbf{B}(\mathbf{A}_2(\mathbf{A}_1\mathbf{v})))$$

In-Class Exercise 7: What theoretical property of multiplication do we need to be true for matrices to resolve questions 1 and 2 above?

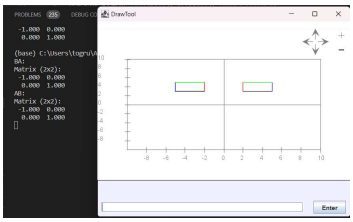
Q1: Yes, we can multiply all the matrices and apply the single resulting matrix to the vector to get the same result. The only constraint is that the matrices should be of compatible dimensions for multiplication.
Q2: The associativity principle which tells us that grouping the matrices in a multiplication sequence will not affect the final result since we are simply applying the resultant matrix on the others without influencing the original order of transformations.

$$A(BC) = (AB)C$$

In-Class Exercise 8: In your earlier program, GeomTransExample3.java, change the matrix multiplication order from the product **BA** to the product **AB**. What do you see? Does the order matter? Confirm by hand calculation. Is there a geometric reason to expect the result? What do you conclude about whether matrix multiplication is commutative?

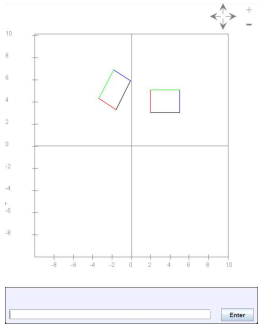
$$AB = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = C$$

The resultant matrix is the same. For this specific operation AB = BA. However, this is not generally true and the order does indeed matter. Matrix multiplication is NOT commutative. Here, since we are applying the same "type" of transformation (reflection), the results are the same.



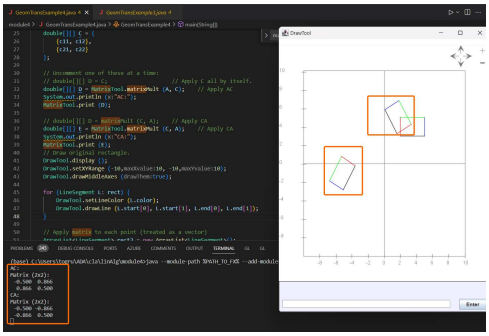
In-Class Exercise 9: Download GeomTransExample4.java, compile and execute. How would you describe the transformation?

This transformation is basically a combination of reflection about the y axis and rotation



In-Class Exercise 10: Now, let's apply the earlier reflection about y-axis and the transformation above in sequence. In GeomTransExample4.java, first apply **AC** and then change to the order to **CA**. What do you see? Does the order matter? Is there a geometric reason to expect the result?

The results are given on the right. We can see that the matrix multiplication AC = CA does NOT hold. The geometric interpretation behind this is that these are 2 different types of transformations. One is a rotation while the other is a reflection

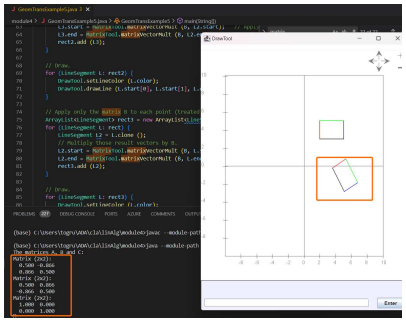


In-Class Exercise 11: Download GeomTransExample5.java, compile and execute. Observe that we apply two transformations in sequence: first, the matrix **A** from above, and then a new matrix **B**. What is the net effect in transforming the rectangle? From that, can you conclude what matrix **B** would achieve when applied by itself to a vector? What is the product matrix $\mathbf{C} = \mathbf{BA}$ when printed out? Consider the generic vector $\mathbf{v} = (v_1, v_2)$ and compute by hand the product $\mathbf{Cv}$.

The final result is that the original rect remains unchanged. The Matrix B is the inverse of A. We can also see the result of applying just B.

$$Cv = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

C is an Identity Matrix and hence will have not affect when multiplied



The following are generally useful properties:

- First, a non-property: matrix multiplication is, alas, NOT commutative: that is, in general,

$$\mathbf{AB} \neq \mathbf{BA}$$

- Matrix multiplication is *associative*:

$$\mathbf{A(BC)} = \mathbf{(AB)C}$$

- Matrix multiplication *distributes* over matrix addition:

$$\mathbf{A(B+C)} = \mathbf{(AB)} + \mathbf{(AC)}$$

- Scalar multiplication can be applied in any order:

$$\alpha\mathbf{(AB)} = \mathbf{(\alpha A)B} = \mathbf{A(\alpha B)}$$

In-Class Exercise 12: Prove the above properties.

1. The 1st property has already been proven above
2. This was also proven above
3. Proving this in matrix form will take too long so let's look at a different approach
4. Can be proven the same way as 3

Let $A \in \mathbb{R}^{m \times n}$, $B, C \in \mathbb{R}^{n \times p}$

$\Rightarrow$ i^{th} entry of $A(B+C) = \sum_{k=1}^{n} a_{ik}(b_{kj} + c_{kj})$ — This is evident if we consider the matrix multiplication

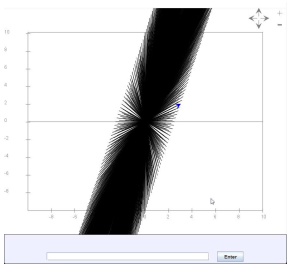
$\Rightarrow \sum_{k=1}^{n} a_{ik}b_{kj} + \sum_{k=1}^{n} a_{ik}c_{kj}$ — Distributivity of $\mathbb{R}$

$\Rightarrow AB + AC$ — Proven

- When applying transformations, order matters
- Inverse of a Matrix is basically a matrix that undoes the transformation applied by the transformation Matrix

In-Class Exercise 14: To explore this notion, write code in ExploreSpan.java to compute a linear combination. Observe the systematic exploration of different values of α, β. Change the range to see if you can "fill up" the space.

```
}

static double[] linComb (double alpha, double[] u, double beta, double[] v)
{
    double[] w = new double [u.length];

    // check if the vectors are the same length
    if (u.length != v.length) {
        System.out.println(x:"Vectors must be the same length");
        return null;
    }

    for (int i=0; i<u.length; i++) {
        w[i] = alpha * u[i] + beta * v[i];
    }

    return w;
}
```



In-Class Exercise 15: What's an example of vectors $\mathbf{u}, \mathbf{v}$ whose span is not the whole space of 2D vectors? Similarly, what's an example of 3D vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$, whose span is not the whole space of 3D vectors?

If the vectors are linearly dependent on each other, i.e. they lie on a single line in 2D or on the same plane on 3D, then their span is not the whole plane or 3D space respectively. By an extension, if the determinant of the transformation matrix composed of a combination of these vectors is 0, then the vectors are linearly dependent.

In-Class Exercise 16: Is there a pair of 3D vectors that spans the whole space of 3D vectors?

No, for 3D space, which is three-dimensional, you need three linearly independent vectors to span the entire space. Two vectors, no matter how they're oriented, will always span a plane in 3D space, not the entire space.

In-Class Exercise 17: Consider the pair of "special" vectors $\mathbf{e}_1 = (1,0), \mathbf{e}_2 = (0,1)$.

1. Express the vector $(1,4)$ as a linear combination of $\mathbf{e}_1, \mathbf{e}_2$.
2. Express the vector $(2,3)$ as a linear combination of $\mathbf{e}_1, \mathbf{e}_2$.
3. Do $\mathbf{e}_1, \mathbf{e}_2$ span 2D space?
4. What are the corresponding three "special" vectors that span 3D space?

1. $1 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$
2. $2 \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$
3. Yes as they are linearly independent
4. $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- For Affine transformation, we are basically projecting to a higher dimension, applying the transformations, and then projecting the resultant higher dimensional representation back into the original No. of. Dims.

In-Class Exercise 18: Explain why this is so. That is, why do we get $(\cos(\theta), \sin(\theta)$ and $(-\sin(\theta), \cos(\theta))$?

For basis vectors I and j,

1. When we rotate the vector i by an angle θ in the counterclockwise direction, its tip lands on the unit circle at a point defined by the coordinates (cos(θ), sin(θ)). This is by the definition of sine and cosine in trigonometry, where cosine gives the x-coordinate and sine gives the y-coordinate of a point on the unit circle after rotation by θ.
2. When we rotate j by θ, we essentially rotate it to a position 90 degrees ahead of the rotated ii. The trigonometric functions are periodic with period 2π. Shifting an angle by 90 degrees swaps sine and cosine values and introduces a negative sign due to the quadrant it lies in. Thus, the coordinates of the rotated j are (−sin(θ),cos(θ)).

In-Class Exercise 20: What happens when the approach is applied to translation? That is, suppose we want each point (x, y) to be translated to $(x+1, y+2)$.

Simple translations (all linear transformations) like the rotation from before cannot achieve this translation using just matrix multiplication. This is because multiplying a matrix (linear transformation) will always keep the origin fixed. However, when translating a point, the new basis vectors themselves will be shifted

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

$j' = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$

In-Class Exercise 21: Prove that no matrix multiplication can achieve translation. That is, no 2×2 matrix can transform every vector (x, y) to $(x+p, y+q)$.

Basically, any transformation, by definition, involves moving all points on the plane (in 2D) to a new location. This includes the origin as well. Let's take a generic matrix A and see if it is possible to move the origin.

Let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $\quad \vec{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\Rightarrow A\vec{0} = \begin{bmatrix} a \cdot 0 + b \cdot 0 \\ c \cdot 0 + d \cdot 0 \end{bmatrix}$ Linear Transformation of $\vec{0}$ by A

$\Rightarrow A\vec{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\Rightarrow$ it is not possible to move the origin

Thus, using matrix multiplication with a 2x2 matrix, the origin can never be moved. If the origin cannot be moved, then it's impossible to achieve a translation of all points in the plane. Therefore, no 2x2 matrix multiplication can achieve translation.

In-Class Exercise 22: Consider some 2D matrix $\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$ applied to a 2D vector (x, y). What is the resulting vector? What is the result of applying $\begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{21} & c_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ to $(x, y, 1)$?

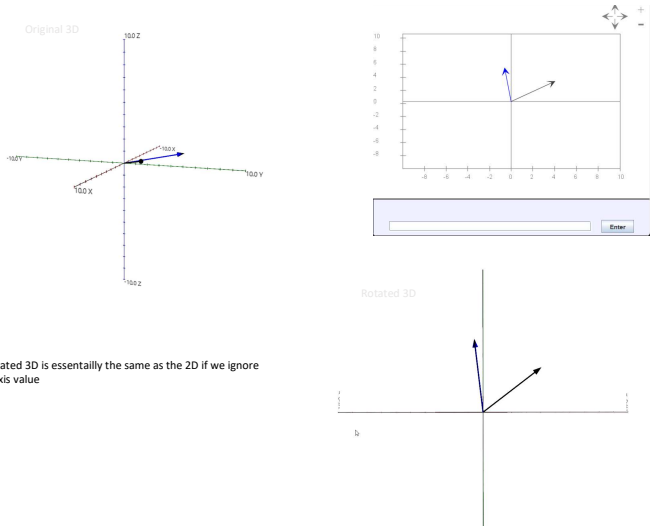
$a = \begin{bmatrix} x \\ y \end{bmatrix}$

$Ca = \begin{bmatrix} c_{11}x + c_{12}y \\ c_{21}x + c_{22}y \end{bmatrix}$

$C' = \begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{21} & c_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\quad a' = (x, y, 1)$

$C'a' = \begin{bmatrix} c_{11}x + c_{12}y \\ c_{21}x + c_{22}y \\ 1 \end{bmatrix}$

In-Class Exercise 23: Examine the code in AffineExample.java, then compile and execute. Then, do the same for Affine3DExample.java. Move the viewpoint so that you see the (x,y) axes in the usual way (looking along the z-axis), and compare with the 2D drawing.



The rotated 3D is essentailly the same as the 2D if we ignore the Z axis value

---

- So, what's important to know: given 2D transformations $\mathbf{A}, \mathbf{B}$, is

$$\mathbf{BAu} = \text{proj}(\text{affine}(\mathbf{B})\ \ \text{affine}(\mathbf{A})\ \ \text{affine}(\mathbf{u}))?$$

- In other words: if we apply affine extensions in sequence through matrix multiplication and *then* project, do we get the same result as we did with matrix multiplication in the lower dimension?

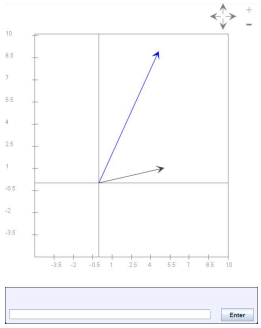
In-Class Exercise 24: Show that this is indeed the case.

$A' = \text{affine(A)} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $B' = \text{affine(B)} = \begin{bmatrix} b_{11} & b_{12} & 0 \\ b_{21} & b_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $u' = \text{affine(u)} = \begin{bmatrix} u_1 \\ u_2 \\ 1 \end{bmatrix}$

$A'B' =$

When we apply the above, the result will always have the third component always equal to 1. The first two components will be affected only by the 2x2 parts of A and B. And when we project, we simply drop the third dimension.

In-Class Exercise 25: Examine AffineExample2.java to see how rotation (by 60°) followed by translation by $(3, 4)$ works via combining the results into a single matrix, and applying it to the point $(5, 1)$. Compile & execute. It's confusing to see what translation does, so it's best to draw an arrow from the shifted origin, $(3, 4)$, to the new coordinates.



- What's interesting: the same matrix is also the identity for *matrices*:

$$\mathbf{IA} = \mathbf{A}$$

for any multiply-compatible matrix $\mathbf{A}$.

In-Class Exercise 26: Prove the above result. Is it true that $\mathbf{AI} = \mathbf{A}$? Is it possible to have an identity matrix for an $m \times n$ matrix?

Clearly, a "crank it out" proof works. But is there a connection with the interpretation we saw earlier, with the standard basis vectors?

Turns out, there is one. And it leads to a useful perspective on matrix multiplication in general.

The columns of a matrix product are the results of the matrix acting on the columns of the right-hand matrix. The identity matrix doesn't change the standard basis vectors. Thus, the i-th column of the identity matrix is precisely the i-th standard basis vector. So, when we multiply any matrix A with the identity matrix, each column of A is a linear combination of its columns where only the corresponding basis vector has coefficient 1; all others have coefficient 0. This results in the columns being unchanged.

So we can say that matrix multiplication is effectively expressing the columns of the right-hand matrix in terms of the linear combinations of the columns of the left-hand matrix. The identity matrix effectively does nothing to the matrix it multiplies.

For non-square matrices, even though we can't have a two-sided identity, we can have left and right identities.. For an m×n matrix A, there's no single matrix that always satisfies both A×I=A and I×A=A unless m=n.

Now for a key observation: $\mathbf{Ab}_i = \mathbf{c}_i$.

In-Class Exercise 27: Why is this true?

Because each c_i is a column (or a vector) on the n-dimensional space represented by C. So we are simply transforming each vector one by one and just expressing the results together.

In-Class Exercise 28: What is the "undo" matrix for the reflection about the y-axis? Multiply the undo matrix (on the left) and the original.

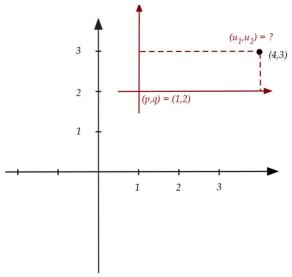
$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

In-Class Exercise 29: What is the "undo" affine-extended matrix for translation by (p, q)? Multiply the undo matrix (on the left) and the original.

$$M = \begin{bmatrix} 1 & 0 & p \\ 0 & 1 & q \\ 0 & 0 & 1 \end{bmatrix}$$

Since we want to apply translation by $\begin{bmatrix} -p \\ -q \end{bmatrix}$

$$\Rightarrow M^{-1} = \begin{bmatrix} 1 & 0 & -p \\ 0 & 1 & -q \\ 0 & 0 & 1 \end{bmatrix}$$



Suppose the new axes are translated by $(1, 2)$.

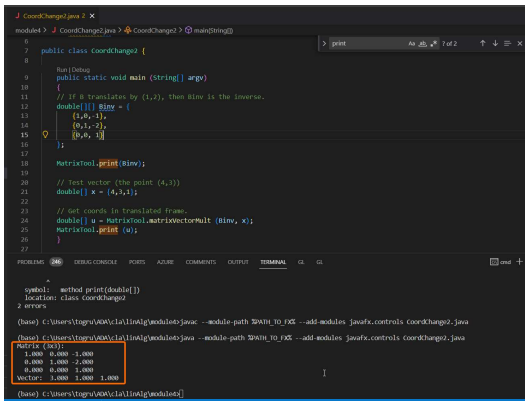
In-Class Exercise 30: Follow the steps to compute the affine-extended matrix that, when applied to the point $(4, 3)$, produces the coordinates in the new frame. Confirm by implementing in CoordChange2.java.

$$\begin{bmatrix} 1 & 0 & p \\ 0 & 1 & q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ 1 \end{bmatrix}$$

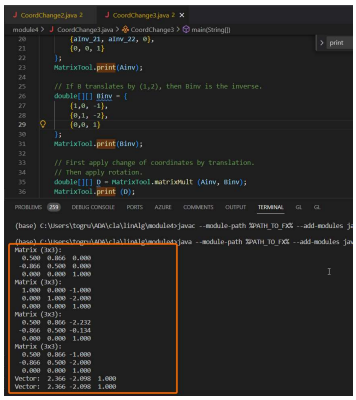
$$\begin{bmatrix} 1 & 0 & -p \\ 0 & 1 & -q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & p \\ 0 & 1 & q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -p \\ 0 & 1 & -q \\ 0 & 0 & 1 \end{bmatrix}}_{M^{-1}} \begin{bmatrix} u_1 \\ u_2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} u_1 - p \\ u_2 - q \\ 1 \end{bmatrix}$$

$$(p, q) = (1, 2) \Rightarrow M = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

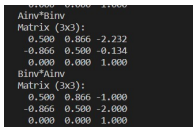


In-Class Exercise 31: Use the matrix for $\mathbf{B}^{-1}$ you computed earlier, and insert the matrix in CoordChange3.java to compute the coordinates in the new frame.

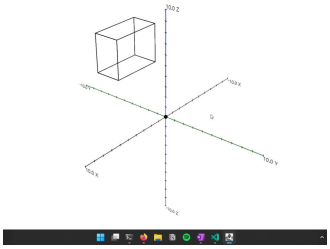


In-Class Exercise 32: Is $\mathbf{A}^{-1}\mathbf{B}^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$? Add a few lines of code to CoordChange3.java to see. Now, go back to the picture with the three frames above. If we were to first rotate the standard frame by 60° and then translate by $(1,2)$, would the resulting frame be the same as if we were to first translate and then rotate? Why then do we get different results when applying a different order to the inverse matrices?

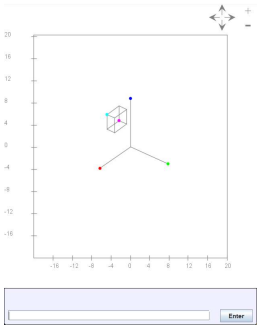
No, since this operation is the same as multiplication of any other 2 matrices. Matrix multiplication is not commutative as can be seen below



In-Class Exercise 33: Compile and execute CoordChange3D.java to see a cuboid drawn along with the position of an eye. Now move the view so that the eye lines up with the origin. Where do you see the cuboid? This is the 2D view we will build.



In-Class Exercise 34: Examine the code in CoordChange2D.java to see all the transforms implemented and multiplied. Compile and execute to see that we do indeed get the desired view.



- Now consider multiplication by

$$\mathbf{B}_\alpha = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}$$

- It's easy to see that $\mathbf{v} = \mathbf{B}_\alpha\mathbf{u}$ stretches the length by α.

In-Class Exercise 35: Verify this by hand.

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad B_\alpha u = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \alpha u_1 \\ \alpha u_2 \end{bmatrix} = \alpha \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

In-Class Exercise 36: What happens when $\mathbf{v} = \mathbf{u}$ in

$$\mathbf{v} \cdot \mathbf{u} = |\mathbf{v}||\mathbf{u}|\cos\theta?$$

Since the angle is 0, this is simply the magnitude of u (or v). Or the dot product of u by itself

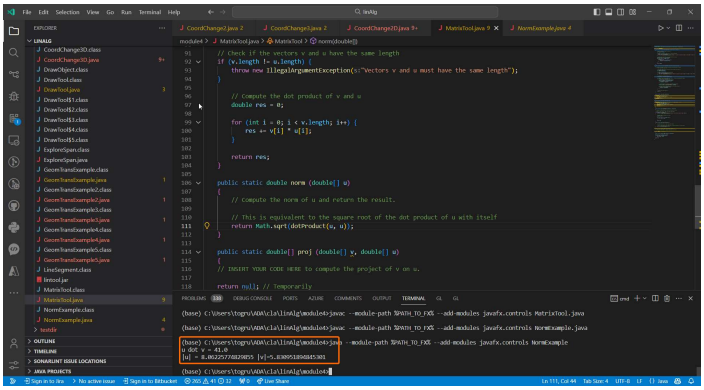
---

In-Class Exercise 37: If $\mathbf{A}$ is an orthogonal matrix, what is $\mathbf{A}^T\mathbf{A}$? What does this say about the inverse of $\mathbf{A}$? First see what you get with $\mathbf{A}$ as the rotation matrix from earlier.

$$\begin{bmatrix} - & c_1 & - \\ - & c_2 & - \\ - & c_3 & - \end{bmatrix}\begin{bmatrix} | & | & | \\ c_1 & c_2 & c_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$c_1 \cdot c_1 = |c_1| \cdot |c_1| \cdot \cos(0) = 1$$

In-Class Exercise 37: Implement dot product and norm in `MatrixTool` and test with NormExample.java.



In-Class Exercise 38: Verify that the columns of the reflection and rotation matrices are orthogonal vectors. Are they orthonormal too?

Ref. about x

$R_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ (v_1, v_2)

$v_1 \cdot v_2 = 1 \cdot -1 = 0 \Rightarrow v_1$ and v_2 are orthogonal

$\|v_1\| = 1 \quad \|v_2\| = 1 \Rightarrow$ the vectors are orthonormal (orthogonal and unit vectors).

Rotation by θ

$R_\theta = \begin{bmatrix} u_1 & u_2 \\ \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$

$u_1 \cdot u_2 = -\cos(\theta)\sin(\theta) + \cos(\theta)\sin(\theta) = 0$

$\|u_1\| = \sqrt{\sin^2(\theta) + \cos^2(\theta)} = \|u_2\| = 1 \Rightarrow$ orthogonal

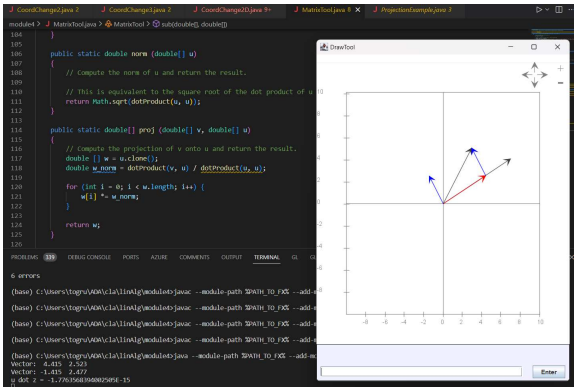
---

In-Class Exercise 39: If $\mathbf{A}$ is an orthogonal matrix, what is $\mathbf{A}^T\mathbf{A}$? What does this say about the inverse of $\mathbf{A}$? First see what you get with $\mathbf{A}$ as the rotation matrix from earlier.

$\begin{bmatrix} - & c_1 & - \\ - & c_2 & - \\ - & c_3 & - \end{bmatrix} \begin{bmatrix} | & | & | \\ c_1 & c_2 & c_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$c_1 \cdot c_1 = \|c_1\| \cdot \|c_1\| \cdot \cos(0) = 1$

In-Class Exercise 40: Implement projection in `MatrixTool` and test with ProjectionExample.java. Confirm that you get the same vectors as in the diagram above.



In-Class Exercise 41: What is the sum of $(1+2i, i, -3+4i, 5)$ and $(-2+i, 2, 4, 1+i)$? (Note: each has 4 components.)

$$\begin{bmatrix} -1+3i \\ 2+i \\ 1+4i \\ 6+i \end{bmatrix}$$

In-Class Exercise 42: What is the scalar product of $\alpha = (1-2i)$ and $\mathbf{u} = (1+2i, i, -3+4i, 5)$?

$$\alpha \mathbf{u} = \begin{bmatrix} 5 \\ 3+i \\ -3+6i+4i+8 \\ 5-10i \end{bmatrix} = \begin{bmatrix} 5 \\ 3+i \\ 5+10i \\ 5-10i \end{bmatrix}$$

In-Class Exercise 43: Work out the two products $(a+bi)(a+bi)$ and $(a+bi)(a-bi)$.

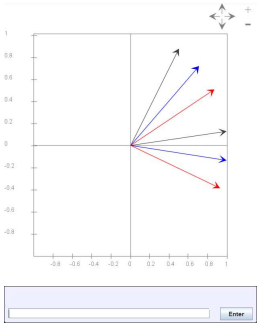
$p_1 = a + 2abi - b^2$
$p_2 = a^2 + b^2$

In-Class Exercise 44: For the complex vector $\mathbf{z} = (z_1, z_2, z_3) = (1+2i, i, 5)$, compute both $\mathbf{z} \cdot \mathbf{z}$ and $z_1^2 + z_2^2 + z_3^2$.

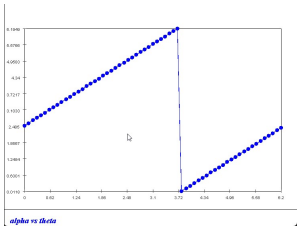
$z \cdot z = \bar{z}_1 z_1 + \bar{z}_2 z_2 + \bar{z}_3 z_3 = 5 + 1 + 25 = 31$

$z_1^2 + z_2^2 + z_3^2 = -3 + 4i - 1 + 25 = 21 + 4i$

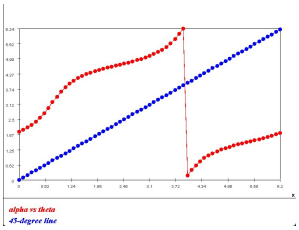
In-Class Exercise 45: Examine the code in OrthoExplore.java to confirm the generation method. You will also need UniformRandom.java (for this, and other exercises). Run a few times to see the results. Confirm that the matrix rotates the vectors but does not change their length.



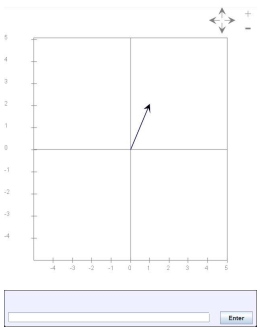
In-Class Exercise 46: Examine the code in OrthoExplore2.java to confirm the above approach. You will also need Function.java and SimplePlotPanel.java. Run a few times to see the results. Explain the graph: α vs. θ.



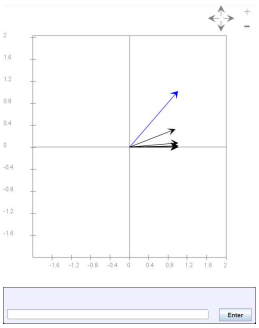
In-Class Exercise 47: Examine the code in MatrixExplore.java to confirm the above approach. Run a few times to see the results. What do you conclude about the points of intersection of the curve and the line?



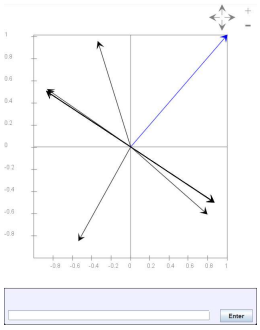
In-Class Exercise 48: Examine the code in MatrixExplore2.java to see that it's a simple example of a matrix that transforms the vector $\mathbf{u} = \langle 1, 2 \rangle$ into another vector. Compile and execute to draw the two vectors. Then try $\mathbf{u} = \langle 1, 0 \rangle$



In-Class Exercise 49: Examine the code in MatrixExplore3.java to see that the same matrix as in the previous exercise iteratively multiplies as above. What do you observe? Try a larger number of iterations, and different starting vectors.



In-Class Exercise 50: The program MatrixExplore4.java produces a random matrix each time it's executed. This matrix is applied iteratively to a starting vector. How often do you see fixed points emerging?



Module 6

Monday, October 9, 2023 3:50 PM

- Establishment of Linear Independence

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 2 \\ 3 & 1 \\ 0 & 2 \end{bmatrix} y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} x = 0$$

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

- Suppose we have three vectors

$$\mathbf{u} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} 6 \\ 1 \\ 7 \end{bmatrix}$$

and ask the question are there scalars α, β such that

$$\mathbf{w} = \alpha\mathbf{u} + \beta\mathbf{v}?$$

That is, can $\mathbf{w}$ be expressed as a linear combination of $\mathbf{u}$ and $\mathbf{v}$?

In-Class Exercise 1: How does one address the question for the above example?

The answer here lies in finding out whether the vectors u and v are linearly independent or not. We have already established that if two vectors are, when 3D is concerned, not on the same plane, then these vectors are enough to span the whole 3D space. Also, we should note that these vectors are not actually uniform basis vectors. For example, vector can be decomposed into the following.

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

So the answer simply depends on whether we can establish that the two vectors are linearly independent or not.

$$\begin{bmatrix} 2 & 0 & | & 6 \\ -1 & 1 & | & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \alpha & \beta \\ 1 & 0 & | & 3 \\ & & | & \end{bmatrix} \Rightarrow \alpha = 3$$

$$\left[\begin{array}{cc|c} 2 & 0 & 6 \\ -1 & 1 & 1 \\ 1 & 1 & 7 \end{array}\right] \xrightarrow{R_1/2} \left[\begin{array}{cc|c} 1 & 0 & 3 \\ -1 & 1 & 1 \\ 1 & 1 & 7 \end{array}\right] \Rightarrow \alpha = 3$$

$\alpha + \beta = 7 \Rightarrow \beta = 4$

In-Class Exercise 2: Can **u** be expressed in terms of **v** and **w**?

Let's test this by calculating as well

$$\left[\begin{array}{cc|c} 0 & 6 & 2 \\ 1 & 1 & -1 \\ 1 & 7 & 1 \end{array}\right] \qquad \beta = 1/3 \Rightarrow \alpha + \beta = -1$$

$\alpha = -4/3$

$\alpha + 7\beta = 1$

$-4/3 + 7/3 = 1$

Proven!

- On the other hand $\mathbf{z} = (6, 1, 8)$ cannot be expressed as a linear combination of **u** and **v**.

In-Class Exercise 3: Show that this is the case.

$$\left[\begin{array}{cc|c} 2 & 0 & 6 \\ -1 & 1 & 1 \\ 1 & 1 & 8 \end{array}\right] \xrightarrow{R_1/2} \left[\begin{array}{cc|c} 1 & 0 & 3 \\ -1 & 1 & 1 \\ 1 & 1 & 8 \end{array}\right] \Rightarrow \alpha = 3$$

$\alpha + \beta = 8 \Rightarrow \beta = 5$

$-\alpha + \beta = 1$, however

$-\alpha + \beta = 1$, however

$-3 + 5 \neq 1$, Contradiction

$\mathbf{w}$ is *linearly dependent* on $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ if there exist scalars $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that

$$\mathbf{w} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_n\mathbf{v}_n$$

Since if $\vec{v}$ is dependent on 2 others they are all dependent on each other, thus

- **Definition:** Vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are *linearly independent* if the only solution to the equation

$$\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_n\mathbf{v}_n = \mathbf{0}$$

is $\alpha_1 = \alpha_2 = \ldots = \alpha_n = 0$.

- The weirdness of the zero vector:
  - Suppose $\mathbf{v}_1 = \mathbf{0} = (0, 0, \ldots, 0)$.
  - Then, consider

$$\mathbf{v}_1 = \beta_2\mathbf{v}_2 + \ldots + \beta_n\mathbf{v}_n$$

  - Here, we can make $\beta_2 = \beta_3 = \ldots = \beta_n = 0$ so that

$$\mathbf{0} = 0\mathbf{v}_2 + \ldots + 0\mathbf{v}_n$$

  - That is, the zero vector is can always be interpreted as "dependent" on any other collection of vectors.
  - If we were to use the definition, we would write:

$$(-1)\mathbf{0} + 0\mathbf{v}_2 + \ldots + 0\mathbf{v}_n = \mathbf{0}$$

  - Thus, *no collection of vectors that includes the zero vector can be linearly independent.*

This is because this coefficient, by definition, would have to be 0 to assume linear indep., but that is not a necessity due to the 0 vect.

- The interpretation in terms of the RREF gives us a *method* to determine whether a collection of vectors is linearly independent.
  - Put the vectors in a matrix.
  - Compute the RREF.
  - See if the RREF = $\mathbf{I}$.

In-Class Exercise 5: Use the definition of linear independence to show both parts of Proposition 6.1 are true for this example, and then generalize to prove 6.1 for all RREFs.

Let's work on the following.

v_1
v_2
v_3
v_4

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

$\alpha_1 v_1 + \alpha_2 v_2 = 0$ only when $\alpha_1, \alpha_2 = 0$

$$\alpha_1 \begin{bmatrix} 1 \\ 0 \\ -1 \\ 3 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 4 \end{bmatrix} = \vec{x}$$ for the first 2 entries of $\vec{x}$ to be 0, the only possibility is for α_1 and α_2 to be 0.

v_3 and v_4 are 0 vec.

$\alpha_3 v_3 + \alpha v_4 = 0$ for any α_3 and α_4

$$\mathbf{c}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{c}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{c}_3 = \begin{bmatrix} -1 \\ 2 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{c}_4 = \begin{bmatrix} 3 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{c}_5 = \begin{bmatrix} 0 \\ 4 \\ 0 \\ 0 \end{bmatrix}$$

- Proposition 6.2 says that vectors $\mathbf{c}_1, \mathbf{c}_2$ are independent.
- And that any of $\mathbf{c}_3, \mathbf{c}_3, \mathbf{c}_5$ can be expressed as a linear combination of $\mathbf{c}_1, \mathbf{c}_2$.

In-Class Exercise 6: Use the definition of linear independence to show both parts of Proposition 6.2 are true for this example, and then generalize to prove 6.2 for all RREFs.

For the same reasons as above, c_1 and c_2 are independent and we can easily find the coeff. with which c_3, c_4 & c_5 can be obtained using just c_1 and c_2

- Suppose we have the following vectors

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{v}_3 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \end{bmatrix} \quad \mathbf{v}_4 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{v}_5 = \begin{bmatrix} 3 \\ -1 \\ 2 \\ 0 \end{bmatrix}$$

- Consider $\mathbf{v}_2$ and $\mathbf{v}_4$. Then, it's easy to see that
  - $\mathbf{v}_2$ and $\mathbf{v}_4$ are independent.
  - $\mathbf{v}_1 = \mathbf{v}_2 - \mathbf{v}_4$
  - $\mathbf{v}_3 = \mathbf{v}_2 + \mathbf{v}_4$
  - $\mathbf{v}_5 = 3\mathbf{v}_2 - \mathbf{v}_4$

  This suggests that we need at least two of these vectors to express the others.

- Similarly, it's easy to show that
  - $\mathbf{v}_1$ and $\mathbf{v}_2$ are independent.
  - $\mathbf{v}_3 = -\mathbf{v}_1 + 2\mathbf{v}_2$
  - $\mathbf{v}_4 = -\mathbf{v}_1 + \mathbf{v}_2$
  - $\mathbf{v}_5 = \mathbf{v}_1 + 2\mathbf{v}_2$

In-Class Exercise 7: Show that the above is true given what was shown earlier for $\mathbf{v}_2$ and $\mathbf{v}_4$.

These vect are all on 3D since el. 4 is 0. v_2 and v_4 both are composed of unit vectors that can span the entire 3D space.

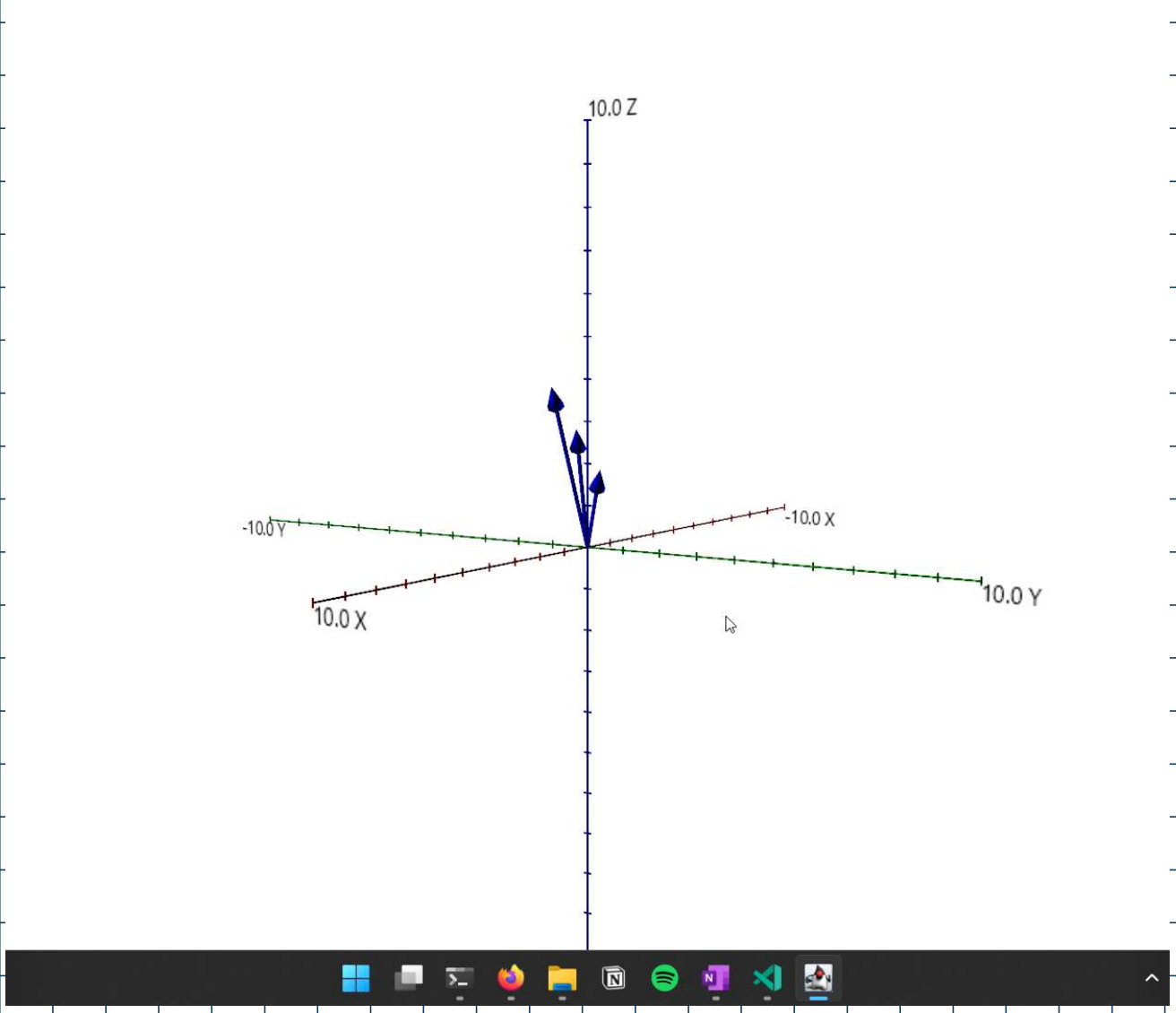
In-Class Exercise 8: Prove the following: if the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are independent, then so is any subset of these vectors.

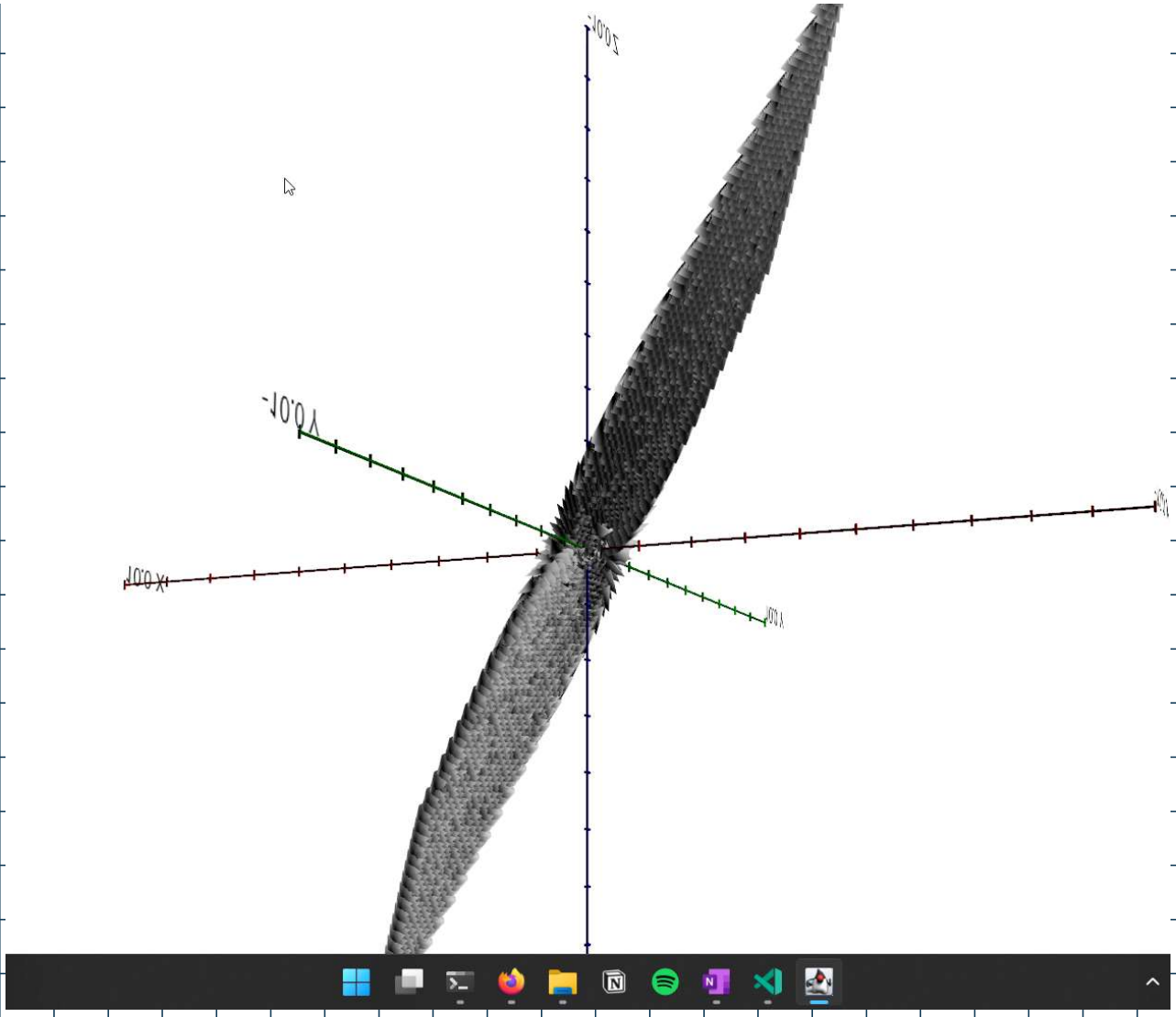
$$\alpha_1 v_1 + \alpha_2 v_2 + \ldots \alpha_n v_n = 0$$

$$\alpha_1 v_1 + \alpha_2 v_2 = 0$$

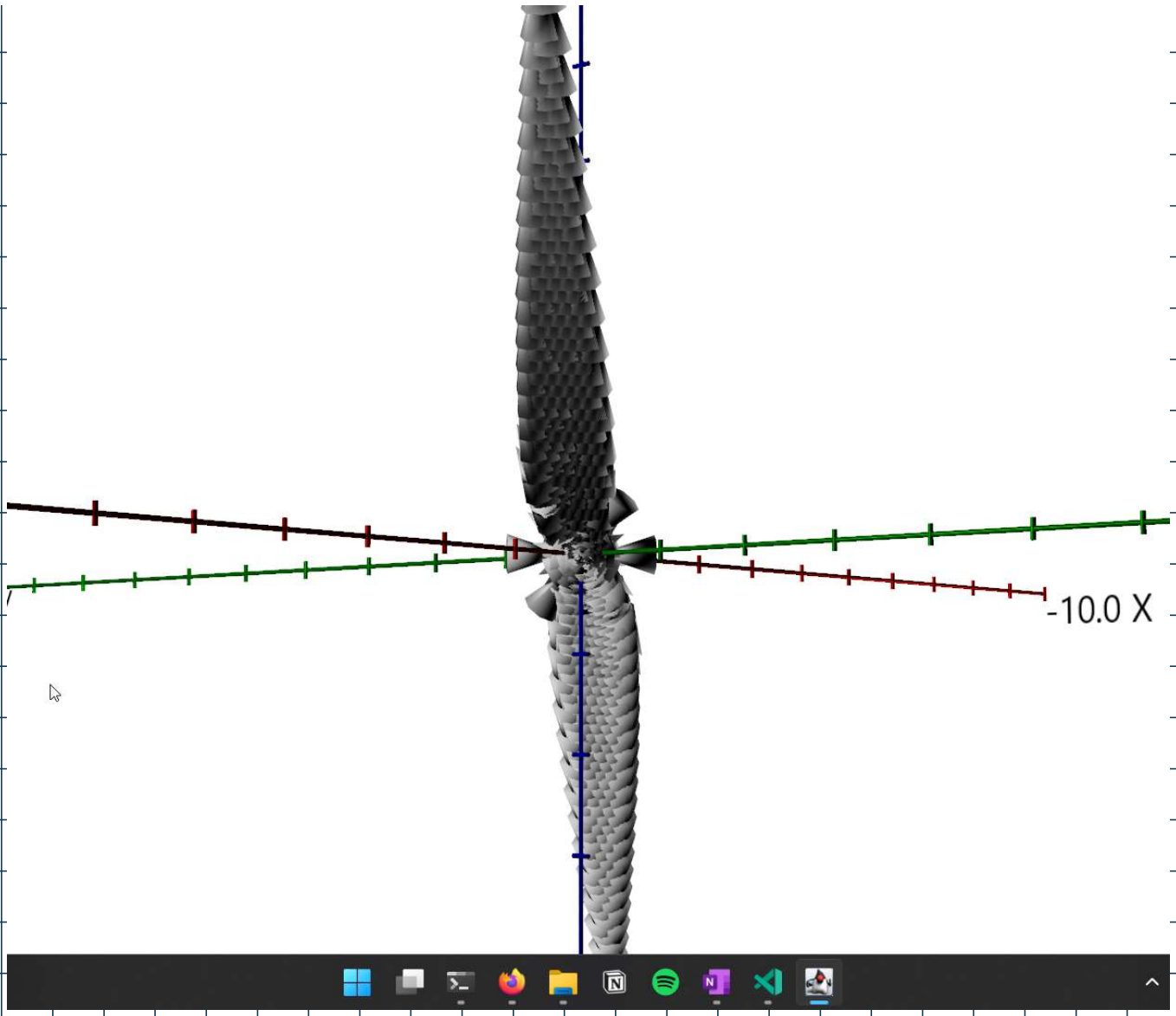
$$-v_1 = \frac{\alpha_2 \alpha_2}{\alpha_1}$$

In-Class Exercise 9: In 3D, explore the span of $\mathbf{u} = (1, 1, 2), \mathbf{v} = (2, 1, 3), \mathbf{w} = (3, 1, 4)$ in Span3DExample.java. You will need your `MatrixTool.java` from earlier, or you can use your implementation of `LinTool` to compute scalar multiplication and addition of vectors.





In-Class Exercise 10: We'll now explore the span of just two of the above vectors: $\mathbf{u} = (1, 1, 2)$, $\mathbf{v} = (2, 1, 3)$ in Span3DExample2.java. Is the span of the two the same as the span of the three? Try different bounds for the scalars. What is the span of just $\mathbf{u} = (1, 1, 2)$ all by itself, and is that the same as any of the other spans?



- Row operations do not change the rowspace, however the columnspace does change!

Thus, for a Matrix A and its RREF A'

$$\text{rowspace}(A) = \text{rowspace}(A')$$

$$\text{columnspace}(A) \neq \text{columnspace}(A')$$

- Basis vectors for a Matrix can be obtained by taking the RREF

- Basis vectors for a Matrix can be obtained by taking the RREF

In-Class Exercise 11: On paper, draw the vectors $\mathbf{u} = (1,1,0), \mathbf{v} = (-1,1,0), \mathbf{w} = (0,-1,0)$. What is the set of vectors spanned by these three vectors? What two obvious vectors should one use to span the same set?

u, v and w span the vector space that can be described as below:

$$\alpha\begin{bmatrix}1\\0\\0\end{bmatrix} + \beta\begin{bmatrix}0\\1\\0\end{bmatrix}$$ → the basis vectors needed to span the same set.

This leads to the next important concept: *dimension of a subspace*

- **Definition:** The *dimension* of a subspace is the minimum *number* of vectors needed to generate the subspace via linear combinations.
- We saw earlier that for the subspace generated by $\mathbf{u} = (1,1,0), \mathbf{v} = (-1,1,0), \mathbf{w} = (0,-1,0)$
  - $\mathbf{u} = (1,1,0), \mathbf{v} = (-1,1,0)$ are enough to generate the subspace
  - Alternatively, $\mathbf{e}_1 = (1,0,0), \mathbf{e}_2 = (0,1,0)$ are sufficient.
  - No single vector by itself can generate the (x,y)-plane.

In-Class Exercise 12: Why?

Because a single vector represents a line with its origin at the origin of the coord. sys. it belongs to. This means that any linear trans. will merely stretch this line (a 1D object) as opposed adding a new Dimension.

- Consider $\mathbf{W} = \text{span}(\mathbf{u}, \mathbf{v}, \mathbf{w})$ where $\mathbf{u} = (1,1,0), \mathbf{v} = (-1,1,0), \mathbf{w} = (0,-1,0)$.
  - We saw earlier that $\dim(\mathbf{W}) = 2$.
  - By Proposition 6.5, $\mathbf{u}, \mathbf{v}, \mathbf{w}$ should *not* be independent.
  - Which is indeed the case.

In-Class Exercise 13: Show that $\mathbf{u}, \mathbf{v}$ are independent and that $\mathbf{w}$ is a linear combination of $\mathbf{u}$ and $\mathbf{v}$.

$$\left[\begin{array}{cc|c} 1 & -1 & 0 \\ 1 & 1 & -1 \\ 0 & 0 & 0 \end{array}\right] \quad R_1 = (R_1 + R_2)/2 \Rightarrow \quad \left[\begin{array}{cc|c} \alpha & \beta & \\ 1 & 0 & -0.5 \\ 1 & 1 & -1 \\ 0 & 0 & 0 \end{array}\right] \Rightarrow \quad \alpha = -0.5 \quad \beta = 0.5$$

In-Class Exercise 14: Prove Proposition 6.5. Hint: start by assuming that one vector among $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is dependent on the others. Can the span be generated by the others?

Let's assume v_1 is lin dep. on the others

$= v_1 = \beta_2 v_2 + \beta_3 v_3 + \ldots + \beta_n v_n$

This means that v_1 lies in the subspace spanned by the other vectors.

$\Rightarrow \dim W = n-1$, Contradiction

We will prove Theorem 6.6 in steps:

- First, recall the three row operations in creating an RREF:
  - Swapping two rows.
  - Scaling a row by a number.
  - Adding to a row a scalar multiple of another row.
- Now consider an $m \times n$ matrix $\mathbf{A}$ with rows $\mathbf{r}_1, \ldots, \mathbf{r}_m$ and its rowspace: $\text{span}(\mathbf{r}_1, \ldots, \mathbf{r}_m)$.
- Suppose we apply a row operation to $\mathbf{A}$ and $\mathbf{A}'$ is the resulting matrix.

In-Class Exercise 15: Show that the rowspace of $\mathbf{A}'$ is the same as the rowspace of $\mathbf{A}$. Do this for each of three types of row operations.

If we pay attention, these operations are all lin. trans,

$$\begin{bmatrix} - & r_1 & - \\ - & r_2 & - \\ - & r_3 & - \end{bmatrix} \updownarrow \Rightarrow \begin{bmatrix} - & r_2 & - \\ - & r_1 & - \\ - & r_3 & - \end{bmatrix} \rightarrow \text{no changes to the vectors.}$$

The scaling operation will not change the linear combination of the vectors either, so it'll not affect the resultant rowspace
When it comes to the addition of rows, this is simply a linear combination and can easily be reversed. Meaning that the resultant vectors can be decomposed into the original vectors and is still within the span of the 2 original vectors. This means that the rowspace remains unchanged.

In-Class Exercise 16: Why are the pivot rows independent?

the pivot rows assure that within the same columns all entries except the Diag. are 0s. This means that for a row vector whose i^{th} element has no non-zero counterpart in the other row vectors no scalar except for 0 can be used to get a zero-vector as a lin. comb. of all row vects. > Upper Triangular Structure.

- Which proves Theorem 6.6.
- The proof of Proposition 6.4 is now trivial because we've proved something far stronger.
- Let's remind ourselves ...
  Proposition 6.4: The rank of the collection $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is the *number* of pivot columns of $RREF(\mathbf{A})$.

In-Class Exercise 17: Complete the proof of Proposition 6.4.

The rank of a matrix is defined as the maximum number of linearly independent column vectors in the matrix. This is also equal to the dimension of the column space of the matrix. When a matrix A is transformed into RREF, the pivot columns correspond to the positions of the leading 1's in the rows. These pivot positions, and thus the pivot columns, are linearly independent by definition. The column space of A and the column space of RREF(A) may not be the same because row operations can change the column space. However, the number of linearly independent columns (the rank) remains the same.

Rank = Num Lin. Indep. Cols = Num. Pivot Cols in RREF

In-Class Exercise 18: Prove or disprove the following. Suppose $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is a collection of m-dimensional *linearly independent* vectors and $\mathbf{u} \in \text{span}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$. Then there is exactly one linear combination of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ that will produce $\mathbf{u}$. Consider both cases, $m = n$ and $m \neq n$.

Since these, supposed to be, row vectors are linearly independent and there are n of them in an n-dimensional space, they form a basis for this space R^n. And in this space any point will be unique and can only be expressed as a unique combination of the basis vectors

When n<m
If the vectors are in R^m but there are fewer than m of them (n<m), they cannot the whole.
However, they do span a subspace of R^m, R^n.
Within this subspace, any vector u can still be expressed as a unique linear combination of v1,v2,...,vn because the vectors are linearly independent

In-Class Exercise 19: What is the more natural basis (3D vectors) for the (x,y) plane?

$$u = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

In-Class Exercise 20: Prove this result:

- Apply the definition of linear independence to $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$.
- Then compute the dot product with $\mathbf{v}_1$ on both sides.
- What does this imply for the value of α_1?
- Complete the proof.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + \cdots \alpha_n v_n = 0$$

$$v_1 \cdot 0 = v_1 \cdot 0$$

$$\alpha_1 \cdot \|v_1\|^2 = 0 \Rightarrow \alpha_1 = 0$$

We apply this recursively for $v_2 \ldots v_n$ and it becomes obv that they are all orthogonal as

$$\alpha_2 = 0, \ldots \alpha_n = 0$$