Module 4

Monday, September 25, 2023 3:46 PM

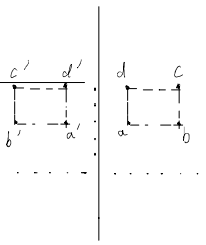
In-Class Exercise 1: On a piece of paper, draw the points $(2,3), (5,3), (5,5)$ and $(2,5)$ and join the dots to get a shape. Now, treating each of these tuples as a 2D vector, multiply each separately by the matrix $\mathbf{A} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ to get four new vectors.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

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

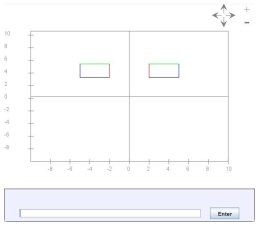
$Aa = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$ $Ab = \begin{bmatrix} -5 \\ 3 \end{bmatrix}$ $Ac = \begin{bmatrix} -5 \\ 5 \end{bmatrix}$ $Ad = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$

a' b' c' d'



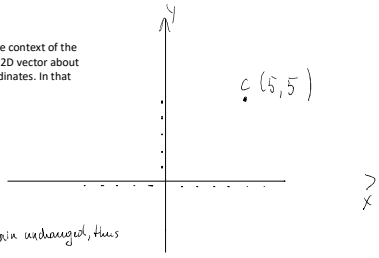
In-Class Exercise 2: Download GeomTransExample.java and MatrixTool.java, and add your matrix-vector multiplication code from earlier to `MatrixTool`. Then, confirm your calculations above. You will also need DrawTool.java.

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

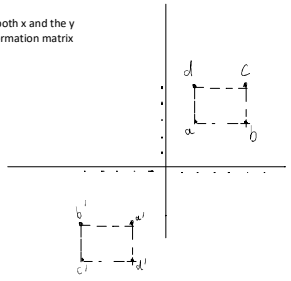
However, here we are considering the simplest matrix that will do the task in a single linear transformation in its reduced orthonormal form. There is an infinite number of matrices that can eventually reach the end goal with many rotations/reflections. This can be proven by solving the above given equation

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

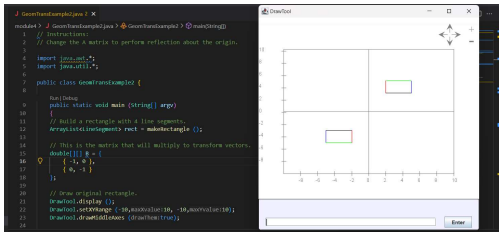
$B = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

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



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.

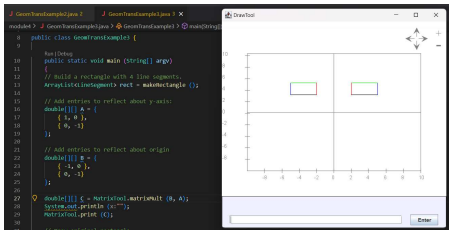
Observing the transformation matrix in the code and the resultant shapes, we can see that this is the right matrix for reflection about the origin



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?

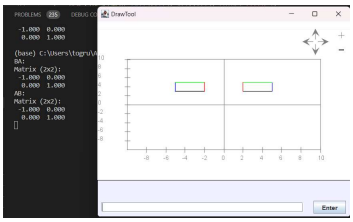
Q1: Yes, we can multiply all the matrices and apply the single resulting matrix to the vector to get the same result. The only constraint is that the matrices should be of compatible dimensions for multiplication.
Q2: The associativity principle which tells us that grouping the matrices in a multiplication 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?

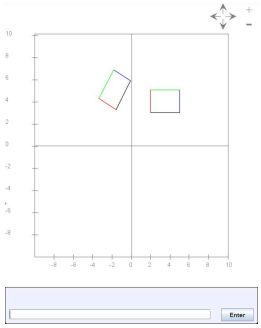
$$AB = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = C$$

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.



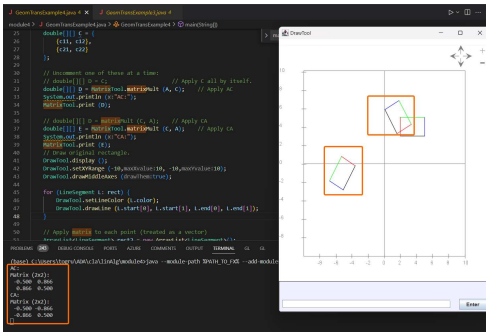
In-Class Exercise 9: Download GeomTransExample4.java, compile and execute. How would you describe the transformation?

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

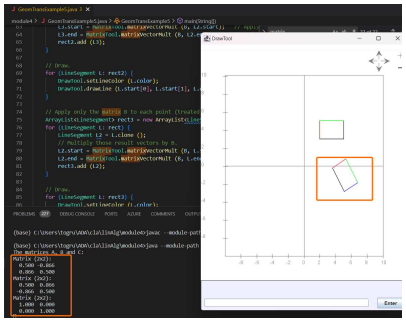


In-Class Exercise 11: Download GeomTransExample5.java, compile and execute. Observe that we apply two transformations in sequence: first, the matrix **A** from above, and then a new matrix **B**. What is the net effect in transforming the rectangle? From that, can you conclude what matrix **B** would 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.

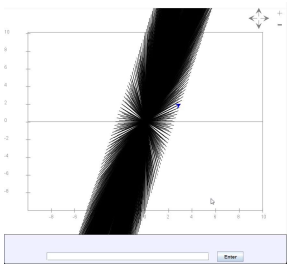
```
}

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 $(x, y, 1)$?

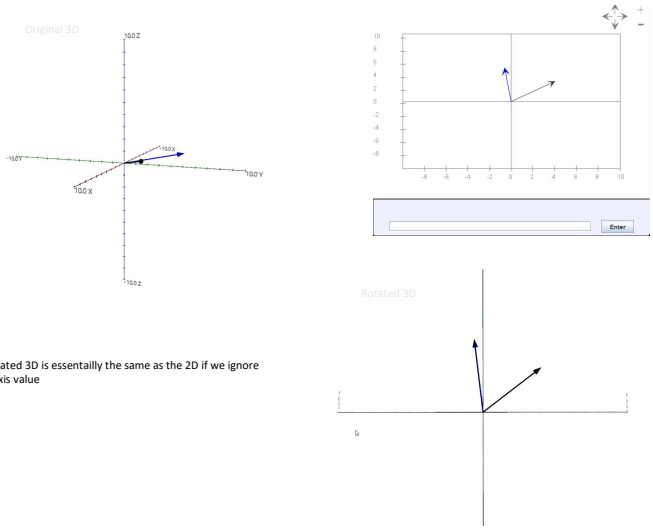
$a = \begin{bmatrix} x \\ y \end{bmatrix}$

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

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

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

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?

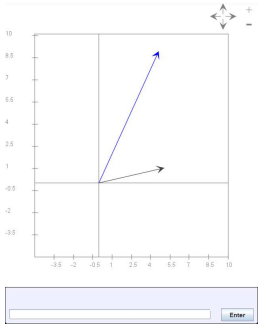
In-Class Exercise 24: Show that this is indeed the case.

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

$A'B' =$

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.

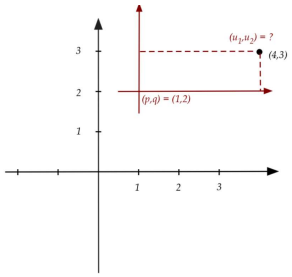
$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

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 $(1, 2)$.

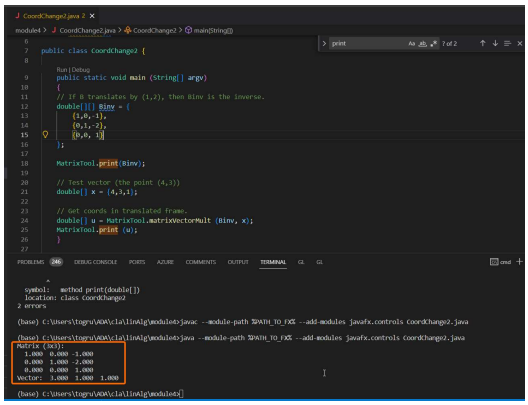
In-Class Exercise 30: Follow the steps to compute the affine-extended matrix that, when applied to the point $(4, 3)$, produces the coordinates in the new frame. Confirm by implementing in CoordChange2.java.

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

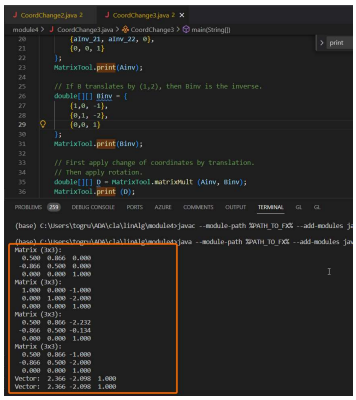
$$\begin{bmatrix} 1 & 0 & -p \\ 0 & 1 & -q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & p \\ 0 & 1 & q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -p \\ 0 & 1 & -q \\ 0 & 0 & 1 \end{bmatrix}}_{M^{-1}} \begin{bmatrix} u_1 \\ u_2 \\ 1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

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

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

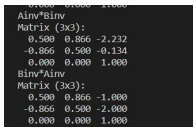


In-Class Exercise 31: Use the matrix for $\mathbf{B}^{-1}$ you computed earlier, and insert the matrix in CoordChange3.java to compute the coordinates in the new frame.

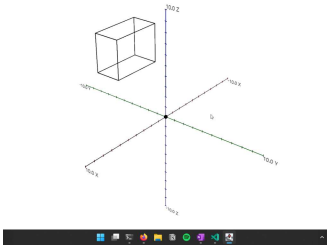


In-Class Exercise 32: Is $\mathbf{A}^{-1}\mathbf{B}^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$? Add a few lines of code to CoordChange3.java to see. Now, go back to the picture with the three frames above. If we were to first rotate the standard frame by 60° and then translate by $(1,2)$, would the resulting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

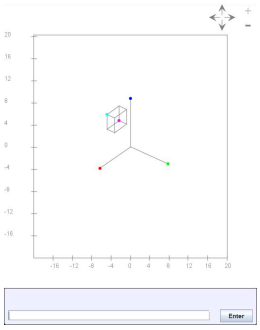
No, since this operation is the same as multiplication of any other 2 matrices. Matrix multiplication is not commutative as can be seen below



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

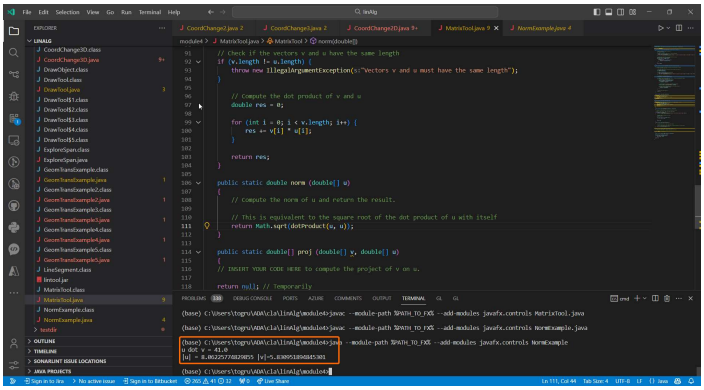
---

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

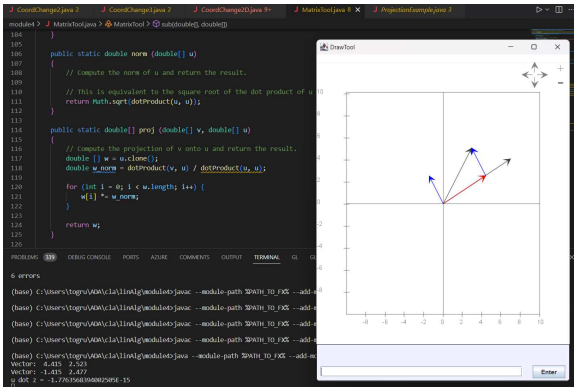
---

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 $(a+bi)(a-bi)$.

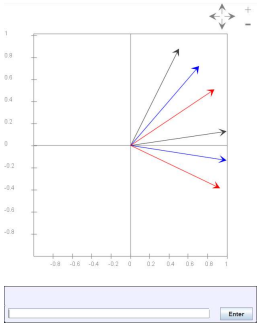
$p_1 = a + 2abi - b^2$
$p_2 = a^2 + b^2$

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 $z_1^2 + z_2^2 + z_3^2$.

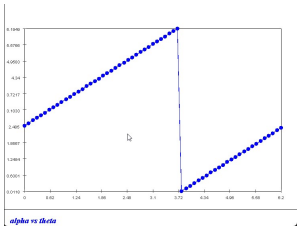
$z \cdot z = \bar{z}_1 z_1 + \bar{z}_2 z_2 + \bar{z}_3 z_3 = 5 + 1 + 25 = 31$

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

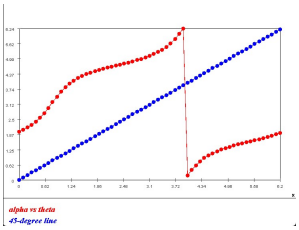
In-Class Exercise 45: Examine the code in OrthoExplore.java to confirm the generation method. You will also need UniformRandom.java (for this, and other exercises). Run a few times to see the results. Confirm that the matrix rotates the vectors but does not change their length.



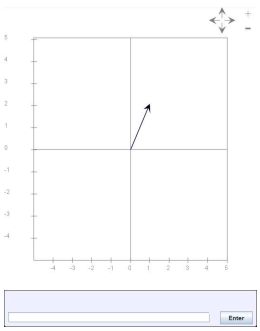
In-Class Exercise 46: Examine the code in OrthoExplore2.java to confirm the above approach. You will also need Function.java and SimplePlotPanel.java. Run a few times to see the results. Explain the graph: α vs. θ.



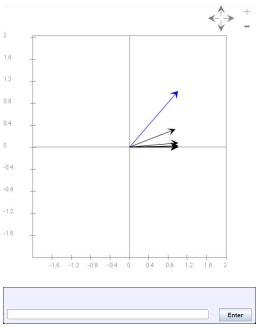
In-Class Exercise 47: Examine the code in MatrixExplore.java to confirm the above approach. Run a few times to see the results. What do you conclude about the points of intersection of the curve and the line?



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?