Aricel Mendez
December 13, 2022
Math120
Dr. Banuelos

Final draft

I. Introduction

For the final I chose the topic, Developing Interactive Teaching Activities + lesson plan. The goal of my teaching activities/lesson plan is to demonstrate how we can use Python to solve linear algebra problems. Linear algebra is the study of vectors and linear functions. A vector has both magnitude and direction. Vectors are used to describe the velocity of moving objects. We will focus on how we multiply two vectors by each other in linear algebra. I will introduce each topic with their definitions and I will also demonstrate how to use the dot and cross product with multiple examples. I will also incorporate computer programming using python to support the instruction. The two concepts we will do in this lesson plan are dot product and cross product. The methods I incorporated when programming was creating a function of the dot product and cross product and for the second method I used numpy to solve using the dot product and cross product. The questions I will be focusing on are; How do you multiply vectors? What is the dot product and cross product? Is there a reason for when to use which? How can you utilize computer programming to solve problems when using the dot or cross product?

II. Methods and Results

Dot Product:

Vectors are quantities of both a magnitude and a direction. Scalars are quantities described only by a magnitude (or numerical value) alone. When it comes to multiplying a vector by another vector there are two methods we use in order to multiply vectors. The first method is the dot product method. With this method you will see that when you multiply the two vectors together your answer will result in a scalar. When using the dot product you are simply just taking the product of each of their corresponding components and then adding them all together where you will just get a scalar.
Here I have vectors $\overline{a}$ and $\overline{b}$ and we want to do the dot product, $\overline{a} \bullet \overline{b}$.

$$\begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_n \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_n \end{bmatrix} = a_1b_1 + a_2b_2 + ... + a_nb_n$$

Here I just went through and multiped a_1 by b_1, then a_2 by b_2 and so on. Then once you have multiplied all the corresponding components you will add them all together and end with a scalar for your answer.

Next let's try some examples. Here I have two vectors and now I'm going to use the dot product to multiply them by each other.

$$\begin{bmatrix} 2 \\ 5 \end{bmatrix} \times \begin{bmatrix} 7 \\ 1 \end{bmatrix} = 2 \times 7 + 5 \times 1 = 14 + 5 = 19$$

Just like we did above with the dot product here we are taking 2×7 and 5×1 then adding the products together. Therefore when multiplying these two vectors we end up with 19 for our answer.

Let's do one last example,

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \times \begin{bmatrix} -2 \\ 0 \\ 5 \end{bmatrix} = 1 \times (-2) + 2 \times 0 + 3 \times 5 = 13$$

For the dot product I created a function that will calculate the dot product using two arrays. Within the function there will be a variable named dot that will store the sum of all the products. To find the dot product there will be a for loop that will iterate through both of the arrays and compute the product of each index. This function can be really useful for students to double check their answer when computing a dot product.

Another interesting way to compute the dot product with python is to use numpy. Importing Numpy will allow us to use the built in functions that numpy has. A benefit of this would make it even simpler for students to be able to compute a dot product on the spot using numpy.dot() rather than having to create a function like I previously did. Python is really interesting because we can use many useful built-in tools and libraries like numpy.

To get the dot product using numpy there are actually two built in functions that compute the dot product a little differently. There is numpy.dot() and numpy.vdot(). The difference from the two is one will take in arrays as the arguments and one will take in vectors as the arguments for the function. The vdot function will handle the complex numbers differently. When the first vector argument is a complex number then it will get the conjugate of the complex number and use this for the dot product calculations.

Cross product:

The second method of vector multiplication is the cross product which is useful but limited when compared to the dot product. The dot product is defined in any dimension

but for the cross product it is only defined in components of three. Thus the cross product is used only to multiply vectors with 3 components. When multiplying two vectors using the cross product we get another vector as our answer.
Here I have vectors $\overline{a}$ and $\overline{b}$ and we want to do the cross product, $\overline{a} \times \overline{b}$.

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix}$$

The cross product is a bit more tricky than the dot product. Let's do an example, to get the first row of your answer you will ignore the top components from the vectors that are being multiplied which are 1 and 5 and then you need to get the product of the components -7 and 4 minus the product of the components -1 and 2 which gives you the first row on your answer, -32.

$$\begin{bmatrix} 1 \\ -7 \\ 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} -7 \times 4 - 1 \times 2 \\ . \\ . \end{bmatrix} = \begin{bmatrix} -32 \\ . \\ . \end{bmatrix}$$

Then for the second row of your answer you ignore the second components which are -7 and 2 and just cross multiply and subtract the components $1 \times 5 - 1 \times 4$ which gives us our second row of your answer, 1.

$$\begin{bmatrix} 1 \\ -7 \\ 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} -7 \times 4 - 1 \times 2 \\ 1 \times 5 - 1 \times 4 \\ . \end{bmatrix} = \begin{bmatrix} -30 \\ 1 \\ . \end{bmatrix}$$

Then for the last row of your answer you would now ignore the third components in the vectors that are being multiplied and do the same procedure, $1 \times 2 - (-7) \times 5$. Our answer results in another vector as you see here which is what you will get when using the cross product.

$$\begin{bmatrix} 1 \\ -7 \\ 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} -7 \times 4 - 1 \times 2 \\ 1 \times 5 - 1 \times 4 \\ 1 \times 2 - (-7) \times 5 \end{bmatrix} = \begin{bmatrix} -30 \\ 1 \\ 37 \end{bmatrix}$$

For the cross product I created a function that will also compute the cross product using two arrays that you can assign values to. Within this function it is doing all of the computations when I initialize the output variable. It will grab the index of each array and multiply them together then subtract for each step in solving the cross product. Within the numpy library we can also just compute the cross product using numpy.cross(). This built-in function can compute the cross product even if the first array only contains two numbers and the second array contains three. The third missing component of the first array will just be assumed to zero.

III. **Discussion**

The dot product and cross product are also commonly used in physics. The cross product is used to get the vector that is perpendicular to the plane surface which is formed by two vectors. Then for the dot product we use it to find the angle between two vectors and also the length of the vector. I wanted to create an easy to follow lesson plan for these topics because these methods are brought in many classes such as calculus, physics, and linear algebra. Therefore this is a nice lesson to look back on for a refresher or also to relearn and get some practice with the dot and cross product. With this lesson plan the student is provided with the definitions of what the methods are and how to use them when multiplying vectors. I also made sure to provide written examples and ways to utilize python as an extra source of instruction for getting the dot and cross product. Seeing that we can utilize computer programming in math is always nice to see because it can make the process of solving a problem quicker for you and to check your answer. Using Numpy is a really quick way for students to be able to compute the dot product and cross product if they are needing to check their work when solving these problems. Also it's always interesting to see the worked out math and the computer programming side by side to see the two different ways of doing a problem.

IV. **References**

[1] Admin. “Cross Product (Vector Product) - Definition, Formula and Properties.” *BYJUS*, BYJU'S, 1 Sep. 2022, https://byjus.com/maths/cross-product/.

[2] “Dot Product - Formula, Examples: Dot Product of Vectors.” *Cuemath*, https://www.cuemath.com/algebra/dot-product/.

[3] “NumPy Cross.” *Numpy.cross - NumPy v1.23 Manual*, https://numpy.org/doc/stable/reference/generated/numpy.cross.html.

[4] “NumPy Dot.” *Numpy.dot - NumPy v1.23 Manual*, https://numpy.org/doc/stable/reference/generated/numpy.dot.html#numpy.dot.

[5] “NumPy Vdot .” *Numpy.vdot - NumPy v1.23 Manual*, https://numpy.org/doc/stable/reference/generated/numpy.vdot.html.