Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 10

Linear Transformation

Submitted by:
Joaquin, Marc Christopher C.

Instructor:
Engr. Dylan Josh D. Lopez

January 01, 2021

I. Objectives

This laboratory activity aims to implement the principles and techniques of linear transformation. In addition, its goal by the end of this laboratory is to make programmer familiar with the role of matrix operations and visualize it, also to justify the precedence of matrix operation through Python

II. Methods

The practices of the activity is how to maximize more the matplotlib, especially in this topic, which is linear transformation. In this activity, it implies how to use Repositioning/Transalation, Shear, Scaling and Rotation in 2D. Also. it teaches the programmer how to transform it in 3D with the same activity in 2D. With this, the programmer nutured his skills with the basic concepts for coding, especially in matplotlib, such as *np.array()* function, where all the elements are in there [1]. *np.arrange()* function played a role in programmers code for it gives values for a span [2]. The *scatter plot* makes a scatter plot [3]. This is one of the functions that the programmer usually used and he actually used it last Laboratory Activity, in Vector Product. Moreover, *np.meshgrid()* function helps the programmer to achieve the objective so he can make the coordinate matrices return to coordinate vectors [4]. *Np.eye()* function make a 2D array return with ones on the diagonal and the rest is zero [5]. *Plt.grid()* function makes a grid line [6] and *plt.show()* function will display the output [7]. Aside for these functions, he used *np.deg2rad()* function in order to convert angles from degrees to radians [8]. At last are the *np.cos() function* and the *np.sin() function, np.cos() function* is for cosine element-wise [9] and *np.sin() function* is for sine element-wise [10].

III. Results

```
#Scatterplot view
def plot_scatter(x,t_mat=np.eye(2)):
    x_prime = x @ t_mat

    R = np.arange(-20,20,2)
    c1, c2 = np.meshgrid(R,R)
    spanRx = c1*x_prime[0][0] + c2*x_prime[1][0]
    spanRy = c1*x_prime[0][1] + c2*x_prime[1][1]
    plt.scatter(spanRx ,spanRy, s=5, alpha=0.75)

    plt.axhline(y=0, color='k')
    plt.axvline(x=0, color='k')
    plt.grid()
    plt.show()
```

Figure 1: Function for scatterplot

In the figure above, this is the programmer's code he used to obtain the goal of the supplementary activity. He remembered how the scatter plot used in Python for he already used it for the last Laboratory, which is Vector Product. He used it again it here in this laboratory exercise such as *np.arange()* function, *meshgrid()* function, *spans* and many more.

```
#Rotation
def rot_matrix(theta):
    theta = np.deg2rad(theta)
    rot_mat = np.array([
        [np.cos(theta), -np.sin(theta)],
        [np.sin(theta), np.cos(theta)]
    ])
    return rot_mat
```

Figure 2: Function for Rotation

In the next figure, it shows the code for rotation. The programmer used the code given from the activity example in the laboratory exercise. With this function, it will help the programmer to rotate the given matrix.

```
scatter = np.eye(2)
plot_scatter(scatter)

rotate = rot_matrix(45)
plot_scatter(scatter@rotate)
```

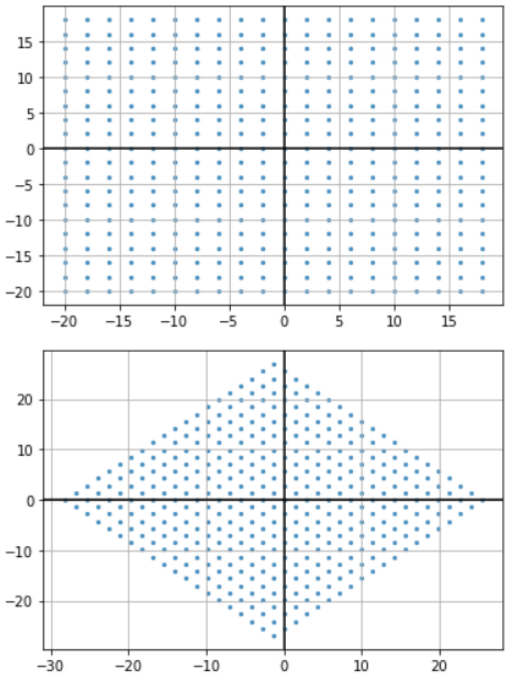

Figure 3: Output for Scatterplot View and Rotation Function

The programmer used his two functions in order to check and to verify if his functions are working well. When he gave a value “scatter”, he used it to his first function, *plot_scatter(*scatter), and the scatter plot view popped out to the result. However, in order to rotate the given matrix by the programmer, he used his second function and the 2nd output showed off.

IV. Conclusion

The programmer achieved the objective of this laboratory exercises. He practiced the usage of every functions that have been discussed to that activity. Moreover, he used also the past codes given from the past laboratory exercise, which makes him understand more how those codes work well in a code. Even though the goal of the supplementary activity there is to do at least one example for scatterplot view using spans, the programmer try to use Rotation too just to try it and and to understand how it works. With this activity, the programmer got familiarized and visualized the output much better with this exercise. In mechanics, it is very useful for the topics are more on matrices and tensors. Tensors are the multiple-dimensional arrays which it can help by this excercises [11]. However it can used in a cantilever beam or annular cylinder to verify internal pressure, via linear transformation [12].

MY LINK

https://github.com/chrisjoaquin29/Laboratory-Activity-10.git

References

[1] *numpy.array — NumPy v1.19 Manual*. (n.d.). Retrieved January 01, 2021, from https://numpy.org/doc/stable/reference/generated/numpy.array.html

[2] *numpy.arange — NumPy v1.19 Manual*. (n.d.). Retrieved January 01, 2021, from https://numpy.org/doc/stable/reference/generated/numpy.arange.html

[3] *Python Machine Learning Scatter Plot*. (n.d.). Retrieved January 01, 2021, from https://www.w3schools.com/python/python_ml_scatterplot.asp

[4] *numpy.meshgrid — NumPy v1.19 Manual*. (n.d.). Retrieved January 1, 2021, from https://numpy.org/doc/stable/reference/generated/numpy.meshgrid.html

[5] *numpy.eye — NumPy v1.19 Manual*. (n.d.). Retrieved January 1, 2021, from https://numpy.org/doc/stable/reference/generated/numpy.eye.html

[6] *matplotlib.pyplot.grid — Matplotlib 3.1.2 documentation*. (n.d.). Retrieved January 1, 2021, from https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.grid.html

[7] *matplotlib.pyplot.show — Matplotlib 3.1.2 documentation*. (n.d.). Retrieved January 1, 2021, from https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.show.html

[8] *numpy.deg2rad — NumPy v1.19 Manual*. (n.d.). Retrieved January 1, 2021, from https://numpy.org/doc/stable/reference/generated/numpy.deg2rad.html

[9] *numpy.cos — NumPy v1.19 Manual*. (n.d.). Retrieved January 1, 2021, from https://numpy.org/doc/stable/reference/generated/numpy.cos.html

[10] *numpy.sin — NumPy v1.19 Manual*. (n.d.). Retrieved January 1, 2021, from https://numpy.org/doc/stable/reference/generated/numpy.sin.html

[11] *Introduction to Tensors | TensorFlow Core*. (n.d.). Retrieved January 1, 2021, from https://www.tensorflow.org/guide/tensor

[12] Saravanan, U. (2013). *Advanced Solid Mechanics*.