Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 10

Linear Transformation

Submitted by:
Chipongian, John Patrick Ryan J.

Instructor:
Engr. Dylan Josh D. Lopez

January 03, 2021

I. Objectives

This laboratory activity aims to implement the principles and techniques in applying linear transformations. By being able to familiarize the roles of matrix operations, we will be able to visualize matrix operations.

II. Methods

In doing this laboratory activity the methods of numpy and matplotlib modules are used in completing the given problem in this activity. The function np.arange() returns an evenly spaced values given in its arguments [1]. Function np.meshgrid() is used to return the coordinate vector values into coordinate matrix values [2]. Function plt.scatter() is used to plot two same lengths of matrix which are the x axis and y axis [3]. Two functions were used to create arrays with values this are the np.array which are the standard function in creating arrays and the second is the np.eye which creates an array which would form a diagonal form of 1 and the other values outside that diagonal would consist of 0 [4]. The last set of functions are used in computation of the angles. This functions are np.deg2rad, which converts angles from degrees into radians [5], np,cos(), which finds the cosine of the array element-wise [6], np.sine(), which finds the sine of the array element-wise [7].

III. Results

```
def scatter_plot(x,t_mat=np.eye(2)):
    x_prime = x @ t_mat
    R = np.arange(-10,20,2)
    c1, c2 = np.meshgrid(R,R)
    spanRx = c1*x_prime[0][0] + c2*x_prime[1][0]
    spanRy = c1*x_prime[1][0] + c2*x_prime[1][1]
    plt.scatter(spanRx,spanRy)

    plt.grid()
    plt.show()
```

Figure 1

Figure 1 shows the function used in creating and showing the scatter plot. This function is to visualize the values of the plots in a grid.

```
matA = np.eye(2)
scatter_plot(matA)
```

Figure 2

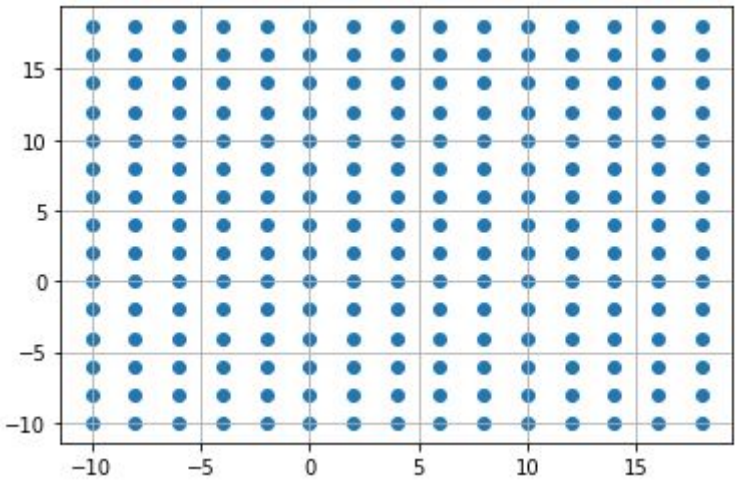

Figure 3

A new matrix is first created using the function np.eye() which would be used to implement in the created function earlier which is the scatter_plot(). The codes is shown in figure 2 and the result of it is shown in figure 3.

```
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 4

```
rotate = rot_matrix(45)
scale = 3*np.eye(2)
scatter_plot(matA@rotate)
scatter_plot(scale@rotate@matA)
```

Figure 5

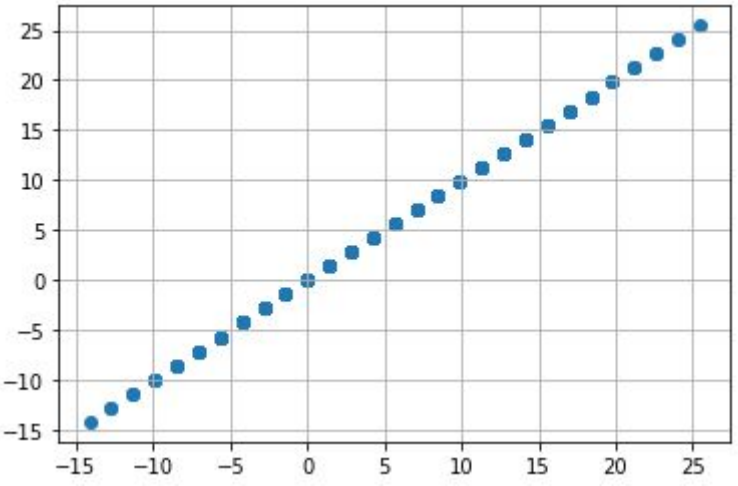

Figure 6

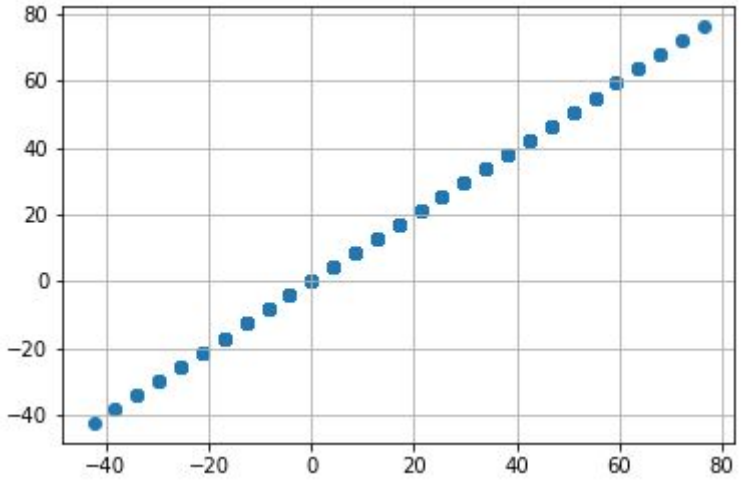

Figure 7

The function rot_matrix is created which its aim is to rotate the matrix that was plotted in the earlier on the activity. The function rot_matrix and its codes is shown in figure 4. In figure 5, the rotate function dictates the angle that the rot_matrix is going to do to the matrix. The scale function scales the function to become bigger or smaller. This is implemented by finding the inner product of the matrix, rotate, and scale function.

IV. Conclusion

Visualizing matrix operations becomes easier by utilizing different linear transformations such as translation, shears, scaling, and rotation of any matrices. This is all done by computing the inner product of the values of the matrices.

Linear transformation can be implemented in mechanics in showing different mechanic models. This can be done by showing different scales of different joints of the machine to deeply visualize how the joint should work and how it should affect the machine.

References

[1] Numpy.org. 2021. *Numpy.Arange — Numpy V1.19 Manual*. [online] Available at: <https://numpy.org/doc/stable/reference/generated/numpy.arange.html> [Accessed 2 January 2021].

[2] Numpy.org. 2021. *Numpy.Meshgrid — Numpy V1.19 Manual*. [online] Available at: <https://numpy.org/doc/stable/reference/generated/numpy.meshgrid.html> [Accessed 2 January 2021].

[3] W3schools.com. 2021. *Matplotlib Scatter*. [online] Available at: <https://www.w3schools.com/python/matplotlib_scatter.asp> [Accessed 2 January 2021].

[4] Numpy.org. 2021. *Numpy.Eye — Numpy V1.19 Manual*. [online] Available at: <https://numpy.org/doc/stable/reference/generated/numpy.eye.html> [Accessed 2 January 2021].

[5] Numpy.org. 2021. *Numpy.Deg2rad — Numpy V1.19 Manual*. [online] Available at: <https://numpy.org/doc/stable/reference/generated/numpy.deg2rad.html> [Accessed 2 January 2021].

[6] Numpy.org. 2021. *Numpy.Cos — Numpy V1.19 Manual*. [online] Available at: <https://numpy.org/doc/stable/reference/generated/numpy.cos.html> [Accessed 2 January 2021].

[7] Numpy.org. 2021. *Numpy.Sin — Numpy V1.19 Manual*. [online] Available at: <https://numpy.org/doc/stable/reference/generated/numpy.sin.html> [Accessed 2 January 2021].