Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 10

Linear Transformations

Submitted by:

Vallarta, Troy Joaquin G.

Instructor:

Engr. Dylan Josh D. Lopez

December 19, 2020

I. Objectives

This laboratory activity aims to implement linear transformation principles and techniques both in two and three-dimensional matrices. The laboratory activity's objective is to teach the researchers about repositioning, shearing, scaling, and rotating a matrix.

II. Methods

- This laboratory activity's practices are to understand how to transform the matrices in the researcher's needs.
  - The laboratory activity implies to teach the in-depth student learning about matrix transformation both in two and three-dimension.
- The laboratory activity provides the researcher a method and process in manipulating a specific matrix. In this laboratory activity, it is an introduction to eigenvector and eigenvalue.
  - The researcher achieves this laboratory activity by using Jupyter and using the MatPlotLib and NumPy libraries.

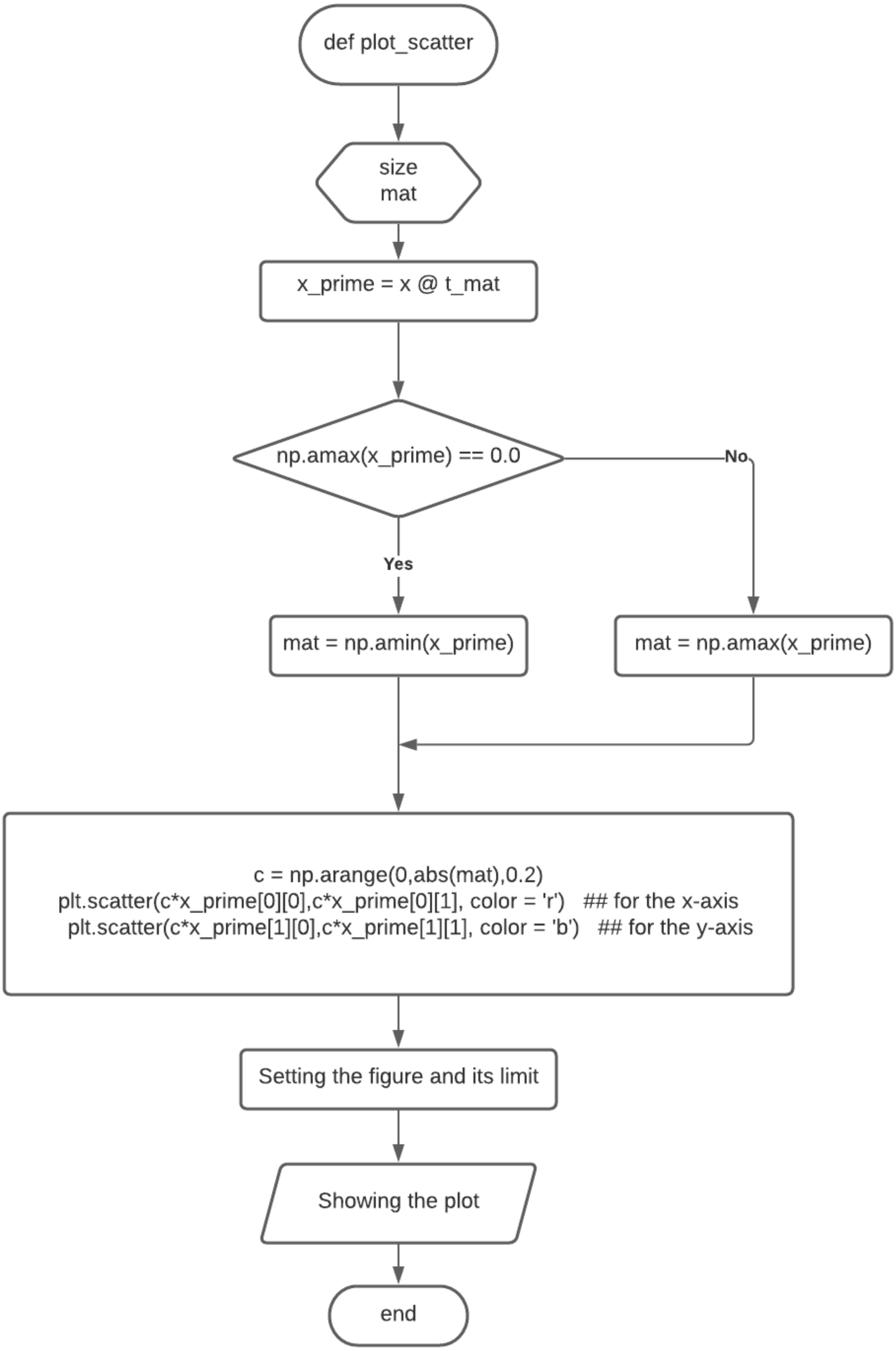

Figure 1. Flowchart for The plot_scatter() function

Figure 2 shows the created flowchart in creating the linear transformation function using span and scatter plots. In this laboratory activity, it combines all the past activity in linear algebra. There are methods used in this laboratory activity that the laboratory three implemented. Its reason is to teach the researcher to combine all the learnings in linear algebra and implement it in the problems shown.

```
def plot_scatter(x,t_mat=np.eye(2)):
    x_prime = x @ t_mat
    size= (2,2)
    plt.figure(figsize=(4,4))


    mat = None

    if np.amax(x_prime) == 0.0:
        mat = np.amin(x_prime)
    else:
        mat = np.amax(x_prime)                ## to get the size of the arrow

    c = np.arange(0,abs(mat),0.2)            ## using abs() function to find the size of the vector

    plt.scatter(c*x_prime[0][0],c*x_prime[0][1], color = 'r')   ## for the x-axis
    plt.scatter(c*x_prime[1][0],c*x_prime[1][1], color = 'b')   ## for the y-axis

    plt.xlim(-size[0],size[0])
    plt.ylim(-size[1],size[1])
    plt.xticks(np.arange((-size[0]), size[0]+1, 1.0))
    plt.yticks(np.arange((-size[1]), size[1]+1, 1.0))

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

Figure 2. Using scatter linear plot transformation.

Figure 2 shows the created function in creating linear transformation using spans using scatter plot. In this function, it shows how to create a linear transformation that uses a span. It copies the laboratory activity's function plot_quiv() function. The researcher removes the quiver function and replaces it with scatter to remove the line. The researcher added an argument to get the vector's end line using the np.amax() and np.amin() function.

III. Results

```
A = np.array([
    [1, 0],
    [0, 1]
])



t_mat = np.array([
    [-1,0],
    [0,-1]
])

shear = np.array([
    [-1,1],
    [1,1]
])
```

Figure 3. Initialization of matrix

Figure 3 shows the researcher's created matrices to check if the function created can make a linear transformation using spans and scatter plots. In this case, the researcher checks for the repositioning and shearing of a matrix. The A vector is an identity matrix used to check the matrices, if possible, for a dot product. The two matrices below shown in Figure 3 is the matrix transformation, the first one is the repositioning of a matrix, and the last one is shearing a matrix.

```
plot_scatter(A, t_mat)
```

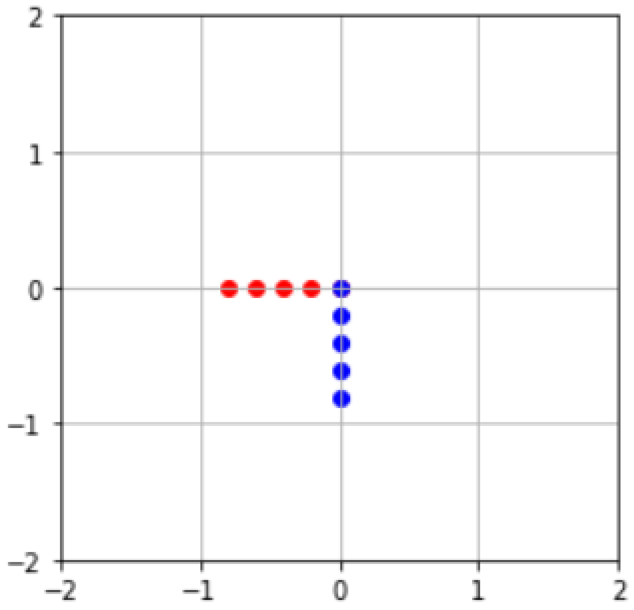

Figure 4. Repositioning a Matrix

```
plot_scatter(A, shear)
```

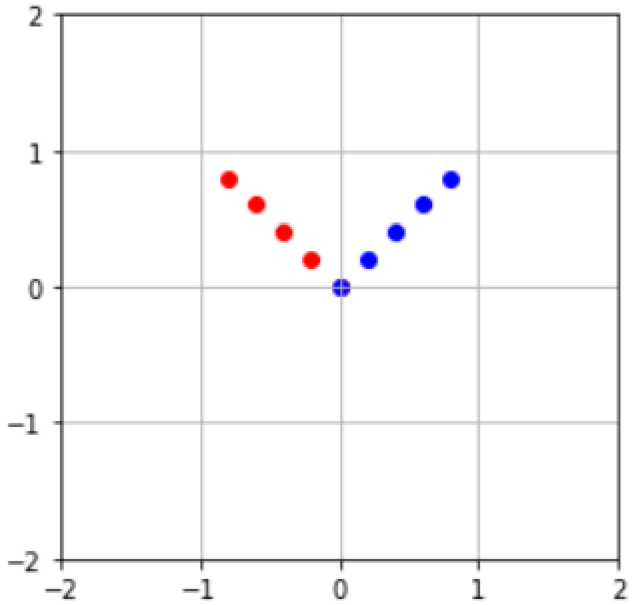

Figure 4.1 Shearing a matrix

Figure 4 shows that the created function is a success in creating a repositioning of a matrix. Figure 4.1 also shows that the created function is successful in creating a shear of a matrix. In this laboratory activity, the researchers created a function that can create a span and scatter plot, as seen in the figures above.

IV. Conclusion

The laboratory activity is useful in creating the needed matrix transformation of a particular matrix. The use of matrix transformation in two and three-dimension is essential in visualizing the matrices when transformed. Our day-to-day activity is most common when adjusting the pictures in photoshop or any platform that has a picture. We can rotate the pictures resulting in rotation transformation. Making the picture large or small is a scaling transformation; adjusting one side of the picture is the shearing transformation. Lastly, mirroring or flipping the picture is the repositioning transformation. There are many ways to implement matrix transformation. Linear transformation can also be implemented in mecahnics[1], such as in force. The use of linear transformation in mechanics has many uses, like applying Chetaev's postulates. A linear transformation also implements the theorem of holonomic mechanics.

References

[1] S. K. Soltakhanov, M. P. Yushkov, and S. A. Zegzhda, "Linear Transformation Of Forces," *SpringerLink,* 01-Jan-1970. [Online]. Available: https://doi.org/10.1007/978-3-540-85847-8_3. [Accessed: 19-Dec-2020].

Index

Github repo: https://github.com/Zofserif/Linear-Algebra/blob/master/Lab10/LinAlg%20Lab%2010.ipynb