Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 7

Matrix Algebra

Submitted by:

Surio, Rovil Jr., M.

Instructor:

Engr. Dylan Josh D. Lopez

December 30, 2020

I. Objectives

This laboratory activity aims to implement the principles and techniques of matrix operations. The student will be familiarized with the fundamental of matrix operations and learn how to apply them to solve equations.

II. Methods

The practices of the activity involve doing some operations between arrays and to be familiarized with it. This activity teaches and implies the techniques of some specific operations that are vital to this course such as transposition and dot product. Meanwhile, the deliverables of this activity are to perform operations and prove the dot product properties. It was achieved by making a code with Python that will prove that the property is True.

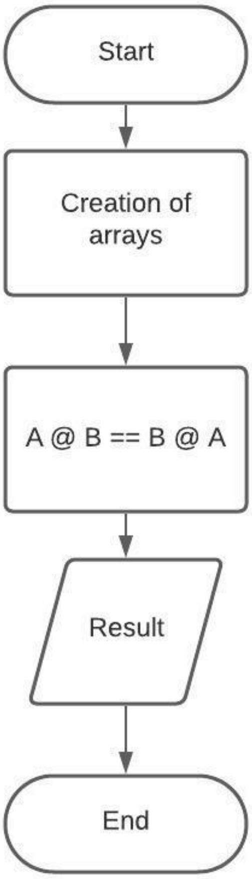


Figure 1 First property

Figure 1 shows the flowchart created for proving the first property of the dot product. This property states that a dot product between A and B is not equal to the dot product between B and A.

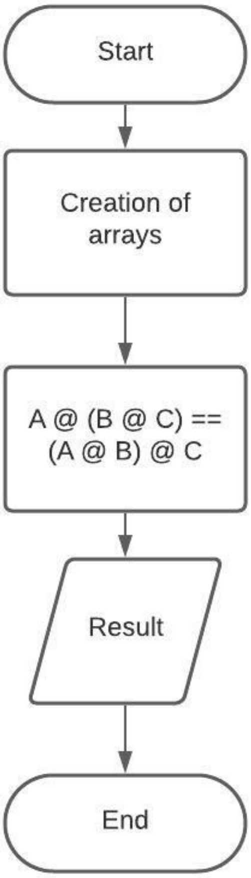


Figure 2 Second property

The second property of the dot product states that a dot product between matrix A and the result of the product between matrix B and C is equal to the dot product of matrix C between a dot product result between matrix A and B.

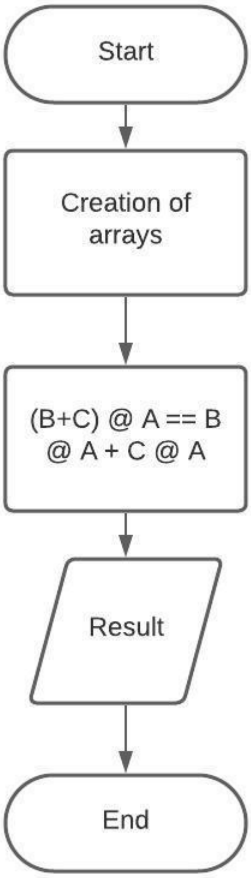


Figure 3 Fourth property

The fourth property of the dot product states that the dot product between A and the result from the addition between B and C is equal to the dot product between B and A added to the dot product between C and A.

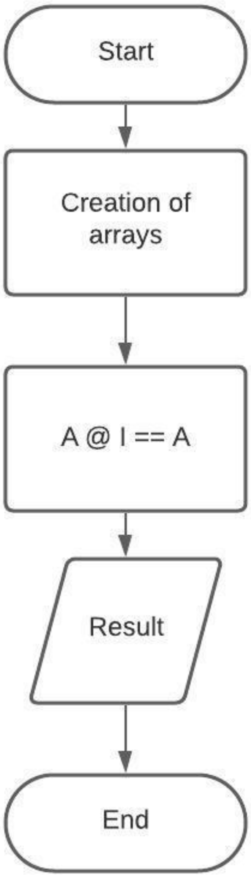


Figure 4 Fifth property

The fifth property of the dot product states that the dot product between a matrix and an identity matrix is equal to that matrix.

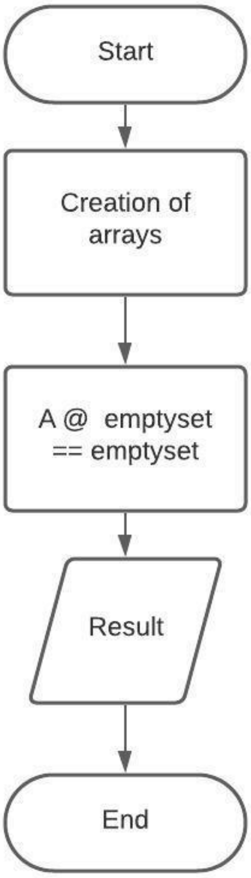


Figure 5 Sixth property

The sixth property of the dot product states that when a dot product between a matrix and a null set is performed, it will result in a null set or empty set.

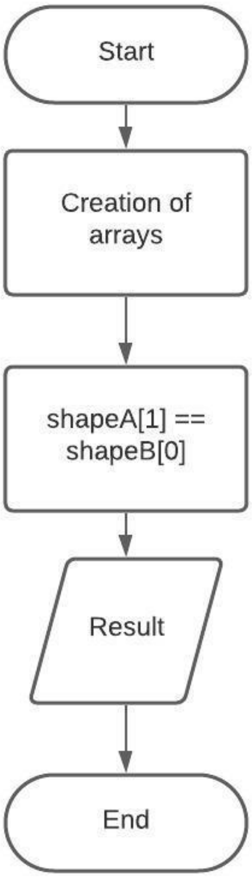
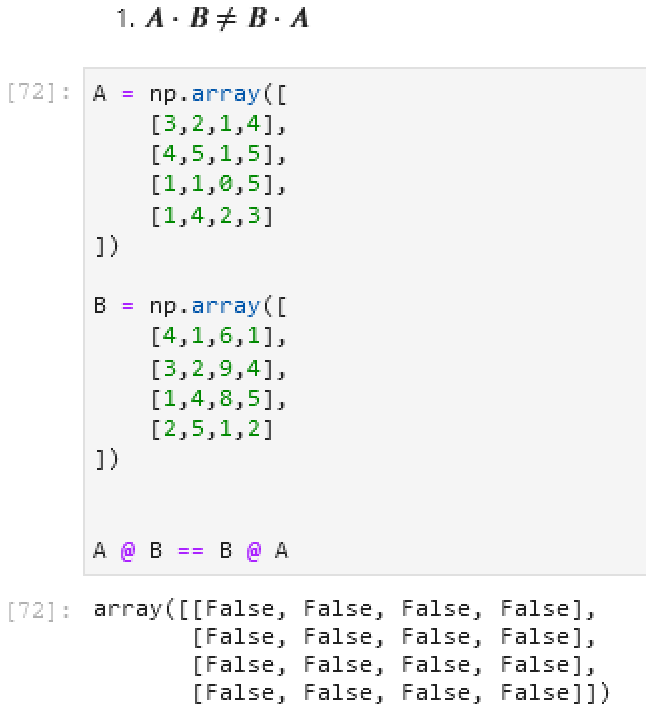


Figure 6 Seventh property

The seventh property of the dot product states that in A = (n, k) and B = (k, m), the column space size on A or (k) in A must equal to row space size on B or (k) in B to be able to perform dot product between matrices.

III. Results

The results from implementing the properties of the dot product with Python for proving it are presented in this section



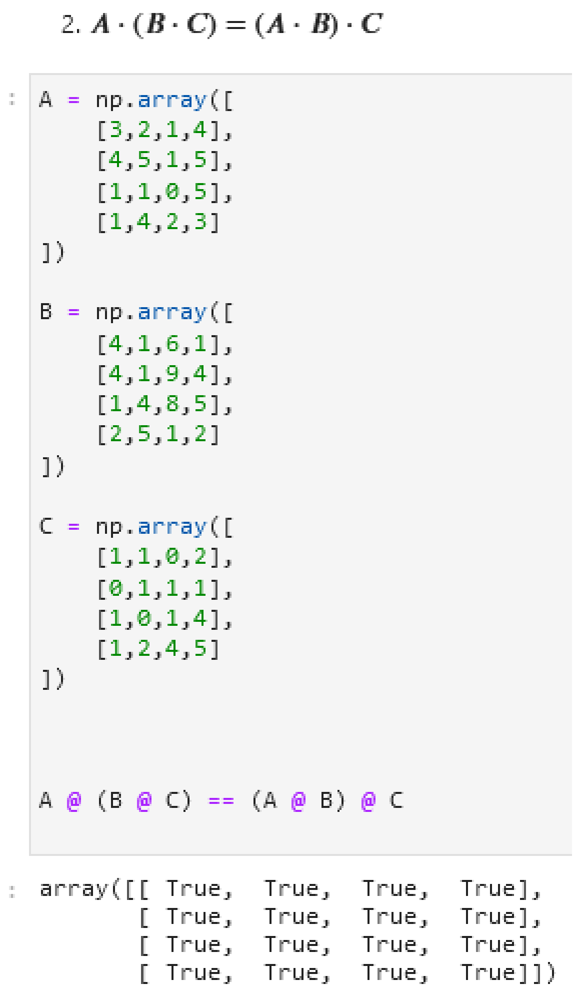
1. $\boldsymbol{A} \cdot \boldsymbol{B} \neq \boldsymbol{B} \cdot \boldsymbol{A}$

```
[72]: A = np.array([
          [3,2,1,4],
          [4,5,1,5],
          [1,1,0,5],
          [1,4,2,3]
      ])

      B = np.array([
          [4,1,6,1],
          [3,2,9,4],
          [1,4,8,5],
          [2,5,1,2]
      ])


      A @ B == B @ A
```

```
[72]: array([[False, False, False, False],
             [False, False, False, False],
             [False, False, False, False],
             [False, False, False, False]])
```

Figure 7 First property

Figure 7 shows that getting the equivalence of the two matrices that performed based on the first property results in False which proves that the property is right.



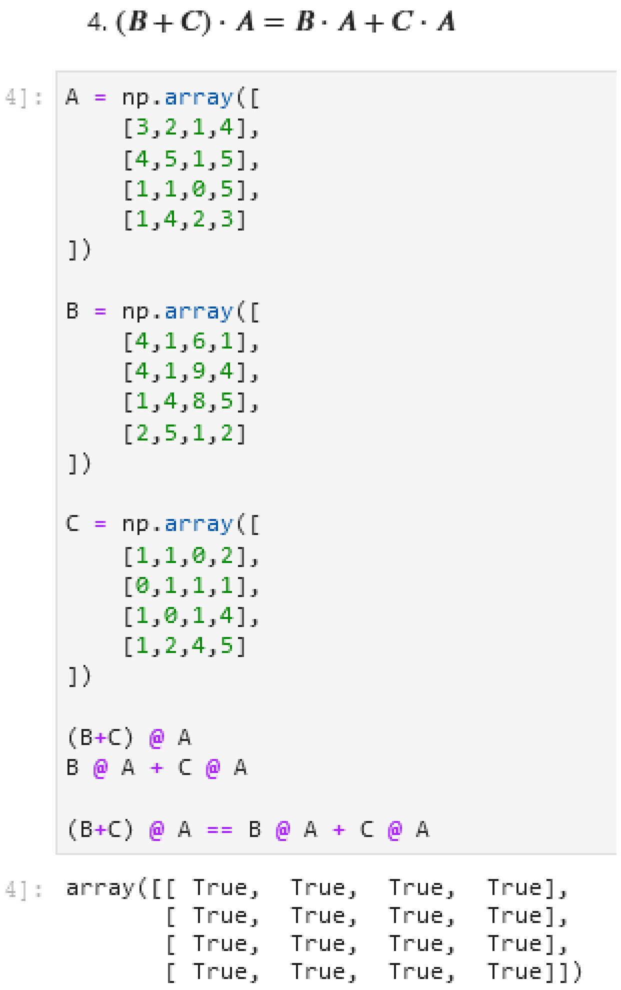
2. $\boldsymbol{A} \cdot (\boldsymbol{B} \cdot \boldsymbol{C}) = (\boldsymbol{A} \cdot \boldsymbol{B}) \cdot \boldsymbol{C}$

```
A = np.array([
    [3,2,1,4],
    [4,5,1,5],
    [1,1,0,5],
    [1,4,2,3]
])

B = np.array([
    [4,1,6,1],
    [4,1,9,4],
    [1,4,8,5],
    [2,5,1,2]
])

C = np.array([
    [1,1,0,2],
    [0,1,1,1],
    [1,0,1,4],
    [1,2,4,5]
])



A @ (B @ C) == (A @ B) @ C
```

```
array([[ True,  True,  True,  True],
       [ True,  True,  True,  True],
       [ True,  True,  True,  True],
       [ True,  True,  True,  True]])
```

Figure 8 Second property

Figure 8 shows that getting the equivalence of the two matrices that performed based on the second property results in True which proves that the property is right.



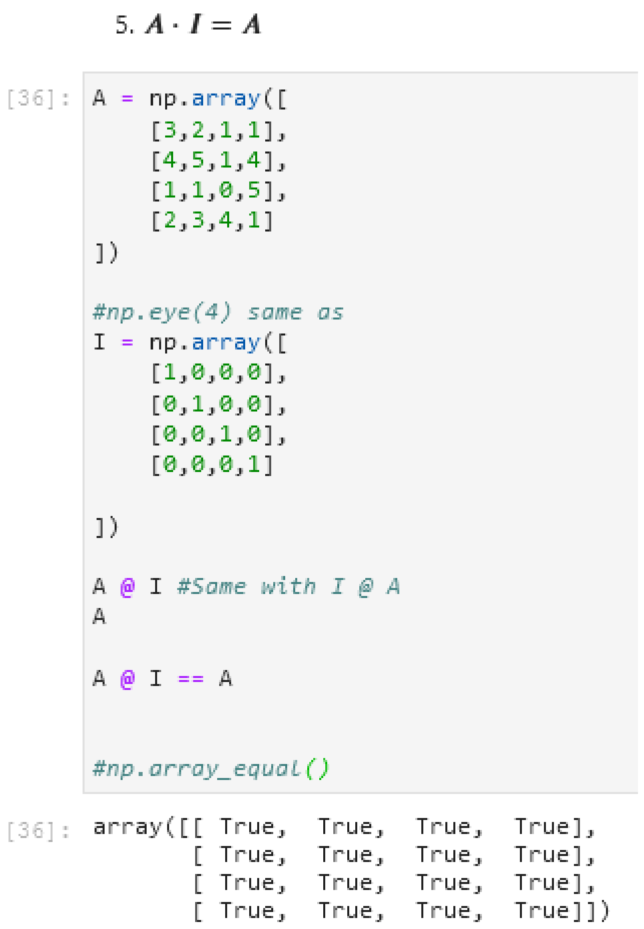
4. $(B+C) \cdot A = B \cdot A + C \cdot A$

```
4]: A = np.array([
        [3,2,1,4],
        [4,5,1,5],
        [1,1,0,5],
        [1,4,2,3]
    ])

    B = np.array([
        [4,1,6,1],
        [4,1,9,4],
        [1,4,8,5],
        [2,5,1,2]
    ])

    C = np.array([
        [1,1,0,2],
        [0,1,1,1],
        [1,0,1,4],
        [1,2,4,5]
    ])

    (B+C) @ A
    B @ A + C @ A

    (B+C) @ A == B @ A + C @ A
```

```
4]: array([[ True,  True,  True,  True],
           [ True,  True,  True,  True],
           [ True,  True,  True,  True],
           [ True,  True,  True,  True]])
```

Figure 9 Fourth property

Figure 9 shows the results from performing an equivalence test based on what the fourth property stated, and it proves that this property is right.



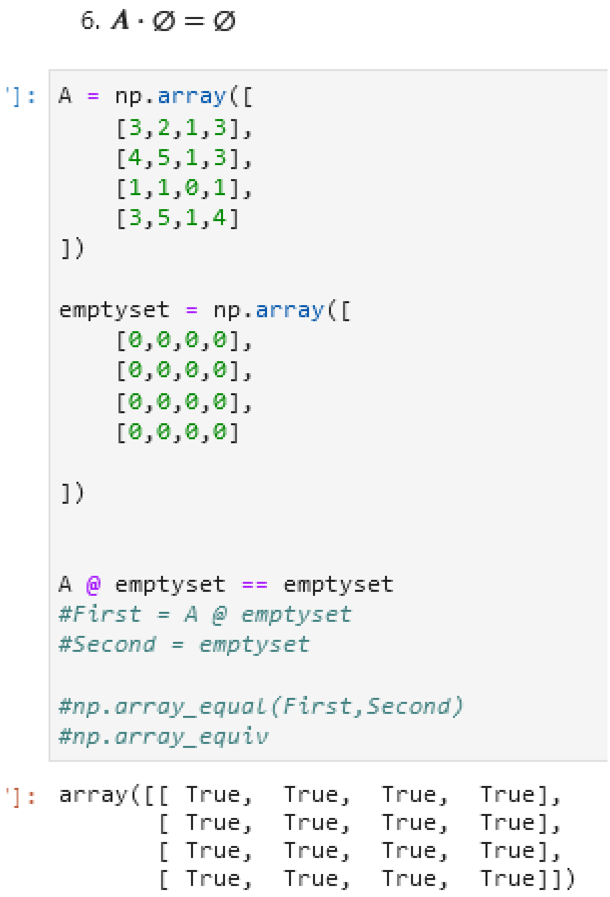
5. $A \cdot I = A$

```
[36]: A = np.array([
          [3,2,1,1],
          [4,5,1,4],
          [1,1,0,5],
          [2,3,4,1]
      ])

      #np.eye(4) same as
      I = np.array([
          [1,0,0,0],
          [0,1,0,0],
          [0,0,1,0],
          [0,0,0,1]

      ])

      A @ I #Same with I @ A
      A

      A @ I == A


      #np.array_equal()
```

```
[36]: array([[ True,  True,  True,  True],
             [ True,  True,  True,  True],
             [ True,  True,  True,  True],
             [ True,  True,  True,  True]])
```

Figure 10 Fifth property

Figure 10 shows the proof that the dot product between an array and an identity array results in the array itself.



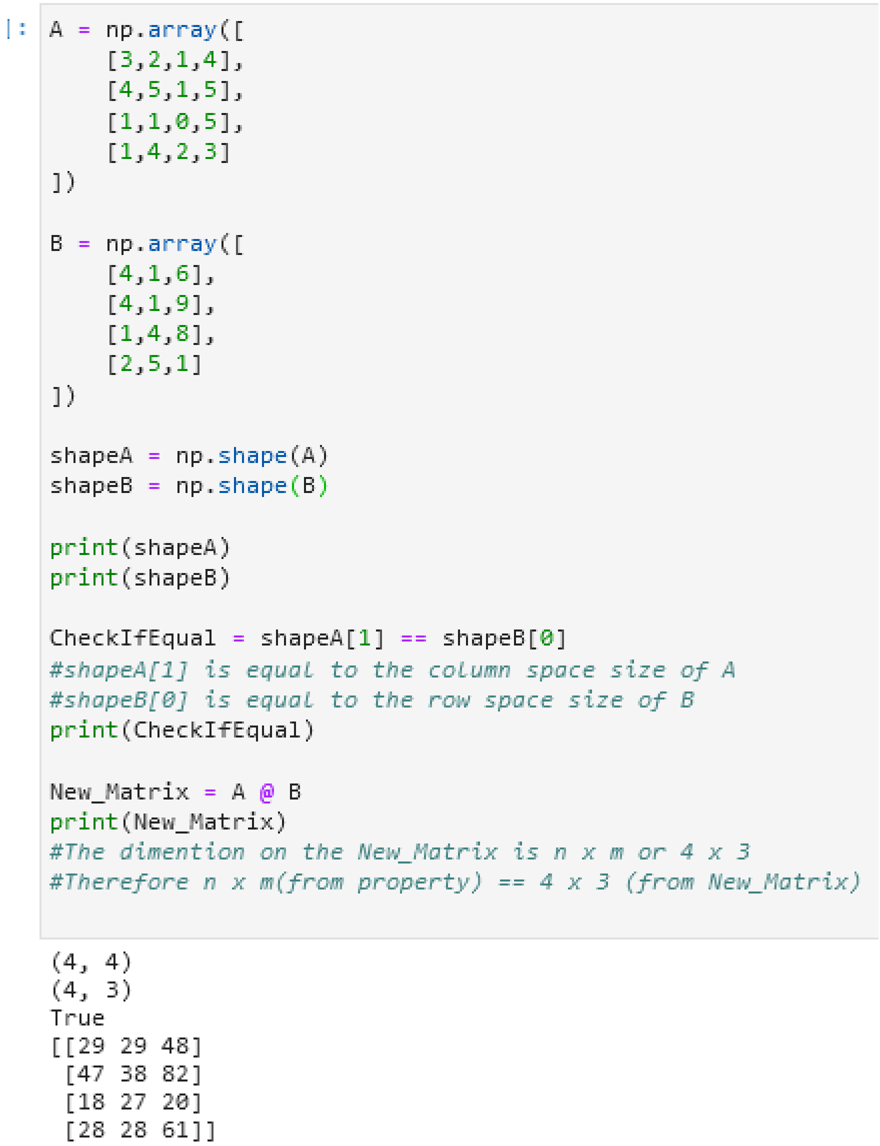
6. $\boldsymbol{A} \cdot \varnothing = \varnothing$

```
']: A = np.array([
        [3,2,1,3],
        [4,5,1,3],
        [1,1,0,1],
        [3,5,1,4]
    ])

    emptyset = np.array([
        [0,0,0,0],
        [0,0,0,0],
        [0,0,0,0],
        [0,0,0,0]

    ])


    A @ emptyset == emptyset
    #First = A @ emptyset
    #Second = emptyset

    #np.array_equal(First,Second)
    #np.array_equiv
```

```
']: array([[ True,  True,  True,  True],
           [ True,  True,  True,  True],
           [ True,  True,  True,  True],
           [ True,  True,  True,  True]])
```

Figure 11 Sixth property

Figure 11 shows the results from performing an equivalence test based on what the sixth property stated, and it proves that this property is right.



```python
A = np.array([
    [3,2,1,4],
    [4,5,1,5],
    [1,1,0,5],
    [1,4,2,3]
])

B = np.array([
    [4,1,6],
    [4,1,9],
    [1,4,8],
    [2,5,1]
])

shapeA = np.shape(A)
shapeB = np.shape(B)

print(shapeA)
print(shapeB)

CheckIfEqual = shapeA[1] == shapeB[0]
#shapeA[1] is equal to the column space size of A
#shapeB[0] is equal to the row space size of B
print(CheckIfEqual)

New_Matrix = A @ B
print(New_Matrix)
#The dimention on the New_Matrix is n x m or 4 x 3
#Therefore n x m(from property) == 4 x 3 (from New_Matrix)
```

```
(4, 4)
(4, 3)
True
[[29 29 48]
 [47 38 82]
 [18 27 20]
 [28 28 61]]
```

Figure 12 Seventh property

Figure 12 shows the proof of the dimension property. The shape of both matrices is shown first and since the column of the first matrix is equal to the number of rows in the second matrix, the dot product between the two matrices is successful.

IV. Conclusion

This laboratory successfully informed me about the implementation and the techniques of performing matrix operations such as transposition, dot product, inverse, and determinant. I also was able to familiarize myself with the different special properties of the dot product that become an activity for this laboratory. Moreover, I learned how to do transposition, dot product, getting the determinant and the inverse of the matrix using Python and NumPy library. Learning all of this in this laboratory are my precious achievement. Lastly, matrix operations can solve problems in health care through the technology created with matrices such as medical imaging. According to [1], the problem of image reconstruction from projections can be considered as a system of linear equations of the form:

$$Ax \approx b,$$

Figure 13 Equation

It was explained thoroughly in [1] how it works but the main idea of this is that the system matrix A in Figure 13 simulates CT operation and the elements of this depend on the projection number and the angle. The values on the column of the matrix represent the intensities of the image and the column matrix b represents the collected projections by the scanner.

References

[1]L. Flores, V. Vidal, and G. Verdú, "System Matrix Analysis for Computed Tomography Imaging", *plos.org*, 2015. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0143202. [Accessed: 12- Dec- 2020].

Appendix

This is the GitHub Repository https://github.com/RovilSurioJr/Laboratory-7