Adamson University
College of Engineering
Computer Engineering Department

Linear Algebra

Laboratory Activity No. 6

Matrices

Submitted by:
Aquiro, Freddielyn E.

Instructor:
Engr. Dylan Josh D. Lopez

November 26, 2020

I. Objectives

This laboratory activity aims to implement the principles and techniques of perfoming the basic matrix operations and to familiarize the relation of matrices to linear equations. Also to translate matrix equations and operations using Python.

II. Methods

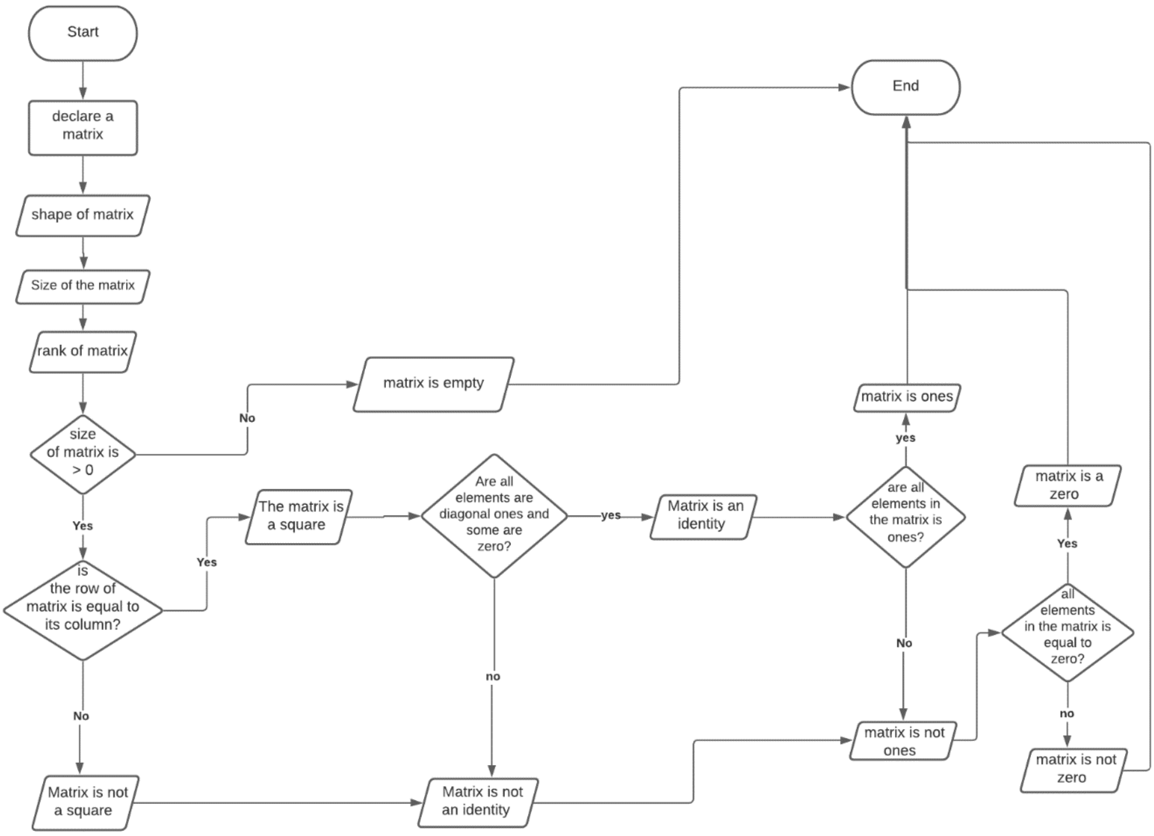

Figure 1: Flowchart for task 1

In figure 1 shown the flowchart for the task which to create a program that describes a matrix. The program should start in declaring its matrix then it should display the shape of the matrix, size of the matrix, and rank of the matrix. Then it will go over to some conditions that will determine if the matrix is empty, if not it will continue to determine if the matrix is a square it should have equal rows and columns followed by identifying if the matrix has a 1 in the diagonal section and 0 in it. It will display matrix is an identity. Then if all the elements in the matrix are 1 it will display the matrix as ones thus if the elements of the matrix are in zero it will display the matrix as zero.

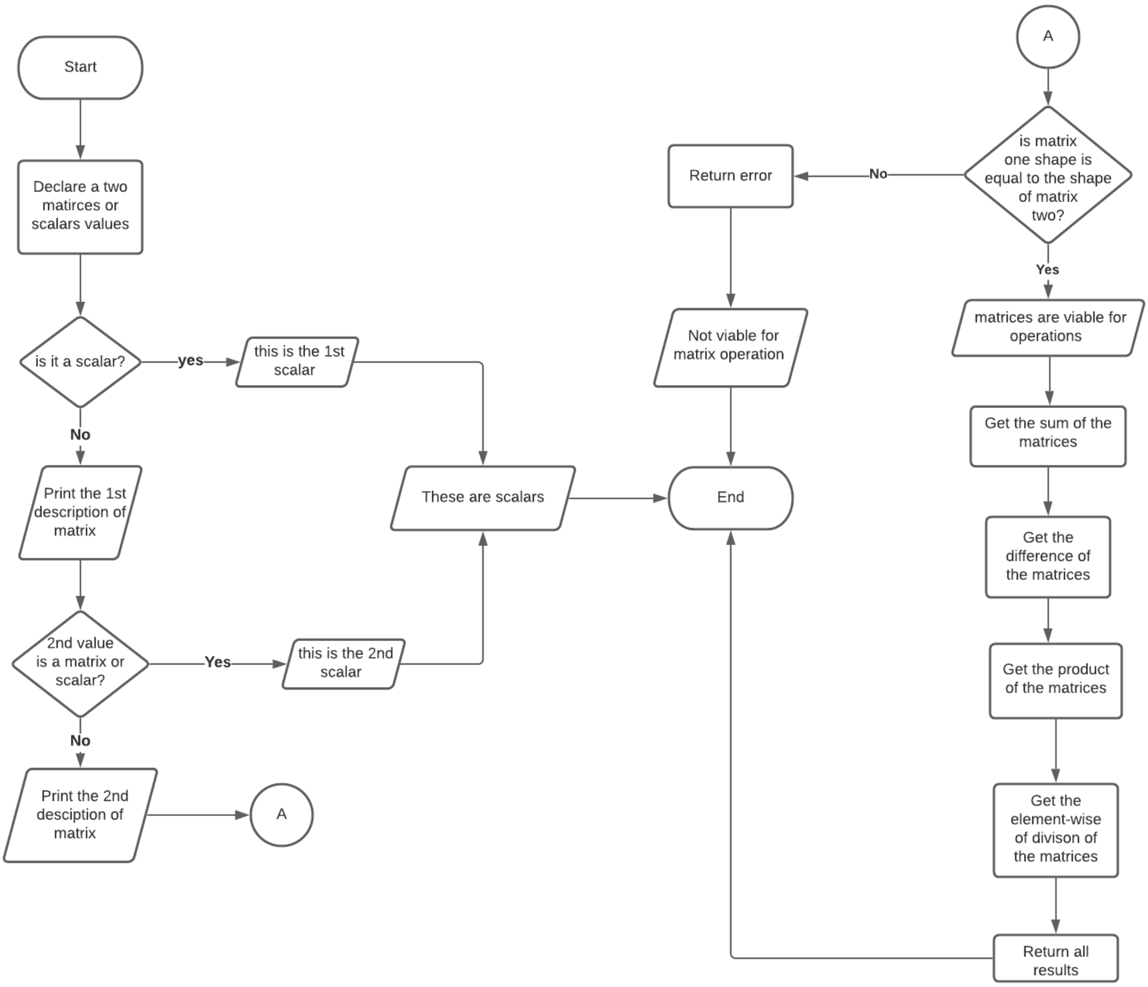

Figure 2: flowchart for task 2

In figure 2 shown the flowchart for task 2 which to create a program that determines its parameters, description of matrices, and matrix operations. The first is to declare the values of two matrices or scalars. Then it will determine if the parameters are a scalar if not it will display the first description of the matrix such as shape, size, and rank. Then for the second parameter if it's a matrix it will do the same procedure as in the first matrix. Then it will continue through the program conditions, if the two matrices have an equal shape it will proceed to the matrix operations which has and if the condition is not met the program will return to its error message.

The practices of this activity are to be familiar with basic operations and with their relations, as well as to translate the matrix equations. The deliverables of this activity are to create a code and function that will display the description of the matrix and perform its corresponding matrix operations. In order to achieve that NumPy functions are used in this activity such as np.array() to create an array [1] that represents the matrices or matrix. Next is .shape is used to identify the matrix shape [2], then to identify the size of the matrix np. prod().shape is used [3], to complete the description of the matrix .ndim is used to know the rank of the matrix. [4], np.allclose() was used to check if the matrix is equal to its row and columns [5] then to create a matrix that will identify if there is ones to its diagonal section and zeros values np.identity() was used [6], along with its functions np.all() is used to test all elements if it is equal to 1 or 0 to determine if it is a ones or zero matrices [7], np.isscalar is used to check if the element is a scalar [8], np.add() is used to get the sum of the two matrices [9], np.subtract() is used to subtract the two matrices [10], np.multiply() is used to multiply the two matrices [11], to get a quotient of the two matrices np.division() is used [12].

III. Results

```
# Function Area
def mat_desc(matrix):
   print(f'{matrix}\n\nShape:\t{matrix.shape}\nSize:\t{np.product(matrix.shape)}\nRank:\t{matrix.ndim}\n')
   if matrix.size > 0:
    is_square = True if matrix.shape[0] == matrix.shape[1] else False
    is_identity = True if matrix.shape[0] == matrix.shape[1] and np.allclose(matrix, np.identity(matrix.shape[0])) else False
    is_ones = True if np.all((matrix == 1)) else False
    is_zero = True if np.all((matrix == 0)) else False
    print(
        f'is it a square?: {is_square}\n'
        f'is it identity?: {is_identity}\n'
        f'Is is a ones: {is_ones}\n'
        f'Is it a zero: {is_zero}\n'
        )
   else:
     print("The Matrix is Empty")
```

Figure 3: Code snippet for function area for task 1.

```python
## Matrix declarations
mtrx_1 = np.array([
    [1,1,1],
    [1,1,1],
    [1,1,1]
])
mtrx_2= np.array([
   [1,0,0],
   [0,1,0],
   [0,0,1]
])
mtrx_3 = np.array([
    [0,0,0],
    [0,0,0],
    [0,0,0]
])
mtrx_4 = np.array([
         [],
         [],
         []
])

mtrx_5 = np.array([
  [1,2,3,4],
  [5,4,3,2],
  [7,8,9,6]
])
```

Figure 4: Code snippet for matrix declation in task 1

```python
#Printing the descriptions of the matrix
print("First Matrix:")
mat_desc(mtrx_1)
```

```python
print("Second Matrix:")
mat_desc(mtrx_2)
```

```python
print("Third Matrix:")
mat_desc(mtrx_3)
```

```python
print("Fourth Matrix:")
mat_desc(mtrx_4)
```

```
print("Fifth Matrix:")
mat_desc(mtrx_5)
```

Figure 5: code snippet for printing the desciptions of the matrix for task 1.

In figure 3 to 5 shown as the code snippet for task 1, the requirement is to display the description of the matrix and create 5 samples of the matrix. To describe the matrices, mat_operations() is used. Also, I used a one liner code or a short code to practice a good documentation.

```
First Matrix:
[[1 1 1]
 [1 1 1]
 [1 1 1]]

Shape:  (3, 3)
Size:   9
Rank:   2

is it a square?: True
is it identity?: False
Is is a ones: True
Is it a zero: False
```

Figure 6: Result of the first matrix for task 1.

In figure 6 shown the first matrix that has the shape of the matrix is 3 by 3 meaning it has equal rows and columns. Followed by its size which is described as 9, it comes from multiplying its shape which is a 3 and 3 that is equivalent to 9. For its rank ndim was used to know its rank which is 2 or a 2-dimensional array. Next, the first matrix is a square since it has an equal number of rows and columns. The first matrix is not an identity matrix even the matrix has ones in the diagonal section but it doesn't have zero values. Then it is shown that the first matrix is a ones since all elements are in one thus it is not a zero matrix since not all the elements are zero.

```
Second Matrix:
[[1 0 0]
 [0 1 0]
 [0 0 1]]

Shape:  (3, 3)
Size:   9
Rank:   2

is it a square?: True
is it identity?: True
Is is a ones: False
Is it a zero: False
```

Figure 7: Result of the second matrix for task 1.

In figure 7 shown for the second matrix. It has a shape of the matrix is 3 by 3 meaning it has equal rows and columns. Followed by its size which is described as 9, it comes from multiplying its shape which is a 3 and 3 that is equivalent to 9. For its rank ndim was used to know its rank which is 2 or a 2-dimensional array. Next, the matrix is a square since it has an equal number of rows and columns. The matrix is an identity matrix since the matrix has a ones in the diagonal section and it has zero values. Then it is shown that the matrix is not a ones since all elements are not in one thus it is not a zero matrix since not all the elements are zero.

```
Third Matrix:
[[0 0 0]
 [0 0 0]
 [0 0 0]]

Shape:  (3, 3)
Size:   9
Rank:   2

is it a square?: True
is it identity?: False
Is is a ones: False
Is it a zero: True
```

Figure 8: Reuslt of the third matrix for task 1.

In figure 8 shown for the third matrix. It has a shape of the matrix is 3 by 3 meaning it has equal rows and columns. Followed by its size which is described as 9, it comes from multiplying its shape which is a 3 and 3 that is equivalent to 9. For its rank ndim was used which is 2 or a 2-dimensional array. Next, the matrix is a square since the number of rows and columns are equal. Then the matrix is not an identity matrix even the matrix has zero values, it doesn't have ones values in the diagonal section. After that, it is shown that the matrix is not a ones since all elements are not in one thus it is a zero matrix since all elements are in zero.

```
Fourth Matrix:
[]

Shape:  (3, 0)
Size:   0
Rank:   2

The Matrix is Empty
```

Figure 9: Result of the fourth matrix for task 1.

In figure 9 shown for the fourth matrix. It has a shape of the matrix is 3 by 0 meaning it doesn't have an equal number of rows and columns. Followed by its size which is described as 0, it comes from multiplying its shape which is a 3 and 0 that is equivalent to 0. For its rank ndim was used which is 2 or a 2-dimensional array. Next, the matrix is a square since it has an equal number of rows and columns. Since the matrix has no value or is blank, it will print that the matrix is empty.

```
Fifth Matrix:
[[1 2 3 4]
 [5 4 3 2]
 [7 8 9 6]]

Shape:  (3, 4)
Size:   12
Rank:   2

is it a square?: False
is it identity?: False
Is is a ones: False
Is it a zero: False
```

Figure 10: Result of the last or fifth matrix.

In figure 10 shown is the last or fifth matrix for the problem. It has a shape of the matrix is 3 by 4 meaning it has doesn’t have the same number of rows and columns. Followed by its size which is described as 12, it comes from multiplying its shape which is a 3 and 4 that is equivalent to 12. For its rank ndim was used which is 2 or a 2-dimensional array. Next, the matrix is not a square since it doesn't have an equal number of rows and columns. Then the matrix is not an identity matrix it doesn’t have ones values in the diagonal section and zero. After that, it is shown that the matrix is not a ones since all elements are not one thus it is not a zero matrix since not all the elements are in zero.

```python
# Function Area
def mat_operations(mtrx1, mtrx2): # Function for operation of matrices
    alpha=10**-10
    if np.isscalar(mtrx1) == True and np.isscalar(mtrx2) == True:
      print(f'This is 1st scalar = {mtrx1}\n'
        f'This is 2nd scalar = {mtrx2}')
    else:
      print(f'The following description is the 1st matrix:\n'
          f'{mtrx1}\n\n'
          f'Shape : {mtrx1.shape}\n'
          f'Rank: {mtrx1.ndim}\n'
          f'Size: {np.product(mtrx1.shape)}\n')
      if mtrx1.size >0:
          is_square = True if mtrx1.shape[0] == mtrx1.shape[1] else False
          is_identity = True if mtrx1.shape[0] == mtrx1.shape[1] and np.allclose(mtrx1, np.identity(mtrx1.shape[0])) else False
          is_ones = True if np.all((mtrx1== 1)) else False
          is_zero = True if np.all((mtrx1 == 0)) else False
          print(
        f'is it a square matrix?: {is_square}\n'
        f'is it identity matrix?: {is_identity}\n'
        f'Is is a ones matrix?: {is_ones}\n'
        f'Is it a zero matrix?: {is_zero}\n'
        )

      print(f'This is a matrix, The following description is the 2nd matrix:\n'
        f'{mtrx2}\n\n'
        f'Shape : {mtrx2.shape}\n'
        f'Rank: {mtrx2.ndim}\n'
        f'Size: {np.product(mtrx2.shape)}\n')
      if mtrx2.size > 0:
          is_square = True if mtrx2.shape[0] == mtrx2.shape[1] else False
          is_identity = True if mtrx2.shape[0] == mtrx2.shape[1] and np.allclose(mtrx2, np.identity(mtrx2.shape[0])) else False
          is_ones = True if np.all((mtrx2== 1)) else False
          is_zero = True if np.all((mtrx2 == 0)) else False
          print(f'is it a square matrix?: {is_square}\n'
        f'is it identity matrix?: {is_identity}\n'
        f'Is is a ones matrix?: {is_ones}\n'
        f'Is it a zero matrix?: {is_zero}\n'
        )
      if (mtrx1.shape == mtrx2.shape):
          print("\nMatrices are viable for operation ")
          sum = "Sum of matrices",np.add(mtrx1,mtrx2)
          difference ="Difference of matrices",np.subtract(mtrx1,mtrx2)
          product ="Element-wise of multiplication",np.multiply(mtrx1,mtrx2)
          quotient ="Element-wise of division",np.divide(mtrx1,mtrx2+alpha)
          return sum,difference,product,quotient

      else:
         error = "Not viable for matrix operation"
         return error
```

Figure 11: Code snippet for task 2

```python
# Matrix declaration
mtrx_1= np.array([
   [4,3,2],
   [1,2,3],
   [7,8,9]
])
mtrx_2 = np.array([
    [1,1,1],
    [1,1,1],
    [1,1,1]
])
mtrx_3 = np.array([
    [1,0,0],
    [0,1,0],
    [0,0,1]
])
mtrx_4 = np.array([
    [0,0,0,0],
    [0,0,0,0],
    [0,0,0,0]
])

mtrx_5= np.array([
    [1,2,3],
    [5,4,3],
    [7,8,9]
])
```

```python
#Scalars
x = 1
y= 2
```

Figure 12: Code snippet for declaration of scalars and matrix for task 2

```python
# Printing Matrices
print('First Pair')
mat_operations(mtrx_1,mtrx_2)
```

```python
print('Second Pair')
mat_operations(mtrx_2,mtrx_4)
```

```python
print('Third Pair')
mat_operations(mtrx_3,mtrx_2)
```

```python
print('Fourth Pair')
mat_operations(mtrx_4,mtrx_5)
```

```python
print('Fifth Pair')
mat_operations(mtrx_5,mtrx_1)
```

```python
# Printing both scalars
print('These are scalar parameters')
mat_operations(x,y)
```

Figure 13: Code snippet for printing the result for task 2

In figure 11 to 13 shown as the code snippet for task 2. The requirement in this task is to create a function mat_operations which has two matrices or scalars values and its five example pairs which is not lower by (3,3) shape. Also, I used a one liner code or a short code to practice a good documentation. Then the condition is to display each matrix if the parameter is scalar it will print the scalar values. After that, it will determine if the values of the matrices are viable for the operations such as addition, subtraction, element-wise multiplication, and to get a specific answer for the element-wise of divison alpha is used to display a small value and to prevent getting infinite value. If the condition is not met it should return to its error message

```
First Pair
The following description is the 1st matrix:
[[4 3 2]
 [1 2 3]
 [7 8 9]]

Shape : (3, 3)
Rank: 2
Size: 9

is it a square matrix?: True
is it identity matrix?: False
Is is a ones matrix?: False
Is it a zero matrix?: False

This is a matrix, The following description is the 2nd matrix:
[[1 1 1]
 [1 1 1]
 [1 1 1]]

Shape : (3, 3)
Rank: 2
Size: 9

is it a square matrix?: True
is it identity matrix?: False
Is is a ones matrix?: True
Is it a zero matrix?: False


Matrices are viable for operation
(('Sum of matrices', array([[ 5,  4,  3],
         [ 2,  3,  4],
         [ 8,  9, 10]])), ('Difference of matrices', array([[3, 2, 1],
         [0, 1, 2],
         [6, 7, 8]])), ('Element-wise of multiplication', array([[4, 3, 2],
         [1, 2, 3],
         [7, 8, 9]])), ('Element-wise of division', array([[4., 3., 2.],
         [1., 2., 3.],
         [7., 8., 9.]])))
```

Figure 14: First pair of matrix for task 2

In figure 14 shown as the first pair of the matrices. The first matrix has a 3 by 3 shape meaning it has an equal number of rows and columns. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 9 because the shape of the matrix is multiplied by 3 and 3 which is equivalent to 9. The first matrix is a square since its rows and columns were equal. Next, it is not an identity matrix since it doesn't have ones values in the diagonal section as well as zero values. After that, the matrix is not ones since it doesn't have ones in the element thus it is not a zero matrix since all elements are not in zero. The second matrix is has a shape of 3 by 3 meaning it has a equal rows and columns. Then since it is a

2dimensional array the rank of the matrix is 2, followed by its size which is 9 because the shape of the matrix is multiplied by 3 and 3 which is equivalent to 9. The second matrix is square matrix since its rows and columns were equal. Next, it is not an identity matrix even it has ones values in the diagonal section, it doesn't have zero in it. After that, the matrix is determined as ones since all its elements are one thus it is not a zero matrix since all elements are not in zero. Since then it is both matrices it is viable for matrix operation, first it the sum of the matrices is $\begin{bmatrix} 5 & 4 & 3 \\ 2 & 3 & 4 \\ 8 & 9 & 10 \end{bmatrix}$ then for the difference of the marices is $\begin{bmatrix} 3 & 2 & 1 \\ 0 & 1 & 2 \\ 6 & 7 & 8 \end{bmatrix}$ after that it will determine the element wise of mulitplicationo of the matrices which is $\begin{bmatrix} 4 & 3 & 2 \\ 1 & 2 & 3 \\ 7 & 8 & 9 \end{bmatrix}$ and lastly for its element-wise of divison of the matrices is rounded off which is equivalent to $\begin{bmatrix} 4. & 3. & 2. \\ 1. & 2. & 3. \\ 7. & 8. & 9. \end{bmatrix}$.

```
Second Pair
The following description is the 1st matrix:
[[1 1 1]
 [1 1 1]
 [1 1 1]]

Shape : (3, 3)
Rank: 2
Size: 9

is it a square matrix?: True
is it identity matrix?: False
Is is a ones matrix?: True
Is it a zero matrix?: False

This is a matrix, The following description is the 2nd matrix:
[[0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]

Shape : (3, 4)
Rank: 2
Size: 12

is it a square matrix?: False
is it identity matrix?: False
Is is a ones matrix?: False
Is it a zero matrix?: True

'Not viable for matrix operation'
```

Figure 15: Second pair of matrix for task 2

In figure 15 shown the second pair of the matrices. The first matrix has a 3 by 3 shape meaning it has equal rows and columns. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 9 because the shape of the matrix is multiplied by 3 and 3 which is equivalent to 9. The first matrix is a square since its rows and columns were equal. Next, it is not an identity matrix even it has ones values in the diagonal section it doesn't have zero in it. After that, the matrix is determined as ones matrix since it has ones in all elements of one thus it is not a zero matrix since all elements are not in zero. For the second matrix is has a shape of 3 by 4 meaning the rows and columns are not equal. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 12 because the shape of the matrix is multiplied by 3 and 4 which is equivalent to 12. The second matrix is not a square matrix since its rows and columns were not equal. Next, it is not an identity matrix even it has zero in it, in the diagonal section doesn't have a ones values. After that, the matrix is determined as not ones matrix since all its elements are not in one thus it is a zero matrix since all elements are in zero. Since the matrices don't have the same shape the condition is not met therefore, it will return to its error message.

```
Third Pair
The following description is the 1st matrix:
[[1 0 0]
 [0 1 0]
 [0 0 1]]

Shape : (3, 3)
Rank: 2
Size: 9

is it a square matrix?: True
is it identity matrix?: True
Is is a ones matrix?: False
Is it a zero matrix?: False

This is a matrix, The following description is the 2nd matrix:
[[1 1 1]
 [1 1 1]
 [1 1 1]]

Shape : (3, 3)
Rank: 2
Size: 9

is it a square matrix?: True
is it identity matrix?: False
Is is a ones matrix?: True
Is it a zero matrix?: False


Matrices are viable for operation
(('Sum of matrices', array([[2, 1, 1],
        [1, 2, 1],
        [1, 1, 2]])), ('Difference of matrices', array([[ 0, -1, -1],
        [-1,  0, -1],
        [-1, -1,  0]])), ('Element-wise of multiplication', array([[1, 0, 0],
        [0, 1, 0],
        [0, 0, 1]])), ('Element-wise of division', array([[1., 0., 0.],
        [0., 1., 0.],
        [0., 0., 1.]])))
```

Figure 16: Third pair of matrix for task 2

In figure 16 shown for the third pair of matrices. The first matrix has a 3 by 3 shape meaning it has equal rows and columns. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 9 because the shape of the matrix is multiplied by 3 and 3 which is equivalent to 9. The first matrix is a square since its rows and columns were equal. Next, it is an identity matrix since it has ones in the diagonal section and zero values in it. After that, the matrix is determined as not ones matrix since all elements are not in ones thus it is not a zero matrix since all elements are not in zero. The second matrix has a 3 by 3 shape meaning it has equal rows and columns. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 9 because the shape of the matrix is multiplied by 3 and 3 which is equivalent to 9. The second matrix is a square since its rows and columns were equal. Next, it is not an identity matrix even it has ones in the diagonal section but it doesn't have zero values in it. After that, the matrix is ones matrix since all elements are in ones thus it is not a zero matrix since all elements are not in zero. Since it has an equal shape of matrix it will proceed for the matrix operation first to get it sum which is $\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 1 & 2 \end{bmatrix}$ and for the difference of the matrices is $\begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$ followed by its element-wise of multiplication which is equivalent to $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and lastly its element-wise of division of the matrices is rounded off which is equivalent to $\begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{bmatrix}$.

```
Fourth Pair
The following description is the 1st matrix:
[[0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]

Shape : (3, 4)
Rank: 2
Size: 12

is it a square matrix?: False
is it identity matrix?: False
Is is a ones matrix?: False
Is it a zero matrix?: True

This is a matrix, The following description is the 2nd matrix:
[[1 2 3]
 [5 4 3]
 [7 8 9]]

Shape : (3, 3)
Rank: 2
Size: 9

is it a square matrix?: True
is it identity matrix?: False
Is is a ones matrix?: False
Is it a zero matrix?: False

'Not viable for matrix operation'
```

Figure 17: Fourth pair of matrix for task 2

In figure 17 shown the fourth pair of matrices.. The first matrix is has a shape of 3 by 4 meaning the rows and columns are not equal. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 12 because the shape of the matrix is multiplied by 3 and 4 which is equivalent to 12. The matrix is not a square matrix since its rows and columns were not equal. Next, it is not an identity matrix even it has zero in it, the diagonal section doesn't have ones values. After that, the matrix is determined as not ones matrix since all its elements are not in ones thus it is zero matrices since all elements are in zero. The second matrix has a 3 by 3 shape meaning it has equal rows and columns. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 9 because the shape of the matrix is multiplied by 3 and 3 which is equivalent to 9. The first matrix is a square since its rows and columns were equal. Next, it is not an identity matrix since it doesn't have ones in the diagonal section and zero values in it. After that, the matrix is determined as not ones matrix since its elements are not in ones thus it is not a zero matrix since all elements are not in zero. Since the matrices don't have the equal shape of the matrix the condition is not met therefore, it will return to its error message.

```
Fifth Pair
The following description is the 1st matrix:
[[1 2 3]
 [5 4 3]
 [7 8 9]]

Shape : (3, 3)
Rank: 2
Size: 9

is it a square matrix?: True
is it identity matrix?: False
Is is a ones matrix?: False
Is it a zero matrix?: False

This is a matrix, The following description is the 2nd matrix:
[[4 3 2]
 [1 2 3]
 [7 8 9]]

Shape : (3, 3)
Rank: 2
Size: 9

is it a square matrix?: True
is it identity matrix?: False
Is is a ones matrix?: False
Is it a zero matrix?: False


Matrices are viable for operation
: (('Sum of matrices', array([[ 5,  5,  5],
         [ 6,  6,  6],
         [14, 16, 18]])), ('Difference of matrices', array([[-3, -1,  1],
         [ 4,  2,  0],
         [ 0,  0,  0]])), ('Element-wise of multiplication',
  array([[ 4,  6,  6],
         [ 5,  8,  9],
         [49, 64, 81]])), ('Element-wise of division',
  array([[0.25      , 0.66666667, 1.5       ],
         [5.        , 2.        , 1.        ],
         [1.        , 1.        , 1.        ]])))
```

Figure 18: Fifth pair of the matrix for task 2

In figure 18 shown the last pair or fifth pair of the matrices. The first matrix has a 3 by 3 shape meaning it has equal rows and columns. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 9 because the shape of the matrix is multiplied by 3 and 3 which is equivalent to 9. The first matrix is a square since its rows and columns were equal. Next, it is not an identity matrix since it doesn't have ones in the diagonal section as well as zero in it. After that, the matrix is determined as not ones matrix the elements are not in ones thus it is not a zero matrix since all elements are not in zero. For the second matrix, it has a 3 by 3 shape meaning it has equal rows and columns. Then since it is a 2dimensional array the rank of the matrix is 2, followed by its size which is 9 because the shape of the matrix is multiplied by 3 and 3 which is equivalent to 9. The first matrix is a square since its rows and columns were equal. Next, it is not an identity matrix since it doesn't have ones in the diagonal section and zero values in it. After that, the matrix is not ones since it doesn't have one in the element thus it is not a zero matrix since all elements are not in zero. Since the matrices have an equal shape of matrix it will proceed for the matrix operation first to get it sum which is $\begin{bmatrix} 5 & 5 & 5 \\ 6 & 6 & 6 \\ 14 & 16 & 18 \end{bmatrix}$, for the difference of the matrices is $\begin{bmatrix} -3 & -1 & 1 \\ 4 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ followed by its element-wise of multiplication of the matrices which is equivalent to $\begin{bmatrix} 4 & 6 & 6 \\ 5 & 8 & 9 \\ 49 & 64 & 81 \end{bmatrix}$ and lastly its element-wise of division of the matrices which is rounded off that is equivalent to $\begin{bmatrix} 0.25 & 0.67 & 1.5 \\ 5. & 2. & 1. \\ 1. & 1. & 1. \end{bmatrix}$.

```
These are scalar parameters
This is 1st scalar = 1
This is 2nd scalar = 2
```

Figure 19: Scalar parameters for task 2

In figure shown is an example of an scalar parameters, that satisfies the condition in displaying the values of the two scalars.

IV. Conclusion

In this laboratory activity introduced a concept about how the basic operations of matrices work and it was shown in the results that matrices can be determine according to their shape, size and rank. Also it can be determine if the matrix is empty, square, an identity, ones, or zero matrix. Followed by its basic operations such as getting its sum, difference, element wise of product,and element-wise of the quotient using different NumPy's present functions.The application of this activity is to show different descriptions of the matrices and its basic operations to achieve it I used NumPy's presents functions which is easier to understand. Also, it can be used to work on multiple linear equations and to a system linear equations. [13]

Matrix operations is used to solve problems in agriculture such getting its maximum profit or lowest profit, optimal resource allocation, plant production and etc. Also, it can be a new technique to have an accurate data in terms of managing and controlling its production and stocks. [14]. This approach can be used as a tool in getting its proportionality, additivity, divisibility and certainty of data in the agricultural system. [14]

References

[1] W3Schools,2020. *About "NumPy Creating Arrays"* Available: https://www.w3schools.com/python/numpy_creating_arrays.asp Data Accessed: 11/24/2020

[2] W3Schools,2020. *About "NumPy Array Shape"* Available: https://www.w3schools.com/python/numpy_array_shape.asp Data Accessed: 11/24/2020

[3] The SciPy community,2020. *About "numpy.prod"* Available: https://numpy.org/doc/stable/reference/generated/numpy.prod.html Data Accessed: 11/24/2020

[4] GeeksforGeeks, 2020.. *About "numpy.ndarray.ndim() method | Python"* Available: https://www.geeksforgeeks.org/numpy-ndarray-ndim-method-python/#:~:text=ndim()%20function%20return%20the%20number%20of%20dimensions%20of%20an%20array.&text=Parameters%20%3A,ndarray%2C%20a%20conversion%20is%20attempted. Data Accessed: 11/24/2020

[5] The SciPy community,2020. *About "numpy.allclose"* Available: https://numpy.org/doc/stable/reference/generated/numpy.allclose.html Data Accessed: 11/24/2020

[6]] The SciPy community,2020. *About "numpy.identity."* Available: https://numpy.org/doc/stable/reference/generated/numpy.identity.html Data Accessed: 11/24/2020

[7] GeeksforGeeks, 2020.. *About "numpy.all() in Python."* Available: https://www.geeksforgeeks.org/numpy-all-in-python/ Data Accessed: 11/24/2020

[8] The SciPy community,2020. *About "numpy.isscalar.."* Available: https://numpy.org/doc/stable/reference/generated/numpy.isscalar.html Data Accessed: 11/24/2020

[9] GeeksforGeeks, 2020.. *About "numpy.add() in Python."* Available: https://www.geeksforgeeks.org/numpy-add-in-python/ Data Accessed: 11/24/2020

[10] The SciPy community,2020. *About "numpy.subtract."* Available: https://numpy.org/doc/stable/reference/generated/numpy.subtract.html Data Accessed: 11/24/2020

[11] GeeksforGeeks, 2020. *About "numpy.multiply() in Python."* Available: https://www.geeksforgeeks.org/numpy-multiply-in-python/ Data Accessed: 11/24/2020

[12] GeeksforGeeks, 2020. *About "numpy.divide() in Python."* Available: https://www.geeksforgeeks.org/numpy-divide-python/ Data Accessed: 11/24/2020

[13] Lumen Candela, 2019. *About "Introduction to Matrices."* Available: https://courses.lumenlearning.com/boundless-algebra/chapter/introduction-to-matrices/#:~:text=the%20jth%20column.-,Matrices%20can%20be%20used%20to%20compactly%20write%20and%20work%20with,also%20known%20as%20linear%20maps. Data Accessed: 11/26/2020

[14] Vico&Bodriroga, 2017. *About Tools for planning in agriculture – Linear programming approach."* Available: https://www.researchgate.net/publication/318505728_Tools_for_planning_in_agriculture_-_Linear_programming_approach Data Accessed: 11/26/2020