Lab-11

LAB 4: Computation of basis and dimension for a vector space and graphical representation of linear transformation

4.1 Objectives:

Use python

1. to verify the Rank nullity theorem of given linear transformation
2. to compute the dimension of vector space
3. to represent linear transformations graphically

4.2 Rank Nullity Theorem

Verify the rank-nullity theorem for the linear transformation $T : \mathbb{R}^3 \to \mathbb{R}^3$ defined by $T(x, y, z) = (x + 4y + 7z, 2x + 5y + 8z, 3x + 6y + 9z)$.

```
import numpy as np
from scipy.linalg import null_space

# Define a linear transformation interms of matrix
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

# Find the rank of the matrix A
rank = np.linalg.matrix_rank(A)
print("Rank of the matrix",rank)

# Find the null space of the matrix A
ns = null_space(A)
print("Null space of the matrix",ns)
# Find the dimension of the null space
nullity = ns.shape[1]
print("Null space of the matrix",nullity)
# Verify the rank-nullity theorem
if rank + nullity == A.shape[1]:
    print("Rank-nullity theorem holds.")
else:
    print("Rank-nullity theorem does not hold.")
```

```
Rank of the matrix 2
Null space of the matrix [[-0.40824829]
 [ 0.81649658]
 [-0.40824829]]
Null space of the matrix 1
Rank-nullity theorem holds.
```

4.3 Dimension of Vector Space

Find the dimension of subspace spanned by the vectors $(1, 2, 3), (2, 3, 1)$ and $(3, 1, 2)$.

```
import numpy as np

# Define the vector space V
V = np.array([
    [1, 2, 3],
    [2, 3, 1],
    [3, 1, 2]])
# Find the dimension and basis of V
basis = np.linalg.matrix_rank(V)
dimension = V.shape[0]
print("Basis of the matrix",basis)
print("Dimension of the matrix",dimension)
```

```
Basis of the matrix 3
Dimension of the matrix 3
```

Extract the linearly independent rows in given matrix : Basis of Row space

```
from numpy import *
import sympy as sp
A=[[1,-1,1,1],[2,-5,2,2],[3,-3,5,3],[4,-4,4,4]]
AB=array(A)
S=shape(A)
n=len(A)
for i in range(n):
    if AB[i,i]==0:
        ab=copy(AB)
        for k in range(i+1,S[0]):
            if ab[k,i]!=0:
                ab[i,:]=AB[k,:]
                ab[k,:]=AB[i,:]
                AB=copy(ab)
    for j in range(i+1,n):
        Fact=AB[j,i]/AB[i,i]
        for k in range(i,n):
            AB[j,k]=AB[j,k]-Fact*AB[i,k]
display("REF of given matrix: ",sp.Matrix(AB))
temp = {(0, 0, 0,0)}
result = []
for idx, row in enumerate(map(tuple, AB)):
    if row not in temp:
        result.append(idx)
print("\n Basis are non-zero rows of A:")
display(sp.Matrix(AB[result]))
```

'REF of given matrix: '

$$\begin{bmatrix} 1 & -1 & 1 & 1 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Basis are non-zero rows of A:

$$\begin{bmatrix} 1 & -1 & 1 & 1 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$