Name: Akash Shivaji Varude

Roll No: 231110006

Assignment No 1

Que 1)

i]

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

As both the columns are linearly independent & form a standard basis in 2D space, column space of matrix A is entire 2D euclidean space R^2

$$\therefore \text{Col}(A) = \text{span}\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

ii]

$$B = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

we can easily see second column is linearly dependent on first column. Hence column space of matrix B is one dimensional subspace spanned by first column $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$$\therefore \text{col}(B) = \text{span}\left\{ \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}$$

iii] $D = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 4 \end{bmatrix}$

Here matrix D has 2^{nd} column $\begin{bmatrix} 2 \\ 0 \end{bmatrix}$ linearly dependent on 1^{st} column $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$. 3^{rd} column is linearly independent on first two columns.

$$\therefore \text{col}(D) = \text{span}\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 \\ 4 \end{bmatrix} \right\}$$

Que 2

```
Satellite Positions are:
 [[ 40000000  50000000  66660000]
 [-80000000  25000000 -44000000]
 [ 65000000  98000000  10450000]
 [ 20000000 -43000000 -66060000]
 [ 55000000  35000000  42300000]]
```

a) Fixed user location at (100,100,100) and considered the speed as 3x10^8 mtr/sec
Hence Time=distance/speed
Here I have considered Euclidean distance i.e l2 norm
Hence for each satellite:

Distance = || Sattelite_position – User_Position $||_2$

Hence Time required from each satellite to user is:

```
travel Times are: [0.30810395554513015, 0.3155422105947158,
0.39353339276680904, 0.27106664641805955, 0.2590423290170075]
```

b) Calculating User position using trilateration:
We have equations :

$$
\begin{aligned}
(x-x_1)^2+(y-y_1)^2+(z-z_1)^2 &= r_1^2\\
(x-x_2)^2+(y-y_2)^2+(z-z_2)^2 &= r_2^2\\
(x-x_3)^2+(y-y_3)^2+(z-z_3)^2 &= r_3^2\\
(x-x_4)^2+(y-y_4)^2+(z-z_4)^2 &= r_4^2
\end{aligned}
$$

These Equations can be written in matrix format as:

$$
\begin{bmatrix} 2(x_2-x_1) & 2(y_2-y_1) & 2(z_2-z_1)\\ 2(x_3-x_2) & 2(y_3-y_2) & 2(z_3-z_2)\\ 2(x_4-x_3) & 2(y_4-y_3) & 2(z_4-z_3) \end{bmatrix} \begin{bmatrix} x\\ y\\ z \end{bmatrix} = \begin{bmatrix} (r_1^2-r_2^2)-(x_1^2-x_2^2)-(y_1^2-y_2^2)-(z_1^2-z_2^2)\\ (r_2^2-r_3^2)-(x_2^2-x_3^2)-(y_2^2-y_3^2)-(z_2^2-z_3^2)\\ (r_3^2-r_4^2)-(x_3^2-x_4^2)-(y_3^2-y_4^2)-(z_3^2-z_4^2) \end{bmatrix}
$$

We can get the solution of this by using formula:

$$\hat{x} = (A^T A)^{-1} A^T b$$

Used Command:

```
solution= np.matmul(np.linalg.inv(np.matmul(A.T,A)),np.matmul(A.T,B))
```

Output:

```
User Postion(No Error Introduced):  [ 99.99999999  99.99999997 100.00000002]
inaccuracy in position:  3.5732363043569984e-08
```

c) Calculating User position using trilateration (Introducing error in time):
I used command
`rand_error=np.random.rand()/10**8`

to generate random error and introduced that in time and calculated user position and inaccuracy

```
User Postions:  [ 99.84642565  99.55079524 100.34796417]
inaccuracy in position:  0.5885992308383305
```

d) I have created numpy array of size 20 with randomly generated values in it and divided it by 10**8 as error would be small in time. And resultant graph is :

```
Avarage Inaccuracy : 0.6796791135039068
```