5.13.58

Kartik Lahoti - EE25BTECH11032

September 30, 2025

Question

Let $\omega \neq 1$ be a cube root of unity and $\mathbb{S}$ be the set of all non-singular matrices of the form

$$\begin{pmatrix} 1 & a & b \\ \omega & 1 & c \\ \omega^2 & \omega & 1 \end{pmatrix} \tag{1}$$

where each of a, b and c is either ω or ω^2. Then the number of distinct matrices in the set $\mathbb{S}$ is

Theoretical Solution

Let,

$$\mathbf{A} = \begin{pmatrix} 1 & a & b \\ \omega & 1 & c \\ \omega^2 & \omega & 1 \end{pmatrix} \tag{2}$$

where $\mathbf{A} \in \mathbb{S}$

For $\mathbf{A}$ to be Non-singular , $\mathbf{A}$ should be a full rank matrix.

Theoretical Solution

Thus,

$$rank\,(\mathbf{A}) = 3 \tag{3}$$

$$\begin{pmatrix} 1 & a & b \\ \omega & 1 & c \\ \omega^2 & \omega & 1 \end{pmatrix} \xleftrightarrow{R_2 \to R_2 - \omega R_1} \begin{pmatrix} 1 & a & b \\ 0 & 1-a\omega & c-b\omega \\ \omega^2 & \omega & 1 \end{pmatrix} \tag{4}$$

Theoretical Solution

$$\begin{pmatrix} 1 & a & b \\ 0 & 1-a\omega & c-b\omega \\ \omega^2 & \omega & 1 \end{pmatrix} \xleftrightarrow{R_2 \to R_2 - \omega^2 R_1} \begin{pmatrix} 1 & a & b \\ 0 & 1-a\omega & c-b\omega \\ 0 & \omega - a\omega^2 & 1-b\omega^2 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} 1 & a & b \\ 0 & 1-a\omega & c-b\omega \\ 0 & \omega - a\omega^2 & 1-b\omega^2 \end{pmatrix} \xleftrightarrow{R_3 \to R_3 - \omega R_2} \begin{pmatrix} 1 & a & b \\ 0 & 1-a\omega & c-b\omega \\ 0 & 0 & 1-c\omega \end{pmatrix} \tag{6}$$

Theoretical Solution

For this Row Reduced Echelon Form Matrix to be a full rank matrix, the diagonal pivots should be non-zero.

$$1 - c\omega \neq 0 \implies c = \omega \tag{7}$$

$$1 - a\omega \neq 0 \implies a = \omega \tag{8}$$

Theoretical Solution

Non-singularity does not depends on b thus, $b \in \{\omega, \omega^2\}$

$$\therefore n(\mathbb{S}) = 1 \times 2 \times 1 \tag{9}$$

$$n(\mathbb{S}) = 2 \tag{10}$$

Hence , number of Matrices in Set $\mathbb{S}$ is 2.

C Code - Line Generator

```
void linegen(double *X, double *Y , double *Z , double *A ,
    double *B , int n , int m )
{
    double temp[m] ;
    for (int i = 0 ; i < m ; i++)
    {
        temp [ i ] = (B[i]- A[i]) /(double) n ;
    }
    for (int i = 0 ; i <= n ; i++ )
    {
        X[i] = A[0] + temp[0] * i ;
        Y[i] = A[1] + temp[1] * i ;
        Z[i] = A[2] + temp[2] * i ;
    }
}
```

Python Code - 1

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt

i = 1
def line_cre(P: np.ndarray , Q: np.ndarray, str):
    handc1 = ctypes.CDLL("./line_gen.so")
    global i
    handc1.linegen.argtypes = [
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.POINTER(ctypes.c_double),
        ctypes.c_int , ctypes.c_int
    ]
    handc1.linegen.restype = None
```

Python Code

```
n = 200
    X_1 = np.zeros(n,dtype=np.float64)
    Y_1 = np.zeros(n,dtype=np.float64)
    Z_1 = np.zeros(n,dtype=np.float64)
    handc1.linegen (
        X_1.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        Y_1.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        Z_1.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        P.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        Q.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        n,3
    )
```

Python Code

```
    if i == 1 :
        ax.plot([X_1[0],X_1[-1]],[Y_1[0],Y_1[-1]],[Z_1[0],Z_1
            [-1]],str,label = "Correct Solution" )
    elif i == 2 :
        ax.plot([X_1[0],X_1[-1]],[Y_1[0],Y_1[-1]],[Z_1[0],Z_1
            [-1]],str,label = "Incorrect Solution" )
    else :
        ax.plot([X_1[0],X_1[-1]],[Y_1[0],Y_1[-1]],[Z_1[0],Z_1
            [-1]],str)
    i=i+1
```

Python Code

```
fig = plt.figure()

ax = fig.add_subplot(111,projection="3d")
# let omega correspond to 1 and omega square to -1
O = np.zeros(3,dtype=np.float64).reshape(-1,1)
p1 = np.array([1,1,1],dtype=np.float64).reshape(-1,1)
p2 = np.array([-1,1,1],dtype=np.float64).reshape(-1,1)
p3 = np.array([1,-1,1],dtype=np.float64).reshape(-1,1)
p4 = np.array([1,1,-1],dtype=np.float64).reshape(-1,1)
p5 = np.array([1,-1,-1],dtype=np.float64).reshape(-1,1)
p6 = np.array([-1,1,-1],dtype=np.float64).reshape(-1,1)
p7 = np.array([-1,-1,1],dtype=np.float64).reshape(-1,1)
p8 = np.array([-1,-1,-1],dtype=np.float64).reshape(-1,1)
```

Python Code

```
line_cre(p1,0,"g-")
line_cre(p2,0,"r-")
line_cre(p3,0,"g-")
line_cre(p4,0,"r-")
line_cre(p5,0,"r-")
line_cre(p6,0,"r-")
line_cre(p7,0,"r-")
line_cre(p8,0,"r-")

coords = np.block([[p1,p2,p3,p4,p5,p6,p7,p8]])
ax.scatter(coords[0,:],coords[1,:],coords[2,:])
vert_labels =[r'$p_1$',r'$p_2$',r'$p_3$',r'$p_4$',r'$p_5$',r'
    $p_6$',r'$p_7$',r'$p_8$']
```

Python Code

```
for i, txt in enumerate(vert_labels):
    ax.text(coords[0,i], coords[1,i] ,coords[2,i],f'{txt}\n({
        coords[0,i]:.0f}, {coords[1,i]:.0f},{coords[2,i]:.0f})',
        ha='center', va = 'bottom')

ax.scatter(0, 0, 0, color="black", label="ORIGIN")
ax.text(0,0,0,"",ha='left')
ax.legend(loc='upper left', bbox_to_anchor=(.80, 1.05))
ax.set_xlabel('$a$')
ax.set_ylabel('$b$')
ax.set_zlabel('$c$')
ax.grid()
plt.title("Fig:5.13.58")
fig.savefig("../figs/vector1.png")
fig.show()
#plt.savefig('../figs/vector1.png')
#subprocess.run(shlex.split("termux-open ../figs/vector1.png"))
```

Python Code- 2

```
import math
import sys
sys.path.insert(0, '/home/kartik-lahoti/matgeo/codes/CoordGeo')
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

from line.funcs import *

i = 1
O = np.zeros(3,dtype=np.float64).reshape(-1,1)
```

Python Code

```
def plot_it(P,Q,str):
    global i
    x_l = line_gen_num(P,Q,20)
    if i == 1 :
        ax.plot(x_l[0,:],x_l[1,:],x_l[2,:] , str , label = "
            Correct Solution" )
    elif i == 2 :
        ax.plot(x_l[0,:],x_l[1,:],x_l[2,:] , str , label = "
            Inorrect Solution")
    else :
        ax.plot(x_l[0,:],x_l[1,:],x_l[2,:] , str)
    i += 1
```

Python Code

```
fig = plt.figure()

ax = fig.add_subplot(111,projection="3d")
# let omega correspond to 1 and omega square to -1

p1 = np.array([1,1,1],dtype=np.float64).reshape(-1,1)
p2 = np.array([-1,1,1],dtype=np.float64).reshape(-1,1)
p3 = np.array([1,-1,1],dtype=np.float64).reshape(-1,1)
p4 = np.array([1,1,-1],dtype=np.float64).reshape(-1,1)
p5 = np.array([1,-1,-1],dtype=np.float64).reshape(-1,1)
p6 = np.array([-1,1,-1],dtype=np.float64).reshape(-1,1)
p7 = np.array([-1,-1,1],dtype=np.float64).reshape(-1,1)
p8 = np.array([-1,-1,-1],dtype=np.float64).reshape(-1,1)
```

Python Code

```
plot_it(p1,0,"g-")
plot_it(p2,0,"r-")
plot_it(p3,0,"g-")
plot_it(p4,0,"r-")
plot_it(p5,0,"r-")
plot_it(p6,0,"r-")
plot_it(p7,0,"r-")
plot_it(p8,0,"r-")

coords = np.block([[p1,p2,p3,p4,p5,p6,p7,p8]])
ax.scatter(coords[0,:],coords[1,:],coords[2,:])
vert_labels = [r'$p_1$',r'$p_2$',r'$p_3$',r'$p_4$',r'$p_5$',r'
    $p_6$',r'$p_7$',r'$p_8$']
```

Python Code

```
for i, txt in enumerate(vert_labels):
    ax.text(coords[0,i], coords[1,i] , coords[2,i],f'{txt}\n({
        coords[0,i]:.0f}, {coords[1,i]:.0f}, {coords[2,i]:.0f})',
        ha='center', va = 'bottom')

ax.scatter(0, 0, 0, color="black", label="ORIGIN")
ax.text(0,0,0,"",ha='left')
ax.legend(loc='upper left', bbox_to_anchor=(.80, 1.05))
ax.set_xlabel('$a$')
ax.set_ylabel('$b$')
ax.set_zlabel('$c$')
ax.grid()
plt.title("Fig:5.13.58")
fig.savefig("../figs/vector2.png")
fig.show()
#plt.savefig('../figs/vector2.png')
#subprocess.run(shlex.split("termux-open ../figs/vector2.png"))
```

Plot

In the graph , Let 1 be equivalent to ω and -1 be equivalent to ω^2.

Fig:5.13.58