5.2.35

Kartik Lahoti - EE25BTECH11032

September 18, 2025

Question

Solve the following system of linear equations.

$$3x + 4y = 10$$

$$2x - 2y = 2$$

Theoretical Solution

The equation of line L_1 is,

$$\begin{pmatrix} 3 & 4 \end{pmatrix} \mathbf{x} = 10 \tag{1}$$

The equation of line L_2 is,

$$\begin{pmatrix} 2 & -2 \end{pmatrix} \mathbf{x} = 2 \tag{2}$$

Theoretical Solution

On putting the equations in a matrix, we will get

$$\implies \begin{pmatrix} 3 & 4 \\ 2 & -2 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 10 \\ 2 \end{pmatrix} \tag{3}$$

So the augmented matrix is,

$$\left(\begin{array}{cc|c} 3 & 4 & 10 \\ 2 & -2 & 2 \end{array}\right) \tag{4}$$

Theoretical Solution

$$\left(\begin{array}{cc|c} 3 & 4 & 10 \\ 2 & -2 & 2 \end{array}\right) \xleftrightarrow{R_2 \to R_2 - \frac{2}{3}R_1} \left(\begin{array}{cc|c} 3 & 4 & 10 \\ 0 & \frac{-14}{3} & \frac{-14}{3} \end{array}\right) \tag{5}$$

$$\left(\begin{array}{cc|c} 3 & 4 & 10 \\ 0 & \frac{-14}{3} & \frac{-14}{3} \end{array}\right) \xleftrightarrow{R_2 \to \frac{-3}{14}R_2} \left(\begin{array}{cc|c} 3 & 4 & 10 \\ 0 & 1 & 1 \end{array}\right) \tag{6}$$

$$\left(\begin{array}{cc|c} 3 & 4 & 10 \\ 0 & 1 & 1 \end{array}\right) \xleftrightarrow{R_1 \to R_1 - 4R_2} \left(\begin{array}{cc|c} 3 & 0 & 6 \\ 0 & 1 & 1 \end{array}\right) \tag{7}$$

Theoretical Solution

$$\left(\begin{array}{cc|c} 3 & 0 & 6 \\ 0 & 1 & 1 \end{array}\right) \xleftrightarrow{R_1 \to \frac{1}{3}R_1} \left(\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 1 \end{array}\right) \tag{8}$$

$$\implies \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \equiv \begin{pmatrix} 2 \\ 1 \end{pmatrix} \tag{9}$$

Therefore the two lines will intersect at $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$.

C Code (1)

```
void gaussian(double *A , double *B , double *C , double *X)
{ // works only if all 4 values in A and B are non-zero
    int i = 1 , j = 0 ;
    if ( A[i] != 0 ){
        B[i] -= A[i]/A[j] * B[j];
        C[i] -= A[i]/A[j] * C[j];
        A[i] = 0 ;}
    if(B[i] != 0){
        C[i] = C[i] / B[i] ;
        B[i] = 1;}
    if(B[j] != 0){
        C[j] -= C[i] * B[j];
        B[j] = 0 ;}
    X[j] = C[j] / A[j];
    X[i] = C[i];
}
```

C Code (2) - Function to Generate Points on Line

```c
void linegen(double *XY, 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++ )
        for (int j = 0 ; j < m ; j++)
            XY[j*n + i ] = A[j] + temp[j] * i ; :wq
}
```

Python Code - Using Shared Object

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

handc1 = ct.CDLL("./func.so")

handc1.gaussian.argtypes = [
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double)
]

handc1.gaussian.restype = None
```

Python Code - Using Shared Object

```
A = np.array([3,2], dtype = np.float64).reshape(-1,1)
B = np.array([4,-2], dtype = np.float64).reshape(-1,1)
C = np.array([10,2] , dtype = np.float64).reshape(-1,1)
X = np.zeros(2, dtype = np.float64).reshape(-1,1)

handc1.gaussian(
    A.ctypes.data_as(ct.POINTER(ct.c_double)),
    B.ctypes.data_as(ct.POINTER(ct.c_double)),
    C.ctypes.data_as(ct.POINTER(ct.c_double)),
    X.ctypes.data_as(ct.POINTER(ct.c_double)),
)
print("Vector X = " , X)
```

Python Code - Using Shared Object

```
def line(P: np.ndarray , Q: np.ndarray, str1 , str2):
    handc2 = ct.CDLL("./line_gen.so")

    handc2.linegen.argtypes = [
        ct.POINTER(ct.c_double),
        ct.POINTER(ct.c_double),
        ct.POINTER(ct.c_double),
        ct.c_int , ct.c_int
    ]

    handc2.linegen.restype = None

    handc2.line_cre.restype = None
```

Python Code - Using Shared Object

```
n = 200
    XY = np.zeros((2,n),dtype=np.float64)

    handc2.linegen (
        XY.ctypes.data_as(ct.POINTER(ct.c_double)),
        P.ctypes.data_as(ct.POINTER(ct.c_double)),
        Q.ctypes.data_as(ct.POINTER(ct.c_double)),
        n,2
    )
    plt.plot(XY[0,:],XY[1,:], str1 , label = str2 )
```

Python Code - Using Shared Object

```
P = np.array([6,-2], dtype = np.float64).reshape(-1,1)
Q = np.array([-6,7], dtype = np.float64).reshape(-1,1)
line(P,Q,"g-"," Line 1 ")
P = np.array([8,7], dtype = np.float64).reshape(-1,1)
Q = np.array([-8,-9], dtype = np.float64).reshape(-1,1)
line(P,Q,"r-"," Line 2 ")
plt.scatter(X[0,0], X[1,0])
plt.annotate(f"X\n({X[0,0]},{X[1,0]})",
                    (X[0], X[1]),
                    textcoords = "offset points" ,
                    xytext = (0,12),ha = "center")
```

Python Code - Using Shared Object

```
plt.xlim([-1,4])
plt.ylim([-1,4])
plt.xlabel("$x$")
plt.ylabel("$y$")
plt.grid()
plt.legend(loc="best")
plt.title("5.2.35")
plt.savefig("../figs/intersect1.png")
plt.show()
#plt.savefig('../figs/intersect1.png')
#subprocess.run(shlex.split("termux-open ../figs/intersect1.png")
    )
```

Python Code

```
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

from line.funcs import *

#if using termux
#import subprocess
#import shlex
```

Python Code

```
A = np.array([[3,4],
              [2,-2]], dtype = np.float64)
C = np.array([10,2] , dtype = np.float64).reshape(-1,1)

X = LA.solve(A,C)
print("Vector X = " , X )

def plot_it(P,Q,str1,str2):
    x_l = line_gen_num(P,Q,20)
    plt.plot(x_l[0,:],x_l[1,:] , str1 , label = str2)
```

Python Code

```
plt.figure()
P = np.array([6,-2], dtype = np.float64).reshape(-1,1)
Q = np.array([-6,7], dtype = np.float64).reshape(-1,1)
plot_it(P,Q,"g-"," Line 1 ")
P = np.array([8,7], dtype = np.float64).reshape(-1,1)
Q = np.array([-8,-9], dtype = np.float64).reshape(-1,1)
plot_it(P,Q,"r-"," Line 2 ")
plt.scatter(X[0,0], X[1,0])
plt.annotate(f"X\n({X[0,0]},{X[1,0]})",
                    (X[0], X[1]),
                    textcoords = "offset points" ,
                    xytext = (0,12),ha = "center")
```

Python Code

```
plt.xlim([-1,4])
plt.ylim([-1,4])
plt.xlabel("$x$")
plt.ylabel("$y$")
plt.grid()

plt.legend(loc="best")

plt.title("5.2.35")

plt.savefig("../figs/intersect2.png")
plt.show()

#plt.savefig('../figs/intersect2.png')
#subprocess.run(shlex.split("termux-open ../figs/intersect2.png")
    )
```

Plot