4.3.21

Kartik Lahoti - EE25BTECH11032

September 9 ,2025

Question

In what ratio does the X-axis divide the line segment joining the points $(-4, -6)$ and $(-1, 7)$? Find the coordinates of the point of division.

Theoretical Solution

Given :

Symbol	Value	Description
A	$\begin{pmatrix} -4 \\ -6 \end{pmatrix}$	Given Point
B	$\begin{pmatrix} -1 \\ 7 \end{pmatrix}$	Given Point
P	$\begin{pmatrix} x \\ 0 \end{pmatrix}$	Desired Point

Table: 4.3.21

Let , **P** be the point on x-axis dividing the line **AB** in the ratio say $k : 1$.

Theoretical Solution

Now, **A** , **B** and **P** are collinear ,

$$\therefore rank \begin{pmatrix} \mathbf{B}-\mathbf{A} & \mathbf{P}-\mathbf{B} \end{pmatrix} = 1 \tag{1}$$

$$\begin{pmatrix} 3 & x+1 \\ 13 & -7 \end{pmatrix} \xleftrightarrow{R_2 \to \frac{-13}{3}R_1+R_2} \begin{pmatrix} 3 & x+1 \\ 0 & -\frac{13x}{3}-\frac{34}{3} \end{pmatrix} \tag{2}$$

Theoretical Solution

The number of nonzero rows in the echelon form is defined as the rank. For the above matrix to be of rank 1 ,

$$-\frac{13x}{3} - \frac{34}{3} = 0 \tag{3}$$

$$x = -\frac{34}{13} \tag{4}$$

$\therefore$ The coordinates of the point of intersection is

$$\mathbf{P} = \begin{pmatrix} -\frac{34}{13} \\ 0 \end{pmatrix} \tag{5}$$

Theoretical Solution

Now,

$$k = \frac{(\mathbf{A} - \mathbf{P})^{\top} (\mathbf{P} - \mathbf{B})}{\|\mathbf{P} - \mathbf{B}\|^2} \tag{6}$$

Substituting the values of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{P}$, we get,

$$k = \frac{\begin{pmatrix} -\frac{18}{13} & -6 \end{pmatrix} \begin{pmatrix} -\frac{21}{13} \\ -7 \end{pmatrix}}{\left\| \begin{pmatrix} -\frac{21}{13} \\ -7 \end{pmatrix} \right\|^2} \tag{7}$$

$$k = \frac{6}{7} \tag{8}$$

Thus, the ratio in which the point $\mathbf{P}$ divides the line segment $\mathbf{AB}$ is $6:7$

C Code (1)

```c
#include <math.h>
double norm_sq (double *P , double *Q , int m )
{
        double sum = 0 ;
        for ( int i = 0 ; i < m ; i++ )
                sum += pow(P[i] - Q[i] , 2);
        return sum ;
}
double ratio ( double *A , double *B , double *P , double norm )
{
                double k = (A[0]-P[0]) * (P[0] - B[0]) + (A[1] - P
                    [1]) * (P[1] - B[1]);
                k = k / norm;
                return k ;
}
```

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

```c
#include <math.h>
void line_cre(double *X , double *Y , 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 ;
        }
}
```

Python Code - Using Shared Object

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

handc1 = ct.CDLL("./func.so")
handc1.norm_sq.argtypes = [
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double),
    ct.c_int
]
handc1.norm_sq.restype = ct.c_double
A = np.array([[-4],[-6]] , dtype = np.float64).reshape(-1,1)
B = np.array([[-1],[7]] , dtype = np.float64).reshape(-1,1)
P = np.array([[-34/13] ,[ 0 ]] , dtype = np.float64).reshape
    (-1,1)
```

Python Code - Using Shared Object

```
norm = handc1.norm_sq(
    B.ctypes.data_as(ct.POINTER(ct.c_double)),
    P.ctypes.data_as(ct.POINTER(ct.c_double)),
    2)
handc1.ratio.argtypes = [
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double),
    ct.c_double]

handc1.ratio.restype = ct.c_double
k = handc1.ratio(
    A.ctypes.data_as(ct.POINTER(ct.c_double)),
    B.ctypes.data_as(ct.POINTER(ct.c_double)),
    P.ctypes.data_as(ct.POINTER(ct.c_double)),norm)
```

Python Code - Using Shared Object

```
def line(P : np.ndarray , Q : np.ndarray , str ) :

    handc2 = ct.CDLL("./line_gen.so")
    handc2.line_cre.argtypes = [
        ct.POINTER(ct.c_double),
        ct.POINTER(ct.c_double),
        ct.POINTER(ct.c_double),
        ct.POINTER(ct.c_double),
        ct.c_int,ct.c_int
    ]
    n = 200
    X_l = np.zeros(n,dtype=np.float64)
    Y_l = np.zeros(n,dtype=np.float64)

    handc2.line_cre.restype = None
```

Python Code - Using Shared Object

```
handc2.line_cre(
        X_1.ctypes.data_as(ct.POINTER(ct.c_double)),
        Y_1.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([X_1[0],X_1[-1]],[Y_1[0],Y_1[-1]], str , label = "
        Line Segment AB")
```

Python Code - Using Shared Object

```
plt.figure()

line(A,B,"g--")

coords = np.block([[A,B,P]])

plt.scatter(coords[0,:] , coords[1,:])
vert_label = ['A', 'B' , 'P' ]

for i , txt in enumerate(vert_label) :
    plt.annotate(f"{txt}\n({coords[0,i]:.1f},{coords[1,i]:.1f})",
                 (coords[0,i], coords[1,i]),
                 textcoords = "offset points" ,
                 xytext = (10,20),ha = "center")
```

Python Code - Using Shared Object

```
plt.xlabel("$x$")
plt.ylabel("$y$")
plt.grid()

plt.legend(loc="best")

plt.title("4.3.21")

#plt.savefig("../figs/section1.png")
#plt.show()
plt.savefig('../figs/section1.png')
subprocess.run(shlex.split("termux-open ../figs/section1.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 *
#from triangle.funcs import *
#from conics.funcs import circ_gen

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

Python Code

```
def norm_sq(P,Q) :
    return pow(LA.norm(P-Q),2)

def ratio(A,B,P,norm) :
    k = (A[0]-P[0]) * (P[0] - B[0]) + (A[1] - P[1]) * (P[1] - B
        [1])
    return k /norm
```

Python Code

```
A = np.array([-4,-6]).reshape(-1,1)
B = np.array([-1,7]).reshape(-1,1)
P = np.array([-34/13 , 0 ]).reshape(-1,1)

norm = norm_sq(P,B)

k = ratio(A,B,P,norm)
k = np.squeeze(k)
print("Ratio = " , k )
```

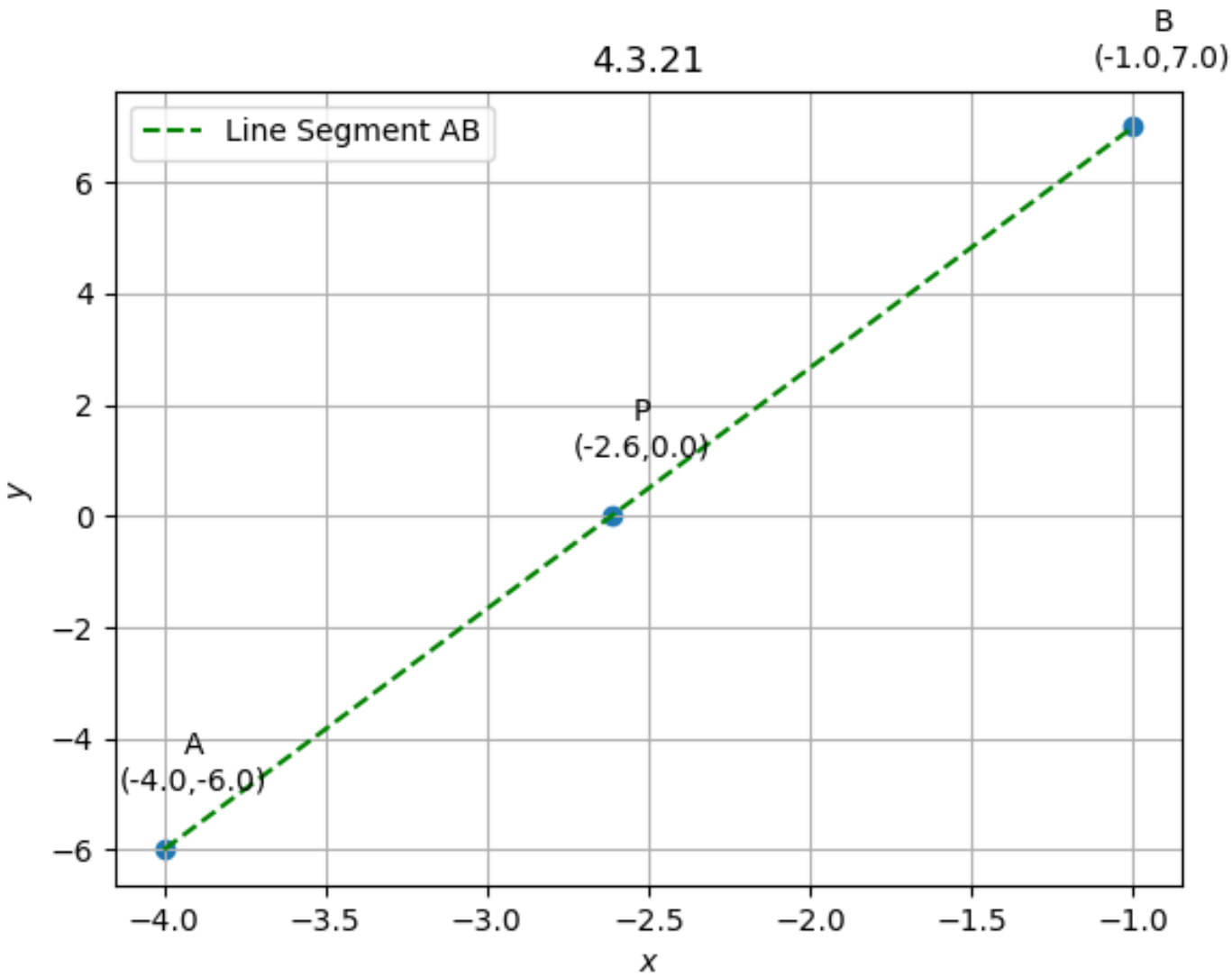
Python Code

```
def plot_it(P,Q,str):
    x_l = line_gen_num(P,Q,20)
    plt.plot(x_l[0,:],x_l[1,:] , str , label = "Line Segment AB"
        )

plt.figure()

plot_it(A,B,"g--")

coords = np.block([[A,B,P]])
plt.scatter(coords[0,:],coords[1,:])
vert_labels = ['A','B','P']
```

Python Code

```
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{txt}\n({coords[0,i]:.1f}, {coords[1,i]:.1f})'
        ,
                 (coords[0,i], coords[1,i]),
                 textcoords="offset points",
                 xytext=(20,15),
                 ha='center')
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.grid()
plt.legend(loc = "best")
plt.title("Fig:4.3.21")
#plt.savefig("../figs/section2.png")
#plt.show()
plt.savefig('../figs/section2.png')
subprocess.run(shlex.split("termux-open ../figs/section2.png"))
```

Plot


4.3.21
B
(-1.0,7.0)
Line Segment AB
P
(-2.6,0.0)
A
(-4.0,-6.0)
6
4
2
0
−2
−4
−6
y
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
x