4.7.24

Kartik Lahoti - EE25BTECH11032

September 12 ,2025

Question

Find the equation of line passing through the point $(5, 2)$ and perpendicular to the line joining the points $(2, 3)$ and $(3, -1)$

Theoretical Solution

Given :

Symbol	Value	Description
A	$\begin{pmatrix} 2 \\ 3 \end{pmatrix}$	Given Point
B	$\begin{pmatrix} 3 \\ -1 \end{pmatrix}$	Given Point
P	$\begin{pmatrix} 5 \\ 2 \end{pmatrix}$	Given Point

Table: 4.7.24

Let , $\mathbf{X}$ be a vector on the Required Line

Theoretical Solution

Direction Vector for the Line **AB** ,

$$\mathbf{B} - \mathbf{A} = \begin{pmatrix} 3 \\ -1 \end{pmatrix} - \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ -4 \end{pmatrix} \tag{1}$$

Direction Vector for the Required Line in terms of **X** ,

$$\mathbf{X} - \mathbf{P} = \left(\mathbf{X} - \begin{pmatrix} 5 \\ 2 \end{pmatrix} \right) \tag{2}$$

Theoretical Solution

Direction Vector for the Line **AB** is perpendicular to the required line

$$\therefore (\mathbf{B}-\mathbf{A})^{\top}\left(\mathbf{X}-\begin{pmatrix}5\\2\end{pmatrix}\right)=0 \tag{3}$$

$$\begin{pmatrix}1 & -4\end{pmatrix}\left(\mathbf{X}-\begin{pmatrix}5\\2\end{pmatrix}\right)=0 \tag{4}$$

Theoretical Solution

Hence, the desired equation is

$$\begin{pmatrix} 1 & -4 \end{pmatrix} \mathbf{X} = -3 \tag{5}$$

C Code (1)

```c
double dot_prod(double *A , double *B , int m )
{
    double sum = 0.0 ;
    for ( int i = 0 ; i < m ; i++ )
    {
        sum += A[i]*B[i] ;
    }
    return sum;
}
```

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 ;

}
```

Python Code - Using Shared Object

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

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

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

handc1.dot_prod.restype = ct.c_double
```

Python Code - Using Shared Object

```
A = np.array([2,3], dtype= np.float64).reshape(-1,1)
B = np.array([3,-1] , dtype= np.float64 ).reshape(-1,1)
P = np.array([5,2] , dtype= np.float64 ).reshape(-1,1)
K = (B-A)
const = handc1.dot_prod(
    K.ctypes.data_as(ct.POINTER(ct.c_double)),
    P.ctypes.data_as(ct.POINTER(ct.c_double)),
    2
)
K = K.reshape(1,-1)
print("Required Line Equation : ")
print(K,"X = ", const )
```

Python Code - Using Shared Object

```
def line_cre(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
    ]
```

Python Code - Using Shared Object

```
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

```
plt.figure()
#the ratio of perp on Line Ab is 7:10
line_cre(P,(10 * A+ 7 * B)/17,"g-","Required Line")
line_cre(A,B,"r--" , "Line AB")

coords = np.block([[A,B,P, (10 * A+ 7 * B)/17 ]])
plt.scatter(coords[0,:],coords[1,:])
vert_labels = ['A','B','P','Q']

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=(25,-12),
                 ha='center', va = 'bottom')
```

Python Code - Using Shared Object

```
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid()

plt.title("Fig:4.7.24")
plt.axis('equal')

plt.savefig("../figs/perpbisector1.png")
plt.show()

#plt.savefig('../figs/perpbisector1.png')
#subprocess.run(shlex.split("termux-open ../figs/perpbisector1.
    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([2,3]).reshape(-1,1)
B = np.array([3,-1]).reshape(-1,1)
P = np.array([5,2]).reshape(-1,1)
K = (B-A).T

x = np.dot(K,P)
x = np.squeeze(x)
print("Required Line Equation : ")
print(K,"X = ", x )
```

Python Code

```
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 )

plt.figure()

plot_it(P,(10 * A+ 7 * B)/17,"g-","Required Line")
plot_it(A,B,"r--" , "Line AB")
```

Python Code

```
coords = np.block([[A,B,P, (10 * A+ 7 * B)/17 ]])
plt.scatter(coords[0,:],coords[1,:])
vert_labels = ['A','B','P','Q']

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=(25,-12),
                 ha='center', va = 'bottom')
```

Python Code

```
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid()

plt.title("Fig:4.7.24")
plt.axis('equal')

plt.savefig("../figs/perpbisector2.png")
plt.show()

#plt.savefig('../figs/perpbisector2.png')
#subprocess.run(shlex.split("termux-open ../figs/perpbisector2.
    png"))
```

Plot

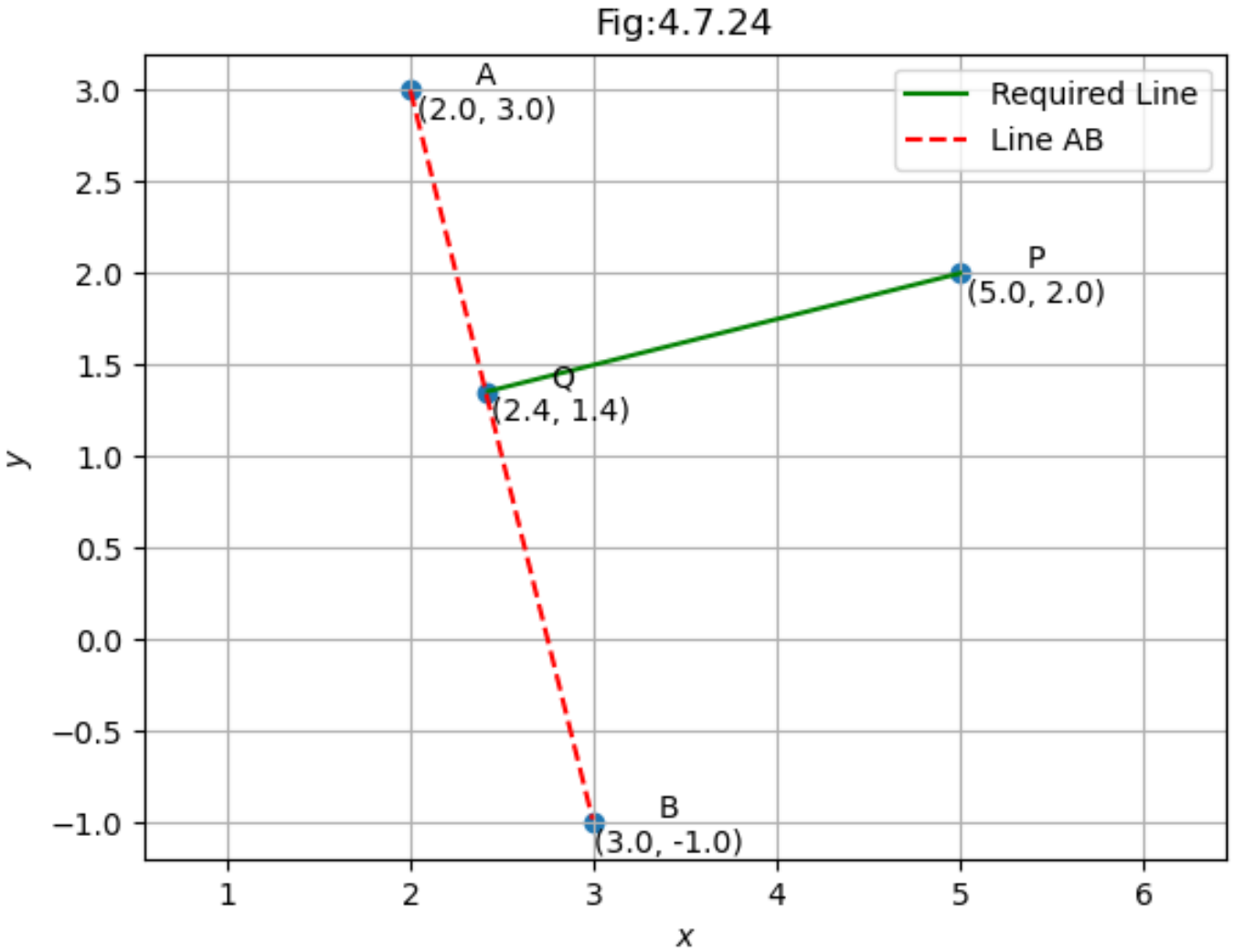
Fig:4.7.24
A
(2.0, 3.0)
Required Line
Line AB
P
(5.0, 2.0)
Q
(2.4, 1.4)
B
(3.0, -1.0)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
y
1
2
3
4
5
6
x