2.4.28

Kartik Lahoti - EE25BTECH11032

August 26,2025

Question

Find the coordinates of the point $\mathbf{Q}$ on the x-axis which lies on the perpendicular bisector of the line segment joining the points $\mathbf{A}\left(-5,-2\right)$ and $\mathbf{B}\left(4,-2\right)$. Name the type of triangle formed by points $\mathbf{Q},\mathbf{A}$ and $\mathbf{B}$.

Theoretical Solution

Given,

$$\mathbf{A} = \begin{pmatrix} -5 \\ -2 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 4 \\ -2 \end{pmatrix} \tag{1}$$

Let $\mathbf{M}$ be the midpoint of $\mathbf{AB}$

Theoretical Solution

$$\mathbf{M} = \frac{1}{2}\left(\mathbf{A} + \mathbf{B}\right) \tag{2}$$

$$= \frac{1}{2}\left(\begin{pmatrix} -5 \\ -2 \end{pmatrix} + \begin{pmatrix} 4 \\ -2 \end{pmatrix}\right) \tag{3}$$

$$= \begin{pmatrix} -0.5 \\ -2 \end{pmatrix} \tag{4}$$

Theoretical Solution

To find the direction vector of perpendicular bisector , we can find the direction vector of **AB** and then rotate it by 90°
Direction Vector of **AB** (represented by $\mathbf{V_{AB}}$):

$$\mathbf{B}-\mathbf{A}=\mathbf{V_{AB}}=\begin{pmatrix}4\\-2\end{pmatrix}-\begin{pmatrix}-5\\-2\end{pmatrix}=\begin{pmatrix}9\\0\end{pmatrix} \tag{5}$$

Rotation Matrix :

$$R\left(\theta\right)=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix} \tag{6}$$

Theoretical Solution

Direction Vector for perpendicular bisector (represented by **V**) :

$$\mathbf{V} = R\left(90^\circ\right)\mathbf{V_{AB}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 9 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 9 \end{pmatrix} \tag{7}$$

Any arbitrary vector on perpendicular bisector can be given by :

$$\mathbf{Q} = \mathbf{M} + t\mathbf{V} \text{ where } t \in \mathbb{R} \tag{8}$$

Theoretical Solution

Finding $\mathbf{Q}$,

$$\mathbf{Q} = \begin{pmatrix} -0.5 \\ -2 \end{pmatrix} + t \begin{pmatrix} 0 \\ 9 \end{pmatrix} \tag{9}$$

$$\mathbf{Q} = \begin{pmatrix} -0.5 \\ -2 + 9t \end{pmatrix} \tag{10}$$

Since y-coordinate of $\mathbf{Q}$ is zero

$$\mathbf{Q} = \begin{pmatrix} -0.5 \\ 0 \end{pmatrix} \tag{11}$$

Theoretical Solution

Since **Q** lies on perpendicular bisector of **AB** , it is equidistant from both **A** and **B**

$$\|\mathbf{Q}-\mathbf{A}\| = \|\mathbf{Q}-\mathbf{B}\| \tag{12}$$

Hence $\triangle ABQ$ is an isosceles triangle.

C Code (1) - Function to find Mid Point of Two given vectors

```c

#include <math.h>
void midpoint(double *A , double *B , double *M , int m )
{

    for ( int i = 0 ; i < m ; i++ )
    {
        M[i] = (A[i]+B[i])/ 2.0 ;
    }

}
```

C Code (2) - Function to rotate a Direction Vector by theta°

```c

void rotate(double *IN , double *OP , double theta )
{
    theta = M_PI / 180.0 * theta ; // converting to radian
    OP[0] = cos(theta)*IN[0] - sin(theta)*IN[1] ;
    OP[1] = sin(theta) * IN[0] + cos(theta) * IN[1] ;
}
```

C Code (3) - Function to generate points on Line

```c
void linegen(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

```python
import ctypes
import numpy as np
import matplotlib.pyplot as plt
handc1 = ctypes.CDLL("./func.so")

handc1.midpoint.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_int
]

handc1.midpoint.restype = None
A = np.array([[-5],[-2]], dtype=np.float64).reshape(-1,1)
B = np.array([[4],[-2]], dtype=np.float64).reshape(-1,1)
M = np.zeros(2,dtype=np.float64).reshape(-1,1)
```

Python Code - Using Shared Object

```
handc1.midpoint (
    A.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    B.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    M.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),2)
AB = np.array([[9],[0]],dtype=np.float64)
theta = 90
handc1.rotate.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_double]

handc1.rotate.restype = None
per = np.zeros(2,dtype=np.float64).reshape(-1,1)
handc1.rotate(AB.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    per.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),theta)

Q = M + 2 / 9 * per
```

Python Code - Using Shared Object

```python
def line_cre(P: np.ndarray , Q: np.ndarray, str):
    handc2 = ctypes.CDLL("./line_gen.so")

    handc2.linegen.argtypes = [
        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
    ]

    handc2.linegen.restype = None
```

Python Code - Using Shared Object

```
n = 200
    X_l = np.zeros(n,dtype=np.float64)
    Y_l = np.zeros(n,dtype=np.float64)

    handc2.linegen (
        X_l.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
        Y_l.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,2
    )
    plt.plot([X_l[0],X_l[-1]],[Y_l[0],Y_l[-1]],str)
```

Python Code - Using Shared Object

```python
plt.figure()
line_cre(A,B,"g-")
line_cre(Q,M,"r-")

coords = np.block([[A,B,M,Q]])
plt.scatter(coords[0,:],coords[1,:])
vert_labels = ['A','B','M','Q']
#for i , txt in enumerate(vert_labels):
#plt.annotate(txt,(coords[0,i],coords[1,i]),textcoords="offset points", xytext=(0,10),ha='center')

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=(20,0),ha='center', va = 'bottom')
```

Python Code - Using Shared Object

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

plt.title("Fig:2.4.28")
plt.axis('equal')

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

#plt.savefig('figs/triangle/ang-bisect.pdf')
#subprocess.run(shlex.split("termux-open figs/triangle/ang-bisect.pdf"))
```

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
import matplotlib.image as mpimg

from line.funcs import *
#from triangle.funcs import *
#from conics.funcs import circ_gen

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

Python Code

```
A = np.array([-5,-2]).reshape(-1,1)
B = np.array([4,-2]).reshape(-1,1)
M = (A+B)/2
AB = np.array([9,0]).reshape(-1,1)
theta = 90
theta = np.deg2rad(theta)
x,y = AB
x_1 = np.cos(theta)*x - np.sin(theta)*y
y_1 = np.sin(theta)*x + np.cos(theta)*y
per = np.array([x_1,y_1]).reshape(-1,1)

Q = M +2/9*per

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

Python Code

```python

plt.figure()
plot_it(A,B,"g-")
plot_it(M,Q,"r-")

coords = np.block([[A,B,M,Q]])
plt.scatter(coords[0,:],coords[1,:])
vert_labels = ['A','B','M','Q']
#for i , txt in enumerate(vert_labels):
 # plt.annotate(txt,(coords[0,i],coords[1,i]),textcoords="offset points", xytext=(0,10),ha='center')
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=(20,0),
                 ha='center',va ='bottom')
```

Python Code

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

plt.title("Fig:2.4.28")
plt.axis('equal')

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

#plt.savefig('figs/triangle/ang-bisect.pdf')
#subprocess.run(shlex.split("termux-open figs/triangle/ang-bisect.pdf"))
```

Plot

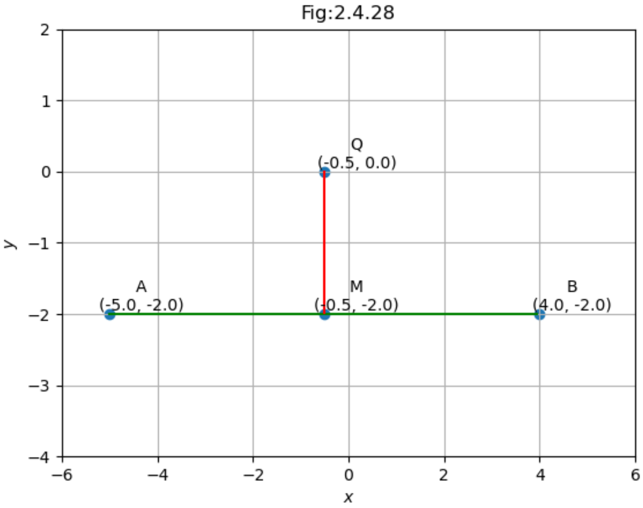
Fig:2.4.28
y
x
Q
(-0.5, 0.0)
A
(-5.0, -2.0)
M
(-0.5, -2.0)
B
(4.0, -2.0)