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

If **Q** lies on the *x*-axis and on the perpendicular bisector of the points **A** and **B**, i.e **Q** is equidistant from points **A** and **B**

$$\left\|\mathbf{Q}-\mathbf{A}\right\| = \left\|\mathbf{Q}-\mathbf{B}\right\| \tag{1}$$

$$\implies \left\|\mathbf{Q}-\mathbf{A}\right\|^2 = \left\|\mathbf{Q}-\mathbf{B}\right\|^2 \tag{2}$$

$$\implies \left\|\mathbf{Q}\right\|^2 - 2\mathbf{Q}^\top\mathbf{A} + \left\|\mathbf{A}\right\|^2 = \left\|\mathbf{Q}\right\|^2 - 2\mathbf{Q}^\top\mathbf{B} + \left\|\mathbf{B}\right\|^2, \tag{3}$$

Theoretical Solution

which can be simplified to obtain,

$$(\mathbf{A}-\mathbf{B})^\top \mathbf{Q} = \frac{\|\mathbf{A}\|^2 - \|\mathbf{B}\|^2}{2}. \tag{4}$$

$$\because \mathbf{Q} = x\mathbf{e}_1, \tag{5}$$

$$x = \frac{\|\mathbf{A}\|^2 - \|\mathbf{B}\|^2}{2(\mathbf{A}-\mathbf{B})^\top \mathbf{e}_1.} \tag{6}$$

Theoretical Solution

$$\|\mathbf{A}\|^2 = 29, \|\mathbf{B}\|^2 = 20 \tag{7}$$

$$(\mathbf{A}-\mathbf{B})^\top = \begin{pmatrix} -9 & 0 \end{pmatrix}, \mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{8}$$

Substituting from (0.7) and (0.8), $x = -0.5$. Thus,

$$\mathbf{Q} = \begin{pmatrix} -0.5 \\ 0 \end{pmatrix}. \tag{9}$$

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

$$\|\mathbf{Q}-\mathbf{A}\| = \|\mathbf{Q}-\mathbf{B}\| \tag{10}$$

Hence $\triangle ABQ$ is an isosceles triangle.

C Code (1) - Function to find norm square

```c
#include <math.h>
double norm_vec_sq(double *A , int m )
{
    double sum = 0.0;
    for ( int i = 0 ; i < m ; i++ )
    {
        sum += pow(A[i] , 2 );
    }
    return sum;
}
```

C Code (2) - Function to x

```c

double x_cal(double *A , double *B , double *E , double na ,
    double nb )
{
    double x , k ;
    k = (A[0]-B[0])*E[0] + (A[1]-B[1])*E[1];
    x = (na - nb) / (2*k);
    return x;
}
```

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")

def norm_cal ( A: np.ndarray , m) :
    handc1.norm_vec_sq.argtypes = [
        ctypes.POINTER(ctypes.c_double),
        ctypes.c_int
    ]

    handc1.norm_vec_sq.restype = ctypes.c_double

    return handc1.norm_vec_sq (
        A.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),m
    )
```

Python Code - Using Shared Object

```
A = np.array([[-5],[-2]], dtype=np.float64).reshape(-1,1)
B = np.array([[4],[-2]], dtype=np.float64).reshape(-1,1)
n1 = norm_cal(A,2)
n2 = norm_cal(B,2)

handc1.x_cal.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_double,
    ctypes.c_double
    ]
```

Python Code - Using Shared Object

```
e = np.array([[1],[0]],dtype=np.float64).reshape(-1,1)
handc1.x_cal.restype = ctypes.c_double

x = handc1.x_cal(
    A.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    B.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    e.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    n1,n2)
Q = np.array([[x],[0]], dtype=np.float64).reshape(-1,1)
```

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,(A+B)/2,"r-")

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

```
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

```python
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

```python

A = np.array([-5,-2]).reshape(-1,1)
B = np.array([4,-2]).reshape(-1,1)
e1 = np.array([1,0]).reshape(-1,1)

x = (LA.norm(A)**2 - LA.norm(B)**2)/(2*np.dot((A-B).T,e1))
#x = float(x)
x = np.squeeze(x)
Q = np.array([[x],[0]],dtype=np.float64).reshape(-1,1)

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

```

plt.figure()
plot_it(A,B,"g-")
plot_it((A+B)/2,Q,"r-")

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

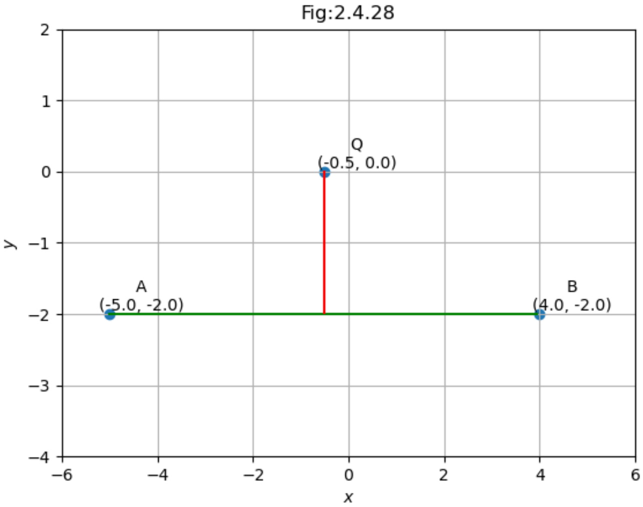
Fig:2.4.28
Q
(-0.5, 0.0)
A
(-5.0, -2.0)
B
(4.0, -2.0)
y
x