

1.9.20

Sai Sreevallabh - EE25BTECH11031

September 5, 2025

Question

Find the point on the Y-Axis which is equidistant from the points $(5, -2)$ and $(-3, 2)$.

Theoretical Solution

Given points are

$$\mathbf{A} = \begin{pmatrix} 5 \\ -2 \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} -3 \\ 2 \end{pmatrix} \quad (1)$$

Let \mathbf{P} be a point on the Y-Axis.

$$\mathbf{P} = \begin{pmatrix} 0 \\ y \end{pmatrix} \quad (2)$$

Theoretical Solution

P is equidistant from both **A** and **B**. Hence the norms of vectors **P** – **B** and **P** – **A** are equal.

$$\|\mathbf{P} - \mathbf{B}\| = \|\mathbf{P} - \mathbf{A}\| \quad (3)$$

$$\implies \|\mathbf{P} - \mathbf{B}\|^2 = \|\mathbf{P} - \mathbf{A}\|^2 \quad (4)$$

$$\implies \|\mathbf{P}\|^2 - 2\mathbf{P}^\top \mathbf{A} + \mathbf{A}^2 = \|\mathbf{P}\|^2 - 2\mathbf{P}^\top \mathbf{B} + \mathbf{B}^2 \quad (5)$$

Theoretical Solution

Simplification of the above results in:

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

$$\therefore \mathbf{P} = y \mathbf{e}_2 \quad (7)$$

$$y = \frac{\|A\|^2 - \|B\|^2}{2(\mathbf{A} - \mathbf{B})^\top \mathbf{e}_2} \quad (8)$$

Theoretical Solution

Substituting the values of **A** and **B**:

$$y = \frac{\left\| \begin{pmatrix} 5 \\ -2 \end{pmatrix} \right\|^2 - \left\| \begin{pmatrix} -3 \\ 2 \end{pmatrix} \right\|^2}{2 \begin{pmatrix} 8 & -4 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}} \quad (9)$$

$$y = -2 \quad (10)$$

∴ The point on the y-axis that is equidistant from the given two points is

$$\mathbf{P} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}.$$

C Code - Function to Find y Coordinate of P

```
#include <stdio.h>
#include <math.h>

double Solve_for_y(double A[2], double B[2]){

    double y = ((pow(A[0],2) + pow(A[1],2)) - (pow(B[0],2) +
        pow(B[1],2)))/(2*(A[1] - B[1]));

    return y;
}
```

Python Code - Using Shared Object

```
import sys
import math
import numpy as np
import matplotlib.pyplot as plt
import numpy.linalg as LA
import ctypes

c_lib=ctypes.CDLL('./code.so')

c_lib.Solve_for_y.argtypes = [
    ctypes.c_double*2,
    ctypes.c_double*2
]

c_lib.Solve_for_y.restype = ctypes.c_double
```


Python Code - Using Shared Object

```
A_c = (ctypes.c_double*2)(5.0, -2.0)
B_c = (ctypes.c_double*2)(-3.0, 2.0)

y = c_lib.Solve_for_y(A_c,B_c)

A = np.array([5,-2]).reshape(-1,1)
B = np.array([-3,2]).reshape(-1,1)

P = np.array([0,y]).reshape(-1,1)

plt.plot([P[0,0], B[0,0]], [P[1,0], B[1,0]], label="Line Segment
$PB$")
plt.plot([P[0,0], A[0,0]], [P[1,0], A[1,0]], label="Line Segment
$PA$")
```

Python Code - Using Shared Object

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

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

for i, txt in enumerate(vert_labels):
    plt.annotate(f'{{{txt}}}({tri_coords[0,i]:.0f}, {tri_coords[1,i]
        :.0f})',
                (tri_coords[0,i], tri_coords[1,i]),
                textcoords="offset points",
                xytext=(20,5),
                ha='center')
```

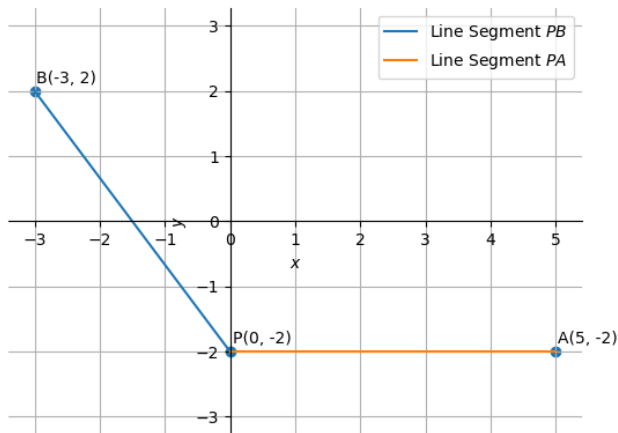
Python Code - Using Shared Object

```
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['bottom'].set_position('zero')
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid()
plt.axis('equal')

plt.savefig("../Figs/plot(py+C).png")

plt.show()
```

Plot-Using Both C and Python



Python Code

```
import sys
import math
sys.path.insert(0, '/home/sai-sreevallabh/Matrix_Theory/Matgeo/
codes/CoordGeo')
import numpy as np
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import numpy.linalg as LA

#local imports
from line.funcs import *
from triangle.funcs import *

#if using termux
import subprocess
import shlex
```

```
A = np.array([5,-2]).reshape(-1,1)
B = np.array([-3,2]).reshape(-1,1)
e_2 = np.array([0,1]).reshape(-1,1)

y = (LA.norm(A)*LA.norm(A) - LA.norm(B)*LA.norm(B))/(2*(A-B).
    T@e_2)

y = y.item()

P = np.array([0,y]).reshape(-1,1)

x_AP = line_gen(A,P)
x_PB = line_gen(P,B)
```

```
plt.plot(x_AP[0,:],x_AP[1:],label='Line Segment $PA$')
plt.plot(x_PB[0,:],x_PB[1:],label='Line Segment $PB$')

tri_coords = np.block([[A,B,P]])
plt.scatter(tri_coords[0,:], tri_coords[1,:])
vert_labels = ['A','B','P']
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{{txt}}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i]:.0f})',
                (tri_coords[0,i], tri_coords[1,i]),
                textcoords="offset points",
                xytext=(20,5),
                ha='center')
```

```
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['bottom'].set_position('zero')

plt.legend(loc='best')
plt.grid()
plt.axis('equal')

plt.savefig("../Figs/plot(py).png")
plt.show()
```


Plot-Using Python only

