2.4.27

Sai Sreevallabh - EE25BTECH11031

September 7, 2025

Question

The perpendicular bisector of the line segment joining the points $\mathbf{A}\,(1, 5)$ and $\mathbf{B}\,(4, 6)$ cuts the y-axis at __________.

Theoretical Solution

Given points are

$$\mathbf{A} = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 4 \\ 6 \end{pmatrix} \tag{1}$$

Let the perpendicular bisector of line segment $\mathbf{B} - \mathbf{A}$ intersect the y-axis at point $\mathbf{P}$.

$$\mathbf{P} = \begin{pmatrix} 0 \\ y \end{pmatrix} \tag{2}$$

Theoretical Solution

All points on the perpendicular bisector of line segment are equidistant from the end points.
Hence, $\mathbf{P}$ is equidistant from both $\mathbf{A}$ and $\mathbf{B}$. So, the norms of vectors $\mathbf{P}-\mathbf{B}$ and $\mathbf{P}-\mathbf{A}$ are equal.

$$\|\mathbf{P}-\mathbf{B}\| = \|\mathbf{P}-\mathbf{A}\| \tag{3}$$

$$\implies \|\mathbf{P}-\mathbf{B}\|^2 = \|\mathbf{P}-\mathbf{A}\|^2 \tag{4}$$

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

Theoretical Solution

Simplification of the above results in:

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

$$\because \mathbf{P} = y\mathbf{e_2} \tag{7}$$

$$y = \frac{\|A\|^2 - \|B\|^2}{2(\mathbf{A}-\mathbf{B})^\top \mathbf{e_2}} \tag{8}$$

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

$$y = \frac{\left\|\begin{pmatrix}1\\5\end{pmatrix}\right\|^2 - \left\|\begin{pmatrix}4\\6\end{pmatrix}\right\|^2}{2\begin{pmatrix}-3 & -1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}} \tag{9}$$

$$y = 13 \tag{10}$$

$\therefore$ The point where the perpendicular bisector of $\mathbf{B} - \mathbf{A}$ intersects the y-axis is the point $\mathbf{P} = \begin{pmatrix}0\\13\end{pmatrix}$.

C Code - Function to Find y Coordinate of P

```c
#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

```python
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 = (ctypes.c_double*2)(1.0,5.0)
B = (ctypes.c_double*2)(4.0,6.0)

y = c_lib.Solve_for_y(A,B)

A = np.array([1,5]).reshape(-1,1)
B = np.array([4,6]).reshape(-1,1)
P = np.array([0,y]).reshape(-1,1)
M = np.array([2.5,5.5]).reshape(-1,1)

plt.plot([A[0,0],B[0,0]],[A[1,0],B[1,0]], label = "Line Segment
    $AB$")
plt.plot([P[0,0],M[0,0]],[P[1,0],M[1,0]], 'o--', label = "
    Perpendicular Bisector")
```

```
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')
```

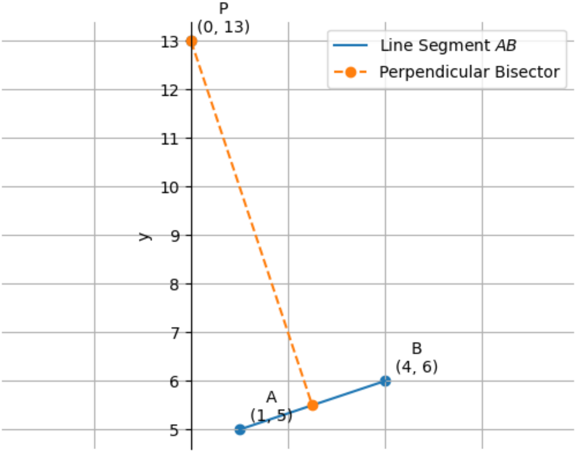
Python Code - Using Shared Object

```python

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

Python Code

```
A = np.array([1,5]).reshape(-1,1)
B = np.array([4,6]).reshape(-1,1)
e_2 = np.array([0,1]).reshape(-1,1)
M = np.array([2.5,5.5]).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)

plt.plot([A[0,0],B[0,0]],[A[1,0],B[1,0]], 'orange', label = "Line
     Segment $AB$")
plt.plot([P[0,0],M[0,0]],[P[1,0],M[1,0]], 'b--', label = "
    Perpendicular Bisector")
```

Python Code

```
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')
```

Python Code

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

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

Plot-Using Python only