1.9.18

Jnanesh Sathisha karmar - EE25BTECH11029

August 31,2025

Question

Find the value of x if the distance between the points $\mathbf{A}\,(0, 0)$ and $\mathbf{B}\,(x, -4)$ is 5 units.

Equation

Distance between 2 vectors **A** and **B** can be represented as:

$$\|AB\| = \sqrt{(\mathbf{B}-\mathbf{A})^T(\mathbf{B}-\mathbf{A})} \tag{1}$$

Theoretical Solution

Given details:

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

$$\mathbf{B} = \begin{pmatrix} x \\ -4 \end{pmatrix} \tag{3}$$

$$\|AB\| = 5 \tag{4}$$

Theoretical Solution

By substituting values:

$$\|AB\| = \sqrt{\begin{pmatrix} x & -4 \end{pmatrix} \begin{pmatrix} x \\ -4 \end{pmatrix}} = \sqrt{x^2 + (-4)^2} = \sqrt{x^2 + 16} \tag{5}$$

Theoretical Solution

Now comparing it with the given distance:

$$\sqrt{x^2 + 16} = 5 \tag{6}$$

Square on both sides

$$x^2 + 16 = 25 \tag{7}$$

$$x^2 = 9 \tag{8}$$

$$x = 3 \text{ or } x = -3 \tag{9}$$

Theoretical Solution

Final answer:

The values of x are 3 and -3

C Code (1) - Function to generate a line segment

```c
#include <math.h>

// Function to calculate distance between two points (x1,y1) and
    (x2,y2)
double distance(double x1, double y1, double x2, double y2) {
    double dx = x2 - x1;
    double dy = y2 - y1;
    return sqrt(dx * dx + dy * dy);
}
```

Python Code - Using Shared Object

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

# Load shared object
lib = ctypes.CDLL("./libline.so")

# Define function signature
lib.distance.argtypes = [ctypes.c_double, ctypes.c_double,
                         ctypes.c_double, ctypes.c_double]
lib.distance.restype = ctypes.c_double
```

Python Code - Using Shared Object

```
# Fixed point A
A = (0, 0)

# Two B points
B1 = (3, -4)
B2 = (-3, 4)

# Distances from A using C function
d1 = lib.distance(A[0], A[1], B1[0], B1[1])
d2 = lib.distance(A[0], A[1], B2[0], B2[1])

# Plot line segments A->B1 and A->B2
plt.figure(figsize=(6,6))
```

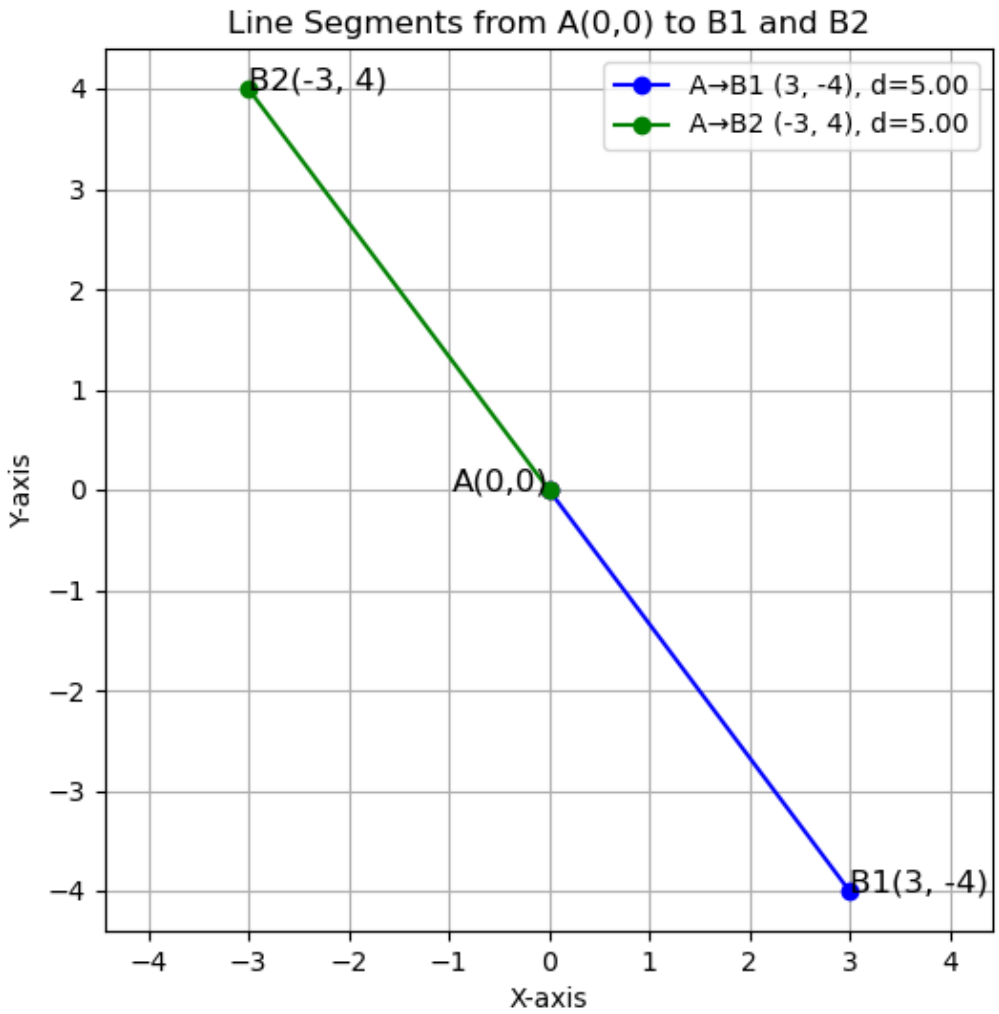
Python Code - Using Shared Object

```
# Line A->B1
plt.plot([A[0], B1[0]], [A[1], B1[1]], 'b-o', label=f"AB1 {B1}, d
    ={d1:.2f}")
plt.text(B1[0], B1[1], f"B1{B1}", fontsize=12, ha='left')

# Line A->B2
plt.plot([A[0], B2[0]], [A[1], B2[1]], 'g-o', label=f"AB2 {B2}, d
    ={d2:.2f}")
plt.text(B2[0], B2[1], f"B2{B2}", fontsize=12, ha='left')

# Mark A
plt.scatter(A[0], A[1], color='red')
plt.text(A[0], A[1], "A(0,0)", fontsize=12, ha='right')
```

Python Code - Using Shared Object

```
# Labels & styling
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Line Segments from A(0,0) to B1 and B2")
plt.legend()
plt.grid(True)
plt.axis("equal")
plt.savefig('figs/distance.png')
9 subprocess.run(shlex.split("termux-open figs/distance.png")
```

Plot-Using Both C and Python



Python Code

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

# Fixed point A
A = (0, 0)

# Two B points
B1 = (3, -4)
B2 = (-3, 4)

# Distances using numpy
d1 = np.sqrt((B1[0]-A[0])**2 + (B1[1]-A[1])**2)
d2 = np.sqrt((B2[0]-A[0])**2 + (B2[1]-A[1])**2)
```

Python Code

```
# Plot line segments A->B1 and A->B2
plt.figure(figsize=(6,6))

# Line A->B1
plt.plot([A[0], B1[0]], [A[1], B1[1]], 'b-o', label=f"AB1 {B1}, d
    ={d1:.2f}")
plt.text(B1[0], B1[1], f"B1{B1}", fontsize=12, ha='left')

# Line A->B2
plt.plot([A[0], B2[0]], [A[1], B2[1]], 'g-o', label=f"AB2 {B2}, d
    ={d2:.2f}")
plt.text(B2[0], B2[1], f"B2{B2}", fontsize=12, ha='left')
```

Python Code

```
# Mark A
plt.scatter(A[0], A[1], color='red')
plt.text(A[0], A[1], "A(0,0)", fontsize=12, ha='right')

# Labels & styling
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Line Segments from A(0,0) to B1 and B2")
plt.legend()
plt.grid(True)
plt.axis("equal")
plt.savefig('figs/distance2.png')
9 subprocess.run(shlex.split("termux-open figs/distance2.png")
```

Plot-Using only Python