

1.9.19

Josyula G S Avaneesh - EE25BTECH11030

August 31,2025

# Question

Find the value of  $x$  for which the distance between the points **A**( $x, 2$ ) and **B**( $9, 8$ ) is 10 units.

# Equation

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

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

# Theoretical Solution

Given details:

$$\mathbf{B} = \begin{pmatrix} 9 \\ 8 \end{pmatrix} \quad (2)$$

$$\mathbf{A} = \begin{pmatrix} x \\ 2 \end{pmatrix} \quad (3)$$

$$\|AB\| = 10 \quad (4)$$

# Theoretical Solution

By substituting values:

$$\|AB\| = \sqrt{\begin{pmatrix} 9-x & 8-2 \end{pmatrix} \begin{pmatrix} 9-x \\ 8-2 \end{pmatrix}} = \sqrt{\begin{pmatrix} 9-x & 6 \end{pmatrix} \begin{pmatrix} 9-x \\ 6 \end{pmatrix}} \quad (5)$$

$$= \sqrt{(9-x)^2 + (6)^2} = \sqrt{(x^2 - 18x + 81) + 36} = \sqrt{x^2 - 18x + 117} \quad (6)$$

# Theoretical Solution

Now from equation (3) and (4), we can say that :

$$\sqrt{x^2 - 18x + 117} = 10 \quad (7)$$

Square on both sides

$$x^2 - 18x + 117 = 100 \quad (8)$$

$$x^2 - 18x + 17 = 0 \quad (9)$$

$$\text{On solving this we get, } x = 1 \text{ or } x = 17 \quad (10)$$

**Final answer:**

The values of  $x$  are 1 and 17. Therefore, the points  $\mathbf{A}(x, 2)$  are  $(1, 2)$  or  $(17, 2)$ .

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

```
#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
import matplotlib as mp
mp.use("TkAgg")

# Load shared object
lib = ctypes.CDLL("./line.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 B
B = (9, 8)

# Two A points
A1 = (1, 2)
A2 = (17,2)

# Distances from B using C function
d1 = lib.distance(B[0], B[1], A1[0], A1[1])
d2 = lib.distance(B[0], B[1], A2[0], A2[1])

# Plot line segments B->A1 and B->A2
plt.figure(figsize=(6,6))
```

# Python Code - Using Shared Object

```
# Line B->A1
plt.plot([B[0], A1[0]], [B[1], A1[1]], 'b-o', label=f"BA1 {A1}, d
        ={d1:.2f}")
plt.text(A1[0], A1[1], f"A1{A1}", fontsize=12, ha='left')

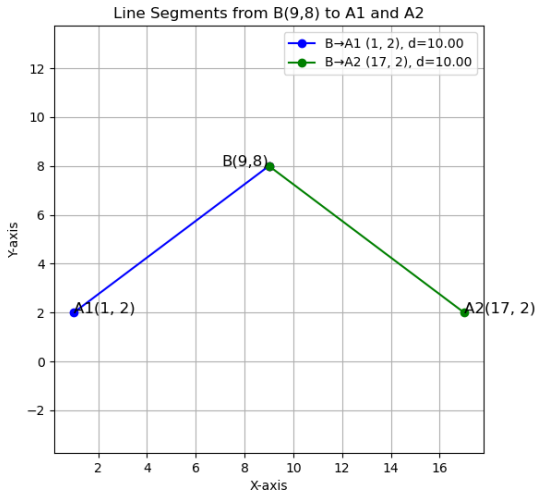
# Line B->A2
plt.plot([B[0], A2[0]], [B[1], A2[1]], 'g-o', label=f"BA2 {A2}, d
        ={d2:.2f}")
plt.text(A2[0], A2[1], f"A2{A2}", fontsize=12, ha='left')

# Mark B
plt.scatter(B[0], B[1], color='red')
plt.text(B[0], B[1], "B(9,8)", fontsize=12, ha='right')
```

# Python Code - Using Shared Object

```
# Labels & styling
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Line Segments from B(9,8) to A1 and A2")
plt.legend()
plt.grid(True)
plt.axis("equal")
plt.savefig('/home/avaneesh1/Matrix/ee1030-2025/ee25btech11030/
            matgeo/1.9.19/figs/distance.png')
plt.show(block=True)
```

# Plot-Using Both C and Python



# Python Code

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mp
mp.use("TkAgg")

# Fixed point B
B = (9, 8)

# Two A points
A1 = (1, 2)
A2 = (17, 2)

# Distances using numpy
d1 = np.sqrt((A1[0]-B[0])**2 + (A1[1]-B[1])**2)
d2 = np.sqrt((A2[0]-B[0])**2 + (A2[1]-B[1])**2)
```

# Python Code

```
# Plot line segments B->A1 and B->A2
plt.figure(figsize=(6,6))

# Line B->A1
plt.plot([B[0], A1[0]], [B[1], A1[1]], 'b-o', label=f"BA1 {A1}", d
         =f"d1:.2f")
plt.text(A1[0], A1[1], f"A1{A1}", fontsize=12, ha='left')

# Line B->A2
plt.plot([B[0], A2[0]], [B[1], A2[1]], 'g-o', label=f"BA2 {A2}", d
         =f"d2:.2f")
plt.text(A2[0], A2[1], f"A2{A2}", fontsize=12, ha='left')
```

```
# Mark B
plt.scatter(B[0], B[1], color='red')
plt.text(B[0], B[1], "B(9,8)", fontsize=12, ha='right')

# Labels & styling
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Line Segments from B(9,8) to A1 and A2")
plt.legend()
plt.grid(True)
plt.axis("equal")
plt.savefig('/home/avaneesh1/Matrix/ee1030-2025/ee25btech11030/
matgeo/1.9.19/figs/distance2.png')
plt.show(block=True)
```



# Plot-Using only Python

