1.9.17

J.NAVYASRI- EE25BTECH11028

August 2025

Question

Write the coordinates of a point **P** on the x-axis which is equidistant from the points **A**$(-\mathbf{2}, \mathbf{0})$ and **B**$(\mathbf{6}, \mathbf{0})$.

Theoretical solution

Let

$$\mathbf{A} = \begin{pmatrix} a \\ 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b \\ 0 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} p \\ 0 \end{pmatrix} \tag{1}$$

Since **P** is equidistant from **A** and **B**, their distances satisfy:

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

Square both sides:

$$\|\mathbf{P} - \mathbf{A}\|^2 = \|\mathbf{P} - \mathbf{B}\|^2 \tag{3}$$

Using the norm squared definition:

$$(\mathbf{P} - \mathbf{A})^\top (\mathbf{P} - \mathbf{A}) = (\mathbf{P} - \mathbf{B})^\top (\mathbf{P} - \mathbf{B}) \tag{4}$$

Expand both sides:

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

Theoretical solution

Cancel $\mathbf{P}^\top\mathbf{P}$ from both sides:

$$-2\mathbf{A}^\top\mathbf{P} + \mathbf{A}^\top\mathbf{A} = -2\mathbf{B}^\top\mathbf{P} + \mathbf{B}^\top\mathbf{B} \tag{6}$$

Rearranged:

$$2(\mathbf{B} - \mathbf{A})^\top\mathbf{P} = \mathbf{B}^\top\mathbf{B} - \mathbf{A}^\top\mathbf{A} \tag{7}$$

Substitute the vectors:

$$2(b - a)p = b^2 - a^2 \tag{8}$$

Rewrite right side as difference of squares:

$$2(b - a)p = (b - a)(b + a) \tag{9}$$

Since $b \neq a$, divide both sides by $(b - a)$:

$$2p = b + a \tag{10}$$

Theoretical solution

Solve for x: Solve for p:

$$p = \frac{a+b}{2} \tag{11}$$

Now substitute $a = -2,\ b = 6$:

$$p = \frac{-2+6}{2} = \frac{4}{2} = 2 \tag{12}$$

Hence, the coordinates of $\mathbf{P}$ are:

$$\boxed{\mathbf{P} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}} \tag{13}$$

Python Code

```python
import matplotlib.pyplot as plt

def setup_plot():
    # Points A, P, and B
    A = (-2, 0)
    P = (2, 0)
    B = (6, 0)

    fig, ax = plt.subplots(figsize=(8, 3))

    # Set axis limits and aspect ratio
    ax.set_xlim(-4, 8)
    ax.set_ylim(-1, 2)
    ax.set_aspect('equal')

    # Remove ticks and spines except left and bottom
    ax.set_xticks([])
    ax.set_yticks([])
    for spine in ['top', 'right']:
```

Python Code

```python
def draw_elements(ax, A, P, B):
    # Plot points
    ax.plot(A[0], A[1], 'ro') # red
    ax.plot(P[0], P[1], 'go') # green
    ax.plot(B[0], B[1], 'bo') # blue

    # Labels below points
    ax.text(A[0], A[1] - 0.25, r'$A(-2,0)$', color='red', ha='center', fontsize=12)
    ax.text(P[0], P[1] - 0.25, r'$P(2,0)$', color='green', ha='center', fontsize=12)
    ax.text(B[0], B[1] - 0.25, r'$B(6,0)$', color='blue', ha='center', fontsize=12)

    # Dashed line between A and B
    ax.plot([A[0], B[0]], [A[1], B[1]], 'k--', linewidth=1)

    # Annotation above dashed line
    ax.text(2, 0.3, r'$P$ is equidistant from $A$ and $B$',
```

Python Code

```
def add_caption_and_show(fig):
    fig.text(0.5, 0.02, 'Fig. 0', ha='center', fontsize=14,
        weight='bold')
    plt.show()

if __name__ == "__main__":
    fig, ax, A, P, B = setup_plot()
    draw_elements(ax, A, P, B)
    add_caption_and_show(fig)
```

C Code

```c
#include <stdio.h>

int main() {
    // Given points A and B on x-axis
    double a = -2.0;
    double b = 6.0;

    // Calculate p = (a + b) / 2
    double p = (a + b) / 2.0;

    printf("Coordinates of P are: (%.2f, 0)\n", p);

    return 0;
}
```

Python and C Code

```python
# Compile the C program
subprocess.run(["gcc", "equidiistance.c", "-o", "equidistance"])

# Run the compiled C program
result = subprocess.run(["./equidistance"], capture_output=True,
    text=True)

# Print the output from the C program
print(result.stdout)
```

Graphical Representation:

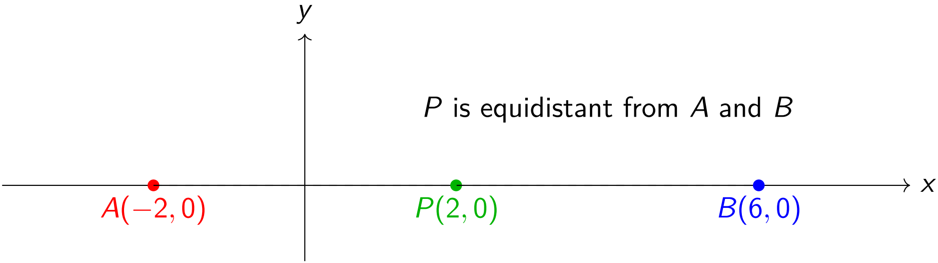

Fig. 0