1.3.9

AI25BTECH11030 - SARVESH TAMGADE

August 30, 2025

Problem Statement

1.3.9 The center of a circle is at $(2, 0)$. If one end of a diameter is at $(6, 0)$, then find the other end.

Solution

Let the center be $\mathbf{C} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$, one end of the diameter $\mathbf{A} = \begin{pmatrix} 6 \\ 0 \end{pmatrix}$, and the other end be $\mathbf{B} = \begin{pmatrix} x \\ y \end{pmatrix}$.

Since the center is the midpoint of the diameter:

$$\mathbf{C} = \frac{\mathbf{A} + \mathbf{B}}{2}$$

Multiply both sides by 2:

$$2\mathbf{C} = \mathbf{A} + \mathbf{B}$$

Rearranging for $\mathbf{B}$:

$$\mathbf{B} = 2\mathbf{C} - \mathbf{A} = 2\begin{pmatrix} 2 \\ 0 \end{pmatrix} - \begin{pmatrix} 6 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \end{pmatrix} - \begin{pmatrix} 6 \\ 0 \end{pmatrix} = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

Answer: The other end of the diameter is at $\begin{pmatrix} -2 \\ 0 \end{pmatrix}$.

Graph

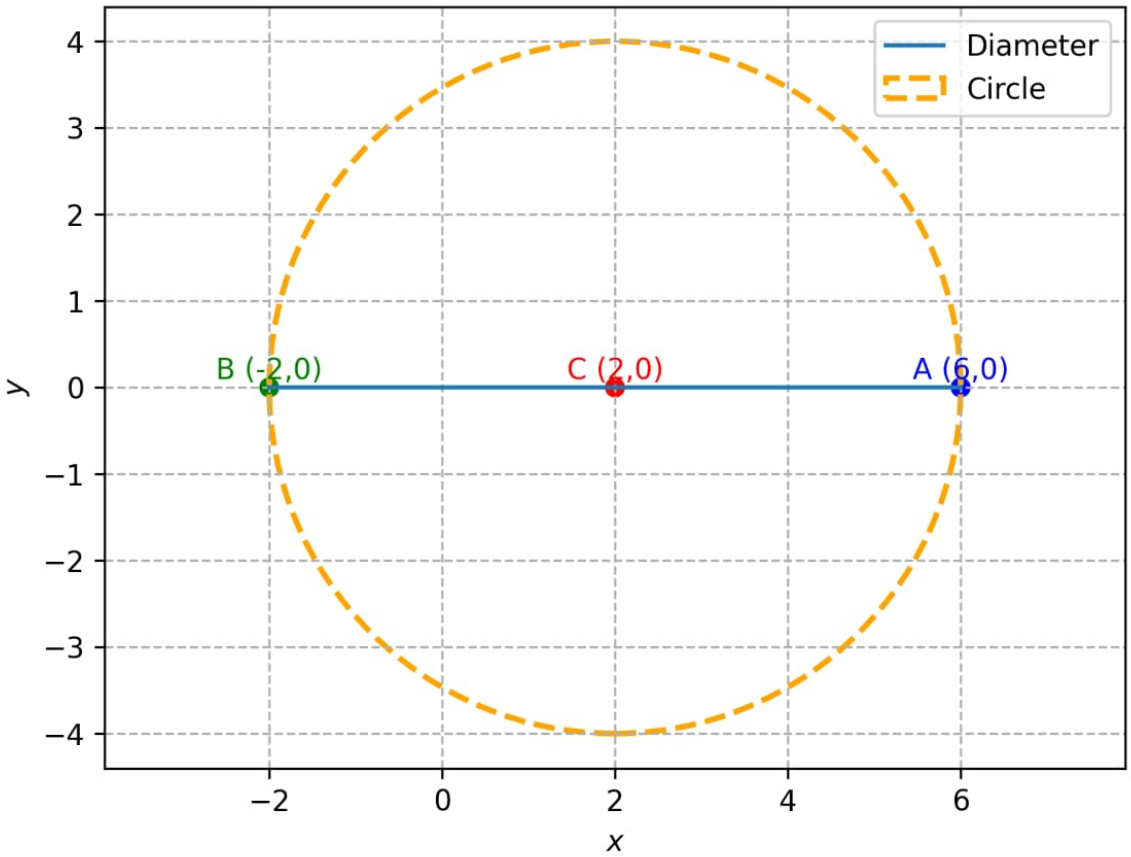

Figure: Diameter of the circle with endpoints $\mathbf{A}(6,0)$ and $\mathbf{B}(-2,0)$, center at $(2,0)$.

C Code

```c
#include <stdio.h>
#include <stdlib.h>
#include "matfun.h"

int main() {
    double **M, **k, **C;
    int cx = 2, cy = 0; // Center of circle
    int ax = 6, ay = 0; // One endpoint of diameter

    // Create matrices: M (2x2), k (2x1), C (2x1)
    M = createMat(2, 2);
    k = createMat(2, 1);

    // Arrange matrix M: columns are points C and A
    M[0][0] = (double)cx; M[1][0] = (double)cy;
    M[0][1] = (double)ax; M[1][1] = (double)ay;

    // Weights vector for B = 2*C - A
    k[0][0] = 2.0;
    k[1][0] = -1.0;
```

C Code

```
    // Calculate B = M * k
    C = Matmul(M, k, 2, 2, 1);

    // Print result B
    printf("Coordinates of other end B = (%.2lf, %.2lf)\n", C[0][0], C
        [1][0]);

    // Free allocated matrices
    freeMat(M, 2);
    freeMat(k, 2);
    freeMat(C, 2);

    return 0;
}
```

Python Plot

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

def line_gen(A, B, num=100):
    """
    Generates points on a line segment between points A and B.
    A, B are 2x1 numpy arrays (column vectors).
    Returns 2 x num numpy array of points.
    """
    lam = np.linspace(0, 1, num)
    return (1 - lam) * A + lam * B

# Points as column vectors
C = np.array([2, 0]).reshape(-1,1) # Center
A = np.array([6, 0]).reshape(-1,1) # One end of diameter
B = 2*C - A # Other end of diameter calculated

coords = np.block([[A,B,C]])

# Generate line points for diameter AB
AB = line_gen(A, B)
```

Python Plot

```
# Plot line AB
plt.plot(AB[0,:], AB[1,:], label='Diameter')

# Plot points
plt.scatter(coords[0,:], coords[1,:], color=['blue', 'green', 'red'])

# Annotations
plt.text(A[0], A[1]+0.1, 'A (6,0)', ha='center', color='blue')
plt.text(B[0], B[1]+0.1, 'B (-2,0)', ha='center', color='green')
plt.text(C[0], C[1]+0.1, 'C (2,0)', ha='center', color='red')

# Draw the circle centered at C with radius = half the distance AB
radius = np.linalg.norm(A - B) / 2

circle = plt.Circle((C[0], C[1]), radius, fill=False, color='orange',
    linestyle='--', linewidth=2, label='Circle')

# Add circle to plot
ax = plt.gca()
ax.add_patch(circle)
```

Python Plot

```
# Labels and grid
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid(True, linestyle='--')
plt.axis('equal')

# Save figure (adjust path as needed)
plt.savefig('circle_diameter_plot_with_circle.png', dpi=300)
plt.show()
```