

1.5.16

J.NAVYASRI- EE25BTECH11028

August 2025

Question

Find the coordinates point A where AB is a diameter of the circle with center $= (3, -1)$ and point $B = (2, 6)$.

given data

Point	Vector
B	$\begin{pmatrix} 2 \\ 6 \end{pmatrix}$
P	$\begin{pmatrix} 3 \\ -1 \end{pmatrix}$

Theory : Center of a circle is the mid-point of the diameter.

Let P be the center of the given circle , with AB as the diameter.

Let \mathbf{A} be the Vector to be found

Given :

$$B \equiv \begin{pmatrix} 2 \\ 6 \end{pmatrix}, \quad P \equiv \begin{pmatrix} 3 \\ -1 \end{pmatrix}$$

Theoretical Solution

Center of a circle is the mid point of the diameter. For a circle with center **P** and ends of diameters represented by vectors **A** and **B**

$$\mathbf{P} = \frac{\mathbf{A} + \mathbf{B}}{2} \quad (0.1)$$

Rearranging , we get:

$$\mathbf{A} = 2\mathbf{P} - \mathbf{B} \quad (0.2)$$

Theoretical Solution

Substituting the given vectors, we get:

$$\mathbf{A} = 2 \begin{pmatrix} 3 \\ -1 \end{pmatrix} - \begin{pmatrix} 2 \\ 6 \end{pmatrix} \quad (0.3)$$

$$\mathbf{A} = \begin{pmatrix} 6 \\ -2 \end{pmatrix} - \begin{pmatrix} 2 \\ 6 \end{pmatrix} \quad (0.4)$$

$$\therefore \mathbf{A} \equiv \begin{pmatrix} 4 \\ -8 \end{pmatrix}$$

Hence , Coordinates of A are

$$\begin{pmatrix} 4 \\ -8 \end{pmatrix}$$

C Code (1) - Function to find A matrix

```
#include <stdio.h>
#include <math.h>
void func(double *P, double *B, double *A , int m )
{
    for ( int i = 0 ; i < m ; i++ )
    {
        A[i] = 2*P[i] - B[i] ;
    }
}
```

C Code (1) - Function to Find Radius

```
double radius(double *P , double *B , int m )
{
    double sum = 0.0;
    for ( int i = 0 ; i < m ; i++ )
    {
        sum += pow(P[i]-B[i] , 2 );
    }
    return sqrt(sum) ;
}
```


C Code (2) - Function to Generate Points on Circle

```
#include <math.h>

void circle_gen(double *X , double *Y , double *P, int n , double
    r)
{
    // n is no. of points to generates. x stores x coor , y stores y
    coor
    for (int i = 0 ; i < n ; i++ )
    {
        double theta = 2.0 * M_PI * i / n ;
        X[i] = P[0] + r * cos(theta);
        Y[i] = P[1] + r * sin (theta);
    }
}
```

C Code (2) - Function to Generate Points on Line

```
void line_gen (double *X, double *Y , double *A , double *B , int
               n , int m )
{
    double temp[m] ;
    for (int i = 0 ; i < m ; i++)
    {
        temp [ i ] = (B[i]- A[i]) /(double) n ;
    }
    for (int i = 0 ; i <= n ; i++ )
    {
        X[i] = A[0] + temp[0] * i ;
        Y[i] = A[1] + temp[1] * i ;
    }
}
```

```
import matplotlib.pyplot as plt

# Center C = (3, -1)
# B = (2, 6)
# Let A = (x, y). Midpoint formula: center = (A + B) / 2 =>
# 3 = (x + 2) / 2, -1 = (y + 6) / 2
# Solve for (x, y):
# x = 2*3 - 2 = 4
# y = 2*(-1) - 6 = -8

A = np.array([4, -8])
B = np.array([2, 6])
C = np.array([3, -1])
```

Python Code

```
# For the circle, radius = distance(center, B)
import numpy as np
def dist(P, Q):
    return np.sqrt((P[0] - Q[0])**2 + (P[1] - Q[1])**2)
radius = dist(C, B)

fig, ax = plt.subplots(figsize=(7,7))

# Plot the circle
circle = plt.Circle(C, radius, color='blue', fill=False,
                    linestyle='dotted', label='Circle')
ax.add_patch(circle)

# Plot A, B, C
ax.scatter(*A, color='red', label='A (unknown, solved)')
ax.scatter(*B, color='green', label='B (2, 6)')
ax.scatter(*C, color='orange', label='Center (3, -1)')
```

```
# Plot line AC
ax.plot([A[0], C[0]], [A[1], C[1]], [A[2], C[2]], color='purple',
        label='Line AC')

# Annotate points
ax.text(*A, ' A', color='red', fontsize=10)
ax.text(*B, ' B', color='green', fontsize=10)
ax.text(*C, ' C', color='blue', fontsize=10)
```

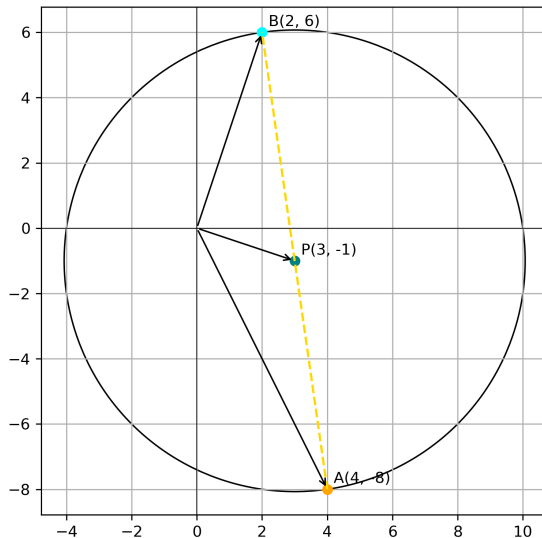
Python Code

```
# Draw diameter AB
ax.plot([A[0], B[0]], [A[1], B[1]], color='purple', linewidth=2,
        linestyle='--', label='Diameter AB')

# Annotate
ax.annotate('A'+str(A), (A[0], A[1]), xytext=(10, -10),
            textcoords='offset points')
ax.annotate('B'+str(B), (B[0], B[1]), xytext=(-40, 10),
            textcoords='offset points')
ax.annotate('C'+str(C), (C[0], C[1]), xytext=(5, -10), textcoords
            ='offset points')

ax.set_xlim(C[0] - radius - 2, C[0] + radius + 2)
ax.set_ylim(C[1] - radius - 2, C[1] + radius + 2)
ax.set_aspect('equal')
ax.grid(True)
plt.legend()
plt.title('Circle with Diameter AB')
plt.xlabel('x')
```

Plot-Using Both C and Python



```
import subprocess

# Compile the C program
subprocess.run(["gcc", "midpoint.c", "-o", "midpoint"])

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

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


Graph

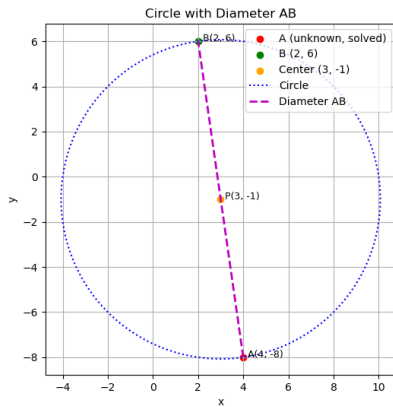


Figure: