

1.5.16

ai24btech11035 - V.Preethika

Question

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

Solution I

Point	Value	Description
C	$(3, -1)$	Centre of the circle
B	$(2, 6)$	Given point B
A	(x, y)	Coordinates of A

Table: Variables Used

$$\mathbf{C} = \frac{\mathbf{A} + \mathbf{B}}{2} \quad (1)$$

$$= \frac{\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 2 \\ 6 \end{pmatrix}}{2} \quad (2)$$

$$= \begin{pmatrix} \frac{x+2}{2} \\ \frac{y+6}{2} \end{pmatrix} \quad (3)$$

Solution II

Given the centre of the circle **C** is (3,-1),we can write

$$\left(\frac{\frac{x+2}{2}}{\frac{y+6}{2}}\right) = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \quad (4)$$

By solving this we get:

$$x = 4 \quad (5)$$

$$y = -8 \quad (6)$$

Therefore, the coordinates of point **A** are (4, -8).

C Code I

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>    // For cos, sin, sqrt, etc.
4 #include "libs/matfun.h"
5 #include "libs/geofun.h"
6
7 int main() {
8     double **A = createMat(2, 1);
9     double **B = createMat(2, 1);
10
11     A[0][0] = 4;
12     A[1][0] = -8;
13
14     B[0][0] = 2;
15     B[1][0] = 6;
16
17     double **C = Matsec(A, B, 2, 1.0);
18
19     printf("The center of the circle (midpoint of A and B) is: \n");
```

```
20     printMat(C, 2, 1);  
21  
22     free(A);  
23     free(B);  
24     free(C);  
25  
26     return 0;  
27 }
```

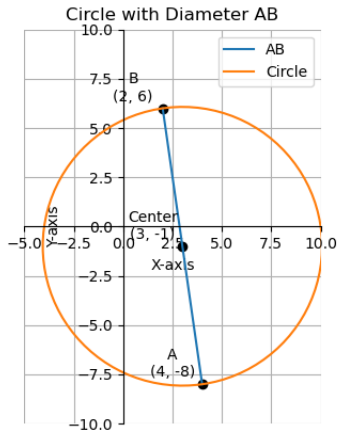


Figure: Plot of $y(x)$

Python Code I

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from numpy import linalg as LA
4
5 # Define points A, B, and the center
6 A = np.array([4, -8]).reshape(-1,1)
7 B = np.array([2, 6]).reshape(-1,1)
8 C = np.array([3, -1]).reshape(-1,1) # center (midpoint of A and
   B)
9
10 # Calculate the radius (half of AB)
11 r = LA.norm(B - A) / 2
12
13 # Function to generate circle points
14 def circ_gen(C, r):
15     theta = np.linspace(0, 2*np.pi, 100)
16     x_circ = np.zeros((2,100))
```


Python Code II

```
17 x_circ[0,:] = r * np.cos(theta)
18 x_circ[1,:] = r * np.sin(theta)
19 x_circ = (x_circ.T + C.T).T
20 return x_circ
21
22 # Line generation between two points
23 def line_gen(A, B):
24     len = 100
25     dim = A.shape[0]
26     x_AB = np.zeros((dim, len))
27     lam_1 = np.linspace(0, 1, len)
28     for i in range(len):
29         temp = A + lam_1[i] * (B - A)
30         x_AB[:, i] = temp.T
31     return x_AB
32
33 # Generate the circle and line
```

Python Code III

```
34 x_circ = circ_gen(C, r)
35 x_AB = line_gen(A, B)
36
37 # Plotting the circle and line
38 plt.plot(x_AB[0,:], x_AB[1,:], label='AB')
39 plt.plot(x_circ[0,:], x_circ[1,:], label='Circle')
40
41 # Scatter and label the points
42 tri_coords = np.block([A, B, C])
43 plt.scatter(tri_coords[0,:], tri_coords[1:], color='black')
44 vert_labels = ['A', 'B', 'Center']
45 for i, txt in enumerate(vert_labels):
46     plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f},\n{tri_coords[1,i]:.0f})',
47                 (tri_coords[0,i], tri_coords[1,i]),
48                 textcoords="offset points",
49                 xytext=(-20,5),
```

Python Code IV

```
50         ha='center')
51
52 # Setting axis properties to match the diagram
53 plt.xlim(-5, 10)
54 plt.ylim(-10, 10)
55 plt.grid() # Minor grid
56
57 # Set equal scaling for x and y axes to ensure the circle
    doesn't appear as an ellipse
58 plt.gca().set_aspect('equal', adjustable='box')
59
60 # Removing the top and right spines
61 ax = plt.gca()
62 ax.spines['top'].set_color('none')
63 ax.spines['right'].set_color('none')
64
65 # Moving the left and bottom spines to the center
```

Python Code V

```
66 ax.spines['left'].set_position('zero')
67 ax.spines['bottom'].set_position('zero')
68
69 # Add labels to the axes
70 plt.xlabel('X-axis')
71 plt.ylabel('Y-axis')
72
73 # Set the title and show the legend
74 plt.title('Circle with Diameter AB')
75 plt.legend(loc='best')
76
77 # Show the plot
78 plt.savefig('plot.png')
```