

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 $(2, 6)$.

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

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

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

$$= \left(\frac{x+2}{2}, \frac{y+6}{2} \right) \quad (3)$$

Solution II

Given the centre of the circle \vec{C} is $(3, -1)$, we can write

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

By solving this two equations we get:

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

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

Therefore, the coordinates of point \vec{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 <string.h>
5 #include <unistd.h>
6 #include <sys/socket.h>
7 #include <netinet/in.h>
8 #include "libs/matfun.h"
9 #include "libs/geofun.h"
10
11 int main() {
12     double **A = createMat(2, 1);
13     double **B = createMat(2, 1);
14
15     A[0][0] = 4;
16     A[1][0] = -8;
17
18     B[0][0] = 2;
19     B[1][0] = 6;
```

```
20
21     double **C = Matsec(A, B, 2, 1.0);
22
23     printf("The center of the circle (midpoint of A and B)
24 is: \n");
25     printMat(C, 2, 1);
26
27     free(A);
28     free(B);
29     free(C);
30
31     return 0;
32 }
```

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:],
44            color='black')
45 vert_labels = ['A', 'B', 'Center']
46 for i, txt in enumerate(vert_labels):
47     plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f},
48                  {tri_coords[1,i]:.0f})',
49                 (tri_coords[0,i], tri_coords[1,i]),
50                 textcoords="offset points",
```

Python Code IV

```
49         xytext=(-20,5),
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
58     circle doesn't appear as an ellipse
59 plt.gca().set_aspect('equal', adjustable='box')
60
61 # Removing the top and right spines
62 ax = plt.gca()
63 ax.spines['top'].set_color('none')
64 ax.spines['right'].set_color('none')
```

Python Code V

```
55 # Moving the left and bottom spines to the center
56 ax.spines['left'].set_position('zero')
57 ax.spines['bottom'].set_position('zero')
58
59 # Add labels to the axes
60 plt.xlabel('X-axis')
61 plt.ylabel('Y-axis')
62
63 # Set the title and show the legend
64 plt.title('Circle with Diameter AB')
65 plt.legend(loc='best')
66
67 # Show the plot
68 plt.savefig('circle.png')
```

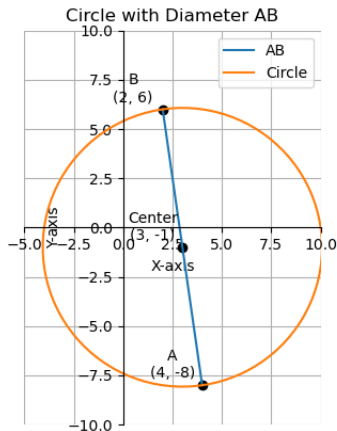


Figure: Plot of $y(x)$