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 cicle
B	$(2, 6)$	Given point B
A	(x, y)	Coordinates of A

Table: Variables Used

$$\mathbf{C} = \frac{\mathbf{A} + \mathbf{B}}{2} \tag{1}$$

$$= \frac{\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 2 \\ 6 \end{pmatrix}}{2} \tag{2}$$

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

Solution II

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

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

By solving this two equations we get:

$$x=4 \tag{5}$$
$$y=-8 \tag{6}$$

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

C Code I

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>   // For cos, sin, sqrt, etc.
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include "libs/matfun.h"
#include "libs/geofun.h"

int main() {
    double **A = createMat(2, 1);
    double **B = createMat(2, 1);

    A[0][0] = 4;
    A[1][0] = -8;

    B[0][0] = 2;
    B[1][0] = 6;
```

C Code II

```c

    double **C = Matsec(A, B, 2, 1.0);

    printf("The center of the circle (midpoint of A and B) is: \n");
    printMat(C, 2, 1);

    free(A);
    free(B);
    free(C);

    return 0;
}
```

Plot

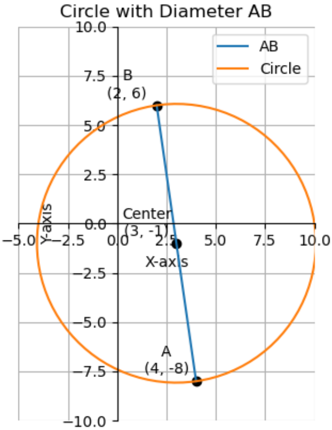

Figure: Plot of $y(x)$

Python Code I

```python
import numpy as np
import matplotlib.pyplot as plt
from numpy import linalg as LA

# Define points A, B, and the center
A = np.array([4, -8]).reshape(-1,1)
B = np.array([2, 6]).reshape(-1,1)
C = np.array([3, -1]).reshape(-1,1)  # center (midpoint of A and B)

# Calculate the radius (half of AB)
r = LA.norm(B - A) / 2

# Function to generate circle points
def circ_gen(C, r):
    theta = np.linspace(0, 2*np.pi, 100)
    x_circ = np.zeros((2,100))
```

Python Code II

```python
    x_circ[0,:] = r * np.cos(theta)
    x_circ[1,:] = r * np.sin(theta)
    x_circ = (x_circ.T + C.T).T
    return x_circ

# Line generation between two points
def line_gen(A, B):
    len = 100
    dim = A.shape[0]
    x_AB = np.zeros((dim, len))
    lam_1 = np.linspace(0, 1, len)
    for i in range(len):
        temp = A + lam_1[i] * (B - A)
        x_AB[:, i] = temp.T
    return x_AB

# Generate the circle and line
```

Python Code III

```python
x_circ = circ_gen(C, r)
x_AB = line_gen(A, B)

# Plotting the circle and line
plt.plot(x_AB[0,:], x_AB[1,:], label='AB')
plt.plot(x_circ[0,:], x_circ[1,:], label='Circle')

# Scatter and label the points
tri_coords = np.block([A, B, C])
plt.scatter(tri_coords[0,:], tri_coords[1,:], color='black')
vert_labels = ['A', 'B', 'Center']
for i, txt in enumerate(vert_labels):
    plt.annotate(f'{txt}\n({tri_coords[0,i]:.0f}, {tri_coords[1,i]:.0f})',
                 (tri_coords[0,i], tri_coords[1,i]),
                 textcoords="offset points",
                 xytext=(-20,5),
```

Python Code IV

```
                  ha='center')

# Setting axis properties to match the diagram
plt.xlim(-5, 10)
plt.ylim(-10, 10)
plt.grid()  # Minor grid

# Set equal scaling for x and y axes to ensure the circle doesn't appear as an ellipse
plt.gca().set_aspect('equal', adjustable='box')

# Removing the top and right spines
ax = plt.gca()
ax.spines['top'].set_color('none')
ax.spines['right'].set_color('none')

# Moving the left and bottom spines to the center
```

Python Code V

```
ax.spines[’left’].set_position(’zero’)
ax.spines[’bottom’].set_position(’zero’)

# Add labels to the axes
plt.xlabel(’X-axis’)
plt.ylabel(’Y-axis’)

# Set the title and show the legend
plt.title(’Circle with Diameter AB’)
plt.legend(loc=’best’)

# Show the plot
plt.savefig(’plot.png’)
```