

7-7.2-25

EE24BTECH11027 - G.V.Satwika

# Problem statement

Find the equation of a circle passing through the point  $(7, 3)$  having radius 3 units and whose centre lies on the line  $y = x - 1$ .

# Solution I

Variables	Description	Value
$P$	point vector	$\begin{pmatrix} 7 \\ 3 \end{pmatrix}$
$-u$	centre of the circle	$\begin{pmatrix} k \\ k - 1 \end{pmatrix}$
$r$	radius of the circle	3 units

Table: Variables

## Solution II

From the given information, the following equations can be formulated using the circle equation

$$\|x\|^2 + 2u^\top x + f = 0 \quad (1)$$

$$\|P\|^2 + 2u^\top P + f = 0 \quad (2)$$

$$\|u\|^2 - f = r^2 \quad (3)$$

From (2) and (3)

$$\|P\|^2 + 2u^\top P + \|u\|^2 = r^2 \quad (4)$$

Substituting the values of  $u, P$  and  $r$ ,

$$2k^2 - 2k + 1 + 6 - 20k + 7^2 + 3^2 - 3^2 = 0 \quad (5)$$

$$2k^2 - 22k + 56 = 0 \quad (6)$$

$$k = 7, 4 \quad (7)$$

resulting in circles with centres

$$-u = \begin{pmatrix} 7 \\ 6 \end{pmatrix} \text{ or } \begin{pmatrix} 4 \\ 3 \end{pmatrix} \quad (8)$$

# Plot

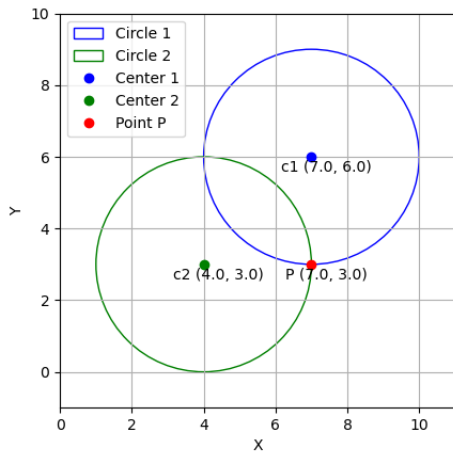


Figure: Plot of circles

# C Code I

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include "/home/g-v-satwika/matgeo/codes/msoft/libs/matfun.h"
5 #include "/home/g-v-satwika/matgeo/codes/msoft/libs/geofun.h"
6
7 int main() {
8     double **P = createMat(1, 2); // Create a point vector for P
9     P[0][0] = 7; // x-coordinate of point P
10    P[0][1] = 3; // y-coordinate of point P
11
12    double r = 5; // Given radius
13    // Final equation:  $2k^2 - 22k + 40 = 0$ 
14    double **CM=createMat(3,1); //coefficient matrix
15    CM[0][0]=2; // Coefficient for  $k^2$ 
16    CM[1][0] = -22; // Coefficient for k
17    CM[2][0] = 56; //constant term
18
19    // Solve the quadratic equation to get two values of k
```

# C Code II

```
20 double **K=createMat(2,1);
21 K=Matquad(CM[0][0],CM[1][0],CM[2][0]);
22 double k1=K[0][0];
23 double k2=K[1][0];
24 // Create two separate vectors for the two centers
25 double **center1 = createMat(1, 2);
26 double **center2 = createMat(1, 2);
27
28 center1[0][0] = k1; // Center 1: (k1, k1-1)
29 center1[0][1] = k1-1;
30
31 center2[0][0] = k2; // Center 2: (k2, k2-1)
32 center2[0][1] = k2-1;
33
34 // Write the centers to a file
35 FILE *file = fopen("circle.txt", "w");
36 if (file == NULL) {
37     fprintf(stderr, "Error opening file for writing.\n");
38     return 1;
```



# C Code III

```
39 }  
40 // Write the two centers to the file  
41 fprintf(file, "C1(%.2f, %.2f)\n", center1[0][0], center1[0][1]);  
42 fprintf(file, "C2(%.2f, %.2f)\n", center2[0][0], center2[0][1]);  
43 fprintf(file, "P(7.00,3.00)");  
44 fclose(file);  
45 // Free the allocated memory  
46 freeMat(center1,1);  
47 freeMat(center2,1);  
48 freeMat(P,1);  
49 freeMat(CM,3);  
50 freeMat(K,2);  
51 return 0;  
52 }
```

# Python Code I

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Function to extract points from the file
5 def read_points_from_file(filename):
6     points = []
7     with open(filename, 'r') as file:
8         for line in file:
9             # Split line by parentheses and commas
10             line = line.strip()
11             point = line[line.find('(')+1 :
12 line.find(')')].split(',')
13             points.append((float(point[0]), float(point[1])))
14     return points
15
16 # Function to plot a circle given the center and radius
17 def plot_circle(ax, center, radius, color, label):
```

# Python Code II

```
17 circle = plt.Circle(center, radius, color=color, fill=False,  
18 label=label)  
19 ax.add_artist(circle)  
20 # Function to label a point with its coordinates  
21 def label_point(ax, point, label):  
22     ax.annotate(f'{label} {point}', (point[0], point[1]),  
23               textcoords="offset points", xytext=(10,-10), ha='center')  
24 # Reading points from 'circle.txt'  
25 points = read_points_from_file('circle.txt')  
26 c1 = points[0]    # First center point  
27 c2 = points[1]    # Second center point  
28 p = points[2]     # Point on the circle  
29  
30 # Given radius  
31 radius = 3
```

# Python Code III

```
32
33 # Create the plot
34 fig, ax = plt.subplots()
35
36 # Plot the circles
37 plot_circle(ax, c1, radius, 'blue', 'Circle 1')
38 plot_circle(ax, c2, radius, 'green', 'Circle 2')
39
40 # Plot the points c1, c2, and p
41 ax.plot(c1[0], c1[1], 'bo', label='Center 1')
42 ax.plot(c2[0], c2[1], 'go', label='Center 2')
43 ax.plot(p[0], p[1], 'ro', label='Point P')
44
45 # Label the points with their coordinates
46 label_point(ax, c1, 'c1')
47 label_point(ax, c2, 'c2')
48 label_point(ax, p, 'P')
```

# Python Code IV

```
49
50 # Set plot limits to ensure the full circles are shown
51 # Find the min and max coordinates to include the entire circle
52 all_x = [c1[0] - radius, c1[0] + radius, c2[0] - radius, c2[0] +
          radius]
53 all_y = [c1[1] - radius, c1[1] + radius, c2[1] - radius, c2[1] +
          radius]
54
55 # Set limits with some padding
56 ax.set_xlim(min(all_x) - 1, max(all_x) + 1)
57 ax.set_ylim(min(all_y) - 1, max(all_y) + 1)
58
59 # Set equal scaling and labels
60 ax.set_aspect('equal', 'box')
61 ax.set_xlabel('X')
62 ax.set_ylabel('Y')
63 ax.legend()
```

```
64  
65  
66 # Show the plot  
67 plt.grid(True)  
68 plt.savefig("circle_plot.png");  
69 plt.show()
```