7.4.34

INDHIRESH S - EE25BTECH11027

30 September, 2025

Question

Find the equation of the circle whose radius is 5 and which touches the circle $x^2 + y^2 - 2x - 4y - 20 = 0$ at the point $(5, 5)$.

Equation I

The general circle equation can be given as:

$$\|\mathbf{x}\|^2 + 2\mathbf{u}^{\mathbf{T}}\mathbf{x} + f = 0 \tag{1}$$

Let the equation of the first circle be

$$\|\mathbf{x}\|^2 + 2\mathbf{u_1^T}\mathbf{x} + f_1 = 0 \tag{2}$$

The equation of the second circle be:

$$\|\mathbf{x}\|^2 + 2\mathbf{u_2^T}\mathbf{x} + f_2 = 0 \tag{3}$$

Theoretical Solution

From the given information:

$$\mathbf{u_1} = \begin{pmatrix} -1 \\ -2 \end{pmatrix} \quad \textit{and} \quad f_1 = -20 \tag{4}$$

Let $\mathbf{c_1}$ and $\mathbf{c_2}$ be the centre of the circle 1 and circle 2:

$$\mathbf{c_1} = -\mathbf{u_1} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \tag{5}$$

And also

$$f_1 = \|\mathbf{u_1}\|^2 - r_1 \tag{6}$$

$$r_1 = 5 \tag{7}$$

Theoretical solution

Let P be the point of contact

$$\mathbf{P} = \begin{pmatrix} 5 \\ 5 \end{pmatrix} \tag{8}$$

If the two circle touch each other externally at $\mathbf{P}$. So the $\mathbf{P}, \mathbf{c_1}$ and $\mathbf{c_2}$ will be collinear and $\mathbf{P}$ will divide the points $\mathbf{c_1}$ and $\mathbf{c_2}$ in the ratio $\frac{r_1}{r_2} : 1$

$$\frac{r_1}{r_2} = 1 \tag{9}$$

$$\mathbf{P} = \frac{\mathbf{c_1} + \mathbf{c_2}}{2} \tag{10}$$

$$\begin{pmatrix} 5 \\ 5 \end{pmatrix} = \frac{\begin{pmatrix} 1 \\ 2 \end{pmatrix} + \mathbf{c_2}}{2} \tag{11}$$

Theoretical Solution

$$\mathbf{c_2} = \begin{pmatrix} 9 \\ 8 \end{pmatrix} \tag{12}$$

If two circles touch each other internally we get:

$$\mathbf{c_2} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \tag{13}$$

This is the same as circle 1, so the two circles touch each other externally. Now

$$r_2 = 5 \;\; and \;\; \mathbf{c_2} = \begin{pmatrix} 9 \\ 8 \end{pmatrix} \tag{14}$$

$$\mathbf{u_2} = -\mathbf{c_2} = \begin{pmatrix} -9 \\ -8 \end{pmatrix} \tag{15}$$

Theoretical Solution

$$f_2 = \|\mathbf{u_2}\|^2 - r_2 \tag{16}$$

$$f_2 = 120 \tag{17}$$

Now the required equation of circle is

$$\|\mathbf{x}\|^2 + 2\mathbf{u_2^T x} + f_2 = 0 \tag{18}$$

$$\|\mathbf{x}\|^2 + 2\begin{pmatrix} -9 \\ -8 \end{pmatrix}^T x + 120 = 0 \tag{19}$$

C Code

```
    #include <math.h>

typedef struct {
    double x;
    double y;
} Point;

void solve_circle_tangency(double G1, double H1, double K1, Point
     p, double r2, Point* c1_center_out, Point* c2_center_out) {
    double c1_x = -G1 / 2.0;
    double c1_y = -H1 / 2.0;
    double m_x = p.x - c1_x;
    double m_y = p.y - c1_y;
    double mag_m = sqrt(m_x * m_x + m_y * m_y);
    double m_hat_x = m_x / mag_m;
    double m_hat_y = m_y / mag_m;
```

C Code

```
        c1_center_out->x = c1_x;
    c1_center_out->y = c1_y;
    c2_center_out->x = p.x + r2 * m_hat_x;
    c2_center_out->y = p.y + r2 * m_hat_y;
}
```

Python Code

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt

# Define a ctypes Structure that mirrors the C struct
class Point(ctypes.Structure):
    _fields_ = [(x, ctypes.c_double),
                (y, ctypes.c_double)]

# --- Load the C library ---
c_lib = ctypes.CDLL('./circle.so')

# --- Define the C function's argument and return types ---
c_lib.solve_circle_tangency.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    Point, ctypes.c_double,
    ctypes.POINTER(Point), ctypes.POINTER(Point)
]
```

Python Code

```
   c_lib.solve_circle_tangency.restype = None

# --- Numerical Inputs are defined in Python using NumPy ---
G1, H1, K1 = -2.0, -4.0, -20.0
p_tangency_np = np.array([5.0, 5.0])
r1, r2 = 5.0, 5.0
# ---------------------------------------------------------

# --- Prepare data for C function call ---
# 1. Convert NumPy array to the ctypes Point structure
p_tangency_ctypes = Point(x=p_tangency_np[0], y=p_tangency_np[1])

# 2. Prepare empty ctypes structures to receive the output
c1_center_ctypes = Point()
c2_center_ctypes = Point()
```

Python Code

```
   # --- Call the C function ---
c_lib.solve_circle_tangency(
    G1, H1, K1, p_tangency_ctypes, r2,
    ctypes.byref(c1_center_ctypes),
    ctypes.byref(c2_center_ctypes)
)

# --- Convert results from ctypes back to NumPy arrays ---
c1 = np.array([c1_center_ctypes.x, c1_center_ctypes.y])
c2 = np.array([c2_center_ctypes.x, c2_center_ctypes.y])
p = p_tangency_np # Use the original numpy array

# --- Plotting with NumPy arrays ---
fig, ax = plt.subplots(figsize=(8, 8))
```

Python Code

```python
    # Plot the circles
circle1 = plt.Circle(c1, r1, fill=False, edgecolor='gray', 
    linestyle='--')
circle2 = plt.Circle(c2, r2, fill=False, edgecolor='gray', 
    linestyle='-')
ax.add_patch(circle1)
ax.add_patch(circle2)

# Plot the line connecting centers
ax.plot([c1[0], c2[0]], [c1[1], c2[1]], 'b-', label='Line of 
    Centers')

# Plot the key points
ax.plot(c1[0], c1[1], 'ro', markersize=10, label=f'C1({c1[0]:.2f
    }, {c1[1]:.2f})')
ax.plot(c2[0], c2[1], 'go', markersize=10, label=f'C2({c2[0]:.2f
    }, {c2[1]:.2f})')
```

Python Code

```
    ax.plot(p[0], p[1], 'm*', markersize=15, label=f'P({p[0]:.2f}, {p[1]:.2f})')

ax.text(c1[0] + 0.2, c1[1] + 0.2, 'C1', fontsize=12, color='red', fontweight='bold')
ax.text(c2[0] + 0.2, c2[1] + 0.2, 'C2', fontsize=12, color='green', fontweight='bold')
ax.text(p[0] + 0.2, p[1] + 0.2, 'P', fontsize=12, color='magenta', fontweight='bold')

# Formatting
ax.set_title('Figure')
ax.grid(True)
ax.axis('equal')
ax.legend()
plt.savefig(/media/indhiresh-s/New Volume/Matrix/ee1030-2025/ee25btech11027/MATGEO/7.4.34/figs/figure1.png)
plt.show()
```

Plot

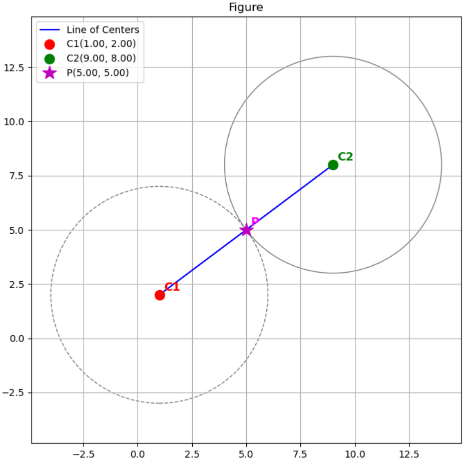

Figure
Line of Centers
C1(1.00, 2.00)
C2(9.00, 8.00)
P(5.00, 5.00)
C1
C2
P
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−2.5
0.0
2.5
5.0
7.5
10.0
12.5