10.7.89

Naman Kumar-EE25BTECH11041

5 Oct,2025

Question)

Lines $5x + 12y - 10 = 0$ and $5x - 12y - 40 = 0$ touch a Circle C_1 of diameter 6. If the centre of C_1 lies in the first quadrant, find the equation of circle C_2 which is concentric with C_1 and cuts intercepts of length 8 on these lines.

Solution

Given lines are tangents, their equations

$$\mathbf{n_1}\mathbf{x} = c_1, \mathbf{n_2}\mathbf{x} = c_2 \tag{1}$$

$\mathbf{c_1}$	10	Constant for line 1
$\mathbf{c_2}$	40	Constant for line 2
$\mathbf{n_1}$	$\begin{pmatrix} 5 \\ 12 \end{pmatrix}$	normal of line 1
$\mathbf{n_2}$	$\begin{pmatrix} 5 \\ -12 \end{pmatrix}$	normal of line 2

(2)

Distance of point from a line

Solution

$$d = \frac{\left|\mathbf{n}^T\mathbf{x} - c\right|}{\left\|\mathbf{n}\right\|} \tag{3}$$

Center must lie on one of the angle bisector of tangents

d_1	Distance of center from tangent 1
d_2	Distance of center from tangent 2
$\mathbf{x}$	center of circle C_1

(4)

$$\frac{\left|\mathbf{n_1}^T\mathbf{x} - c_1\right|}{\left\|\mathbf{n_1}\right\|} = \frac{\left|\mathbf{n_2}^T\mathbf{x} - c_2\right|}{\left\|\mathbf{n_2}\right\|} \tag{5}$$

$$\frac{\left|\mathbf{n_1}^T\mathbf{x} - 10\right|}{13} = \frac{\left|\mathbf{n_2}^T\mathbf{x} - 40\right|}{13} \tag{6}$$

(7)

Solution

$$\mathbf{n_1}^T\mathbf{x} - 10 = \pm(\mathbf{n_2}^T\mathbf{x} - 40) \tag{8}$$

$$\mathbf{n_1}^T\mathbf{x} - 10 = \mathbf{n_2}^T\mathbf{x} - 40,\ \ \mathbf{n_1}^T\mathbf{x} - 10 = -\mathbf{n_2}^T\mathbf{x} + 40 \tag{9}$$

$$(\mathbf{n_2}^T - \mathbf{n_1}^T)\mathbf{x} = 30, (\mathbf{n_2}^T + \mathbf{n_1}^T)\mathbf{x} = 50 \tag{10}$$

$$\begin{pmatrix} 0 & -24 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 30 \tag{11}$$

$$-24y = 30 \implies y = \frac{-5}{6} \tag{12}$$

Solution

$$\begin{pmatrix} 10 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 50 \tag{13}$$

$$10x = 50 \implies x = 5 \tag{14}$$

Since center is in I quadrant so

$$Case : y = \frac{-5}{6}, \text{rejected} \tag{15}$$

$$Case : x = 5, \text{accepted} \tag{16}$$

Solution

Now

$$\frac{\left|\mathbf{n_1}^T\mathbf{x} - c_1\right|}{\left\|\mathbf{n_1}\right\|} = 3 \tag{17}$$

$$\mathbf{n_1}^T\mathbf{x} - c_1 = \pm 39 \tag{18}$$

$$5x + 12y - 10 = \pm 39 \tag{19}$$

$$at, x = 5 \tag{20}$$

$$y = 2, -\frac{54}{12} \tag{21}$$

$$so, center = \mathbf{c} = \begin{pmatrix} 5 \\ 2 \end{pmatrix} \tag{22}$$

Solution

General equation of conic

$$g(\mathbf{x}) = \mathbf{x}^{\mathbf{T}}\mathbf{V}\mathbf{x} + 2\mathbf{u}^{\mathbf{T}}\mathbf{x} + f \tag{23}$$

Intercept by a circle on line

$$r^2 = p^2 + d^2 \tag{24}$$

d	Distance of center from line
p	intercept by circle on line

(25)

$$d = 3, p = \frac{8}{2} = 4 \tag{26}$$

Solution

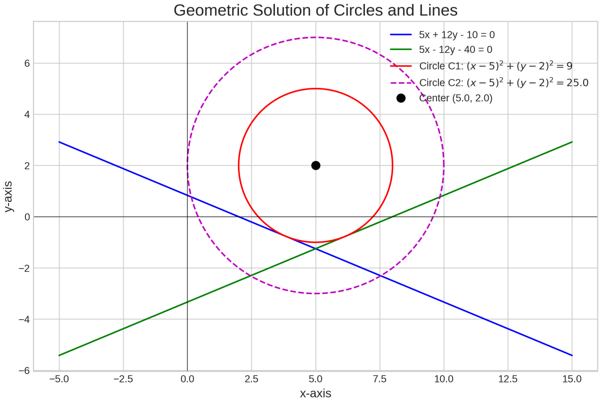
So,

$$r^2 = 4^2 + 3^2 \tag{27}$$
$$r = 5 \tag{28}$$

Equation of circle C_2,

$$\mathbf{x}^{\mathbf{T}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{x} + 2 \begin{pmatrix} -5 \\ -2 \end{pmatrix}^T \mathbf{x} - 5^2 = 0 \tag{29}$$

Figure



Geometric Solution of Circles and Lines
5x + 12y - 10 = 0
5x - 12y - 40 = 0
Circle C1: $(x-5)^2+(y-2)^2=9$
Circle C2: $(x-5)^2+(y-2)^2=25.0$
Center (5.0, 2.0)
y-axis
x-axis
6
4
2
0
−2
−4
−6
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0

Direct Python

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

line1_coeffs = np.array([5, 12, -10])
line2_coeffs = np.array([5, -12, -40])


A = np.array([[line1_coeffs[0], line1_coeffs[1]],
              [line2_coeffs[0], line2_coeffs[1]]])
b = np.array([-line1_coeffs[2], -line2_coeffs[2]])


intersection_point = np.linalg.solve(A, b) # Returns [5., -1.25]
h = intersection_point[0]
r1 = 3
k = 2
center = np.array([h, k])
```

Direct Python

```
intercept_length = 8
half_intercept = intercept_length / 2.0


r2 = np.hypot(r1, half_intercept) # np.hypot(3, 4) = 5


print("--- Solution (calculated with NumPy) ---")
print(f"Intersection of lines: {intersection_point}")
print(f"Center of circles C1 and C2: ({center[0]}, {center[1]})")
```

```python
print(f"Radius of circle C1: {r1}")
print(f"Radius of circle C2: {r2}")
print("\nThe equation of circle C2 is:")
print(f"(x - {center[0]})**2 + (y - {center[1]})**2 = {r2**2}")
print("-----------------------------------------")
```

Direct Python

```python

def get_circle_points(center_h, center_k, radius):
    """Generates x and y coordinates for a circle."""
    theta = np.linspace(0, 2 * np.pi, 200)
    x = center_h + radius * np.cos(theta)
    y = center_k + radius * np.sin(theta)
    return x, y


x_c1, y_c1 = get_circle_points(center[0], center[1], r1)
x_c2, y_c2 = get_circle_points(center[0], center[1], r2)


x_vals = np.linspace(-5, 15, 400)
```

Direct Python

```
y_line1 = (-line1_coeffs[0] * x_vals - line1_coeffs[2]) /
    line1_coeffs[1]

y_line2 = (-line2_coeffs[0] * x_vals - line2_coeffs[2]) /
    line2_coeffs[1]


plt.style.use('seaborn-v0_8-whitegrid')
fig, ax = plt.subplots(figsize=(10, 10))


ax.plot(x_vals, y_line1, 'b-', label='5x + 12y - 10 = 0')
ax.plot(x_vals, y_line2, 'g-', label='5x - 12y - 40 = 0')
```

Direct Python

```python

ax.plot(x_c1, y_c1, 'r-', label=f'Circle C1: $(x-5)^2 + (y-2)^2 = {r1**2}$')
ax.plot(x_c2, y_c2, 'm--', label=f'Circle C2: $(x-5)^2 + (y-2)^2 = {r2**2}$')

ax.plot(center[0], center[1], 'ko', markersize=8, label=f'Center ({center[0]}, {center[1]})')


ax.set_title('Geometric Solution of Circles and Lines', fontsize=16)
ax.set_xlabel('x-axis', fontsize=12)
ax.set_ylabel('y-axis', fontsize=12)
ax.legend(fontsize=10)
```

Direct Python

```
ax.axhline(0, color='black', linewidth=0.5)
ax.axvline(0, color='black', linewidth=0.5)

ax.set_aspect('equal', adjustable='box')
plt.savefig("figure.png", dpi=300)
plt.show()
```

C code

```c
#include <stdio.h>
#include <math.h>

struct Circle {
    double h, k, r;
};
struct Circle find_C1() {
    struct Circle c1;
    c1.h = 5.0;
    c1.k = 2.0;
    c1.r = 3.0;
    return c1;
}
struct Circle find_C2(struct Circle c1, double L) {
    struct Circle c2;
```

C code

```c
    c2.h = c1.h;
    c2.k = c1.k;

    double d = 3.0;
    c2.r = sqrt(pow(L/2, 2) + pow(d, 2));
    return c2;
}
void get_circles(double *h1, double *k1, double *r1, double *h2,
     double *k2, double *r2) {
    struct Circle c1 = find_C1();
    struct Circle c2 = find_C2(c1, 8.0);

    *h1 = c1.h; *k1 = c1.k; *r1 = c1.r;
    *h2 = c2.h; *k2 = c2.k; *r2 = c2.r;
}
```

Python code with shared object

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


lib = ctypes.CDLL('./main.so')


lib.get_circles.argtypes = [
    ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.
        c_double), ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.
        c_double), ctypes.POINTER(ctypes.c_double)
]
```

Python code with shared object

```
h1 = ctypes.c_double()
k1 = ctypes.c_double()
r1 = ctypes.c_double()
h2 = ctypes.c_double()
k2 = ctypes.c_double()
r2 = ctypes.c_double()

# Call C function
lib.get_circles(ctypes.byref(h1), ctypes.byref(k1), ctypes.byref(
    k2), ctypes.byref(h2), ctypes.byref(k2), ctypes.byref(r2))
```

Python code with shared object

```
# But fix: k2 got used twice  fix proper call
lib.get_circles(ctypes.byref(h1), ctypes.byref(k1), ctypes.byref(
    r1), ctypes.byref(h2), ctypes.byref(k2), ctypes.byref(r2))

# Extract values
h1, k1, r1 = h1.value, k1.value, r1.value
h2, k2, r2 = h2.value, k2.value, r2.value
```

Python code with shared object

```
# Plot circles and lines
theta = np.linspace(0, 2*np.pi, 400)
x1, y1 = h1 + r1*np.cos(theta), k1 + r1*np.sin(theta)
x2, y2 = h2 + r2*np.cos(theta), k2 + r2*np.sin(theta)

plt.figure(figsize=(7,7))
plt.plot(x1, y1, 'b', label='Circle C1')
plt.plot(x2, y2, 'g', label='Circle C2')
```

Python code with shared object

```
# Plot lines
x = np.linspace(-5, 15, 400)
y_line1 = (-5*x + 10)/12
y_line2 = (5*x - 40)/12
plt.plot(x, y_line1, 'r--', label='Line 1')
plt.plot(x, y_line2, 'r-.', label='Line 2')
```

Python code with shared object

```
# Format plot
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.legend()
plt.title('Circles C1 & C2 with Tangent Lines')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```