

10.7.76

INDHIRESH S - EE25BTECH11027

7 October, 2025

# Question

The number of common tangents to the circles  $x^2 + y^2 = 4$  and  $x^2 + y^2 - 6x - 8y = 24$  is

- ① 0
- ② 1
- ③ 2
- ④ 3

# Equation I

Let the equation of 1st circle be:

$$\|\mathbf{x}\|^2 + 2\mathbf{u}_1^T \mathbf{x} + f_1 = 0 \quad (1)$$

Let the equation of 2nd circle be

$$\|\mathbf{x}\|^2 + 2\mathbf{u}_2^T \mathbf{x} + f_2 = 0 \quad (2)$$

# Theoretical Solution

Let  $\mathbf{c}_1$  and  $\mathbf{c}_2$  be the center of circles 1 and 2.

From the given information:

$$\mathbf{u}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, f_1 = -4 \text{ and } \mathbf{c}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3)$$

$$\mathbf{u}_2 = \begin{pmatrix} -3 \\ -4 \end{pmatrix}, f_2 = -24 \text{ and } \mathbf{c}_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \quad (4)$$

$$r_1 = \sqrt{\|\mathbf{u}_1\|^2 - f_1} = \sqrt{0 + 4} = 2 \quad (5)$$

$$r_2 = \sqrt{\|\mathbf{u}_2\|^2 - f_2} = \sqrt{25 + 24} = 7 \quad (6)$$

# Theoretical solution

Now calculating the distance between  $\mathbf{c}_1$  and  $\mathbf{c}_2$  :

$$\|\mathbf{c}_2 - \mathbf{c}_1\| = \left\| \begin{pmatrix} 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\| = 5 \quad (7)$$

We also have:

$$|r_2 - r_1| = 5 \quad (8)$$

Here we can observe that the distance between the centers of two circles is equal to the absolute difference of their radii . So the two circles touch each other internally at one point

Therefore the number of common tangents to the given two circles is 1

# C Code

```
#include <math.h>

int find_common_tangents(double x1, double y1, double r1, double
x2, double y2, double r2) {
    // Calculate the distance between the centers
    double d = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

    // Calculate the sum and difference of the radii
    double r_sum = r1 + r2;
    double r_diff = fabs(r1 - r2);

    // Determine the relationship between the circles
    if (d > r_sum) {
        return 4; // Circles are separate and do not intersect
    } else if (d == r_sum) {
        return 3; // Circles touch externally
    } else if (d > r_diff && d < r_sum) {
        return 2; // Circles intersect at two points
    }
}
```

```
else if (d == r_diff) {  
    return 1; // Circles touch internally  
}  
else if (d < r_diff) {  
    return 0; // One circle is completely inside the other  
} else if (d == 0 && r1 == r2) {  
    return -1; // Concentric and identical  
}  
  
return 0; // Default case, including d=0 and r1!=r2  
}
```

# Python Code

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

# --- 1. Load the C shared library ---
lib_name = 'circle.so'
if platform.system() == 'Windows':
    lib_name = 'circle.dll'

try:
    c_lib = ctypes.CDLL(f'./{lib_name}')
except OSError as e:
    print(fError loading shared library: {e})
    print(fPlease compile circle.c into {lib_name} first.)
    exit()
```



```
# --- 2. Define the C function signature for Python ---
c_lib.find_common_tangents.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double, ctypes.c_double
]
c_lib.find_common_tangents.restype = ctypes.c_int

def solve_with_c(c1, r1, c2, r2):
    A Python wrapper that calls the C function.
    return c_lib.find_common_tangents(c1[0], c1[1], r1, c2[0], c2
        [1], r2)

def plot_circles(c1, r1, c2, r2, tangency_point):
    Plots the two circles, their centers, and the point of
    tangency.
    fig, ax = plt.subplots(figsize=(10, 8))
```

```
# Create circle patches
circle1 = plt.Circle(c1, r1, color='blue', fill=False,
    linewidth=2, label=f'Circle 1: $r_1={r1}$')
circle2 = plt.Circle(c2, r2, color='red', fill=False,
    linewidth=2, label=f'Circle 2: $r_2={r2}$')

ax.add_patch(circle1)
ax.add_patch(circle2)

# Plot centers and label them with coordinates
ax.plot(c1[0], c1[1], 'bo', markersize=8, label='Center $C_1$')
ax.text(c1[0] + 0.3, c1[1] + 0.3, f'$C_1$ ({c1[0]:.1f}, {c1[1]:.1f})',
    fontsize=12, color='blue')
```

```
ax.plot(c2[0], c2[1], 'ro', markersize=8, label='Center  
$C_2$')  
ax.text(c2[0] + 0.3, c2[1] + 0.3, f'$C_2$ ({c2[0]:.1f}, {c2  
[1]:.1f})', fontsize=12, color='red')  
  
# Plot tangency point and label it with coordinates  
ax.plot(tangency_point[0], tangency_point[1], 'go',  
        markersize=8, label='Tangency Point T')  
ax.text(tangency_point[0] + 0.3, tangency_point[1] - 0.5, f'T  
({tangency_point[0]}, {tangency_point[1]})', fontsize  
=12, color='green')  
  
# Set plot properties  
ax.set_aspect('equal', adjustable='box')  
plt.title('Relationship Between Two Circles (Internal  
Tangency)')
```

```
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.grid(True)
plt.legend(loc='upper right')
plt.savefig(/media/indhiresh-s/New Volume/Matrix/ee1030-2025/
            ee25btech11027/MATGEO/10.7.76/figs/figure1.png)
plt.show()

# --- Main execution ---
if __name__ == '__main__':
    # Parameters for the circles from the problem
    C1 = (0.0, 0.0)
    r1 = 2.0
    C2 = (3.0, 4.0)
    r2 = 7.0
```

```
# Calculate the number of tangents using the C function
num_tangents = solve_with_c(C1, r1, C2, r2)
print(fNumber of common tangents (via C function): {
      num_tangents})

# Calculate the point of tangency for plotting
v = np.array(C1) - np.array(C2)
u = v / np.linalg.norm(v)
T = tuple(np.array(C2) + r2 * u)
T_rounded = (round(T[0], 2), round(T[1], 2))

# Plot the result
plot_circles(C1, r1, C2, r2, T_rounded)
```

