10.5.8

Naman Kumar-EE25BTECH11041

5 Oct,2025

Question)

Draw two concentric circles of radii 3 cm and 5 cm. Taking a point on outer circle construct the pair of tangents to the other. Measure the length of a tangent and verify it by actual calculation.

Solution

General equation of conic

$$g(\mathbf{x}) = \mathbf{x}^{\mathbf{T}}\mathbf{V}\mathbf{x} + 2\mathbf{u}^{\mathbf{T}}\mathbf{x} + f \tag{1}$$

Equation of circle,

$$\mathbf{x}^{\mathbf{T}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{x} + 2 \begin{pmatrix} 0 \\ 0 \end{pmatrix}^T \mathbf{x} - r^2 = 0, r = \text{radius od circle} \tag{2}$$

$$r_1 = 3cm, r_2 = 5cm \tag{3}$$

Solution

A point lies on the tangent to the conic if it satisfies the following equation

$$\mathbf{m}^T \left[\left(\mathbf{V}\mathbf{h} + \mathbf{u} \right) \left(\mathbf{V}\mathbf{h} + \mathbf{u} \right)^T - \mathbf{V} g(\mathbf{h}) \right] \mathbf{m} = 0 \tag{4}$$

Assuming a point on outer circle as $\mathbf{A}(5, 0)$
putting $\mathbf{A}$ in (2) for inner circle

$$\mathbf{A^T} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{A} + 2 \begin{pmatrix} 0 \\ 0 \end{pmatrix}^T \mathbf{A} - (r_1)^2 \tag{5}$$

$$25 - 9 = 16 \tag{6}$$

$$g(\mathbf{A})_1 = 16 \tag{7}$$

Solution

Calculating $(\mathbf{VA} + \mathbf{u})$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 5 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{8}$$

$$\begin{pmatrix} 5 \\ 0 \end{pmatrix} \tag{9}$$

Solution

putting in (4)

$$\mathbf{m}^T \left[\left(\mathbf{VA} + \mathbf{u}\right)\left(\mathbf{VA} + \mathbf{u}\right)^T - \mathbf{V}g(\mathbf{A})_1 \right] \mathbf{m} = 0 \tag{10}$$

$$\mathbf{m}^T \left[\begin{pmatrix} 5 \\ 0 \end{pmatrix} \begin{pmatrix} 5 \\ 0 \end{pmatrix}^T - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \times 16 \right] \mathbf{m} = 0 \tag{11}$$

$$\mathbf{m}^T \left[\begin{pmatrix} 9 & 0 \\ 0 & -16 \end{pmatrix} \right] \mathbf{m} = 0 \tag{12}$$

Solution

$$\begin{pmatrix} 1 \\ m \end{pmatrix}^T \begin{pmatrix} 9 & 0 \\ 0 & -16 \end{pmatrix} \begin{pmatrix} 1 \\ m \end{pmatrix} = 0 \tag{13}$$

$$9 - 16m^2 = 0 \tag{14}$$

$$m = \pm\frac{3}{4} \tag{15}$$

$$\mathbf{m} = \begin{pmatrix} 1 \\ \pm\frac{3}{4} \end{pmatrix} \tag{16}$$

Solution

Using following formula to find point of contact of tangent

$$\mathbf{q}_j = \left(\pm r \frac{\mathbf{n}_j}{\|\mathbf{n}_j\|} - \mathbf{u} \right), j = 1, 2 \tag{17}$$

$$\mathbf{q_1} = \left(\pm 3 \frac{\begin{pmatrix} \frac{3}{4} \\ 1 \end{pmatrix}}{\sqrt{\left(\frac{3}{4}\right)^2 + 1}} \right) \tag{18}$$

$$\mathbf{q}_1 = \pm \begin{pmatrix} \frac{9}{5} \\ \frac{12}{5} \end{pmatrix} \tag{19}$$

$$Similarly, \mathbf{q}_2 = \pm \begin{pmatrix} \frac{9}{5} \\ \frac{-12}{5} \end{pmatrix} \tag{20}$$

Solution

To take the ones passing through $\mathbf{A}$ taking $\mathbf{q}_1$ and $\mathbf{q}_2$ as

$$\mathbf{q}_1 = \begin{pmatrix} \frac{9}{5} \\ \frac{12}{5} \end{pmatrix} \tag{21}$$

$$\mathbf{q}_2 = \begin{pmatrix} \frac{9}{5} \\ \frac{-12}{5} \end{pmatrix} \tag{22}$$

Length of both tangent will be equal and will be

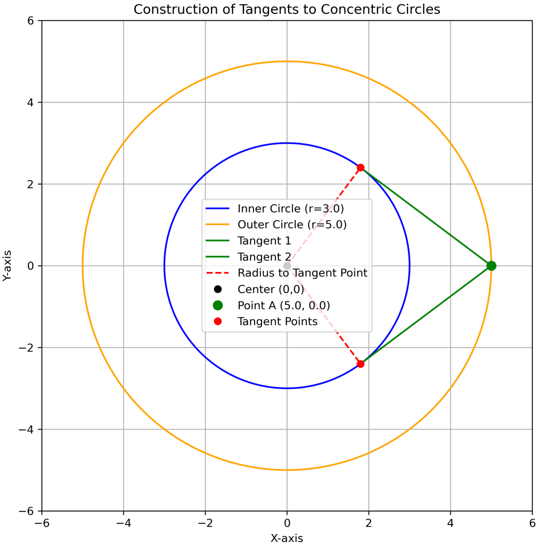
$$\|\mathbf{q_1} - \mathbf{A}\| \tag{23}$$

$$\| \begin{pmatrix} \frac{9}{5} \\ \frac{12}{5} \end{pmatrix} - \begin{pmatrix} 5 \\ 0 \end{pmatrix} \| \tag{24}$$

$$\| \begin{pmatrix} \frac{-16}{5} \\ \frac{12}{5} \end{pmatrix} \| \tag{25}$$

$$= 4 \tag{26}$$

Figure



Construction of Tangents to Concentric Circles
Inner Circle (r=3.0)
Outer Circle (r=5.0)
Tangent 1
Tangent 2
Radius to Tangent Point
Center (0,0)
Point A (5.0, 0.0)
Tangent Points
Y-axis
X-axis
6
4
2
0
−2
−4
−6

Direct Python

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


center = (0, 0)
r_inner = 3.0
r_outer = 5.0


point_A = np.array([5.0, 0.0])


tangent_point_1 = np.array([9/5, 12/5]) # (1.8, 2.4)
tangent_point_2 = np.array([9/5, -12/5]) # (1.8, -2.4)

theta = np.linspace(0, 2 * np.pi, 200)
```

Direct Python

```
x_inner = center[0] + r_inner * np.cos(theta)
y_inner = center[1] + r_inner * np.sin(theta)


x_outer = center[0] + r_outer * np.cos(theta)
y_outer = center[1] + r_outer * np.sin(theta)


plt.figure(figsize=(8, 8))
ax = plt.gca()


ax.plot(x_inner, y_inner, label=f'Inner Circle (r={r_inner})',
    color='blue')
ax.plot(x_outer, y_outer, label=f'Outer Circle (r={r_outer})',
    color='orange')
```

Direct Python

```python

ax.plot([point_A[0], tangent_point_1[0]], [point_A[1], tangent_point_1[1]], 'g-', label='Tangent 1')
ax.plot([point_A[0], tangent_point_2[0]], [point_A[1], tangent_point_2[1]], 'g-', label='Tangent 2')

ax.plot([center[0], tangent_point_1[0]], [center[1], tangent_point_1[1]], 'r--', label='Radius to Tangent Point')
ax.plot([center[0], tangent_point_2[0]], [center[1], tangent_point_2[1]], 'r--')


ax.plot(center[0], center[1], 'ko', label='Center (0,0)')
ax.plot(point_A[0], point_A[1], 'go', markersize=8, label=f'Point A {tuple(point_A)}')
ax.plot(tangent_point_1[0], tangent_point_1[1], 'ro', label='Tangent Points')
ax.plot(tangent_point_2[0], tangent_point_2[1], 'ro')
```

Direct Python

```
# --- 5. Final Plot Adjustments ---
# Ensure the plot has an equal aspect ratio to make circles look
    circular
ax.set_aspect('equal', adjustable='box')

# Add titles and labels for clarity
plt.title('Construction of Tangents to Concentric Circles')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()
plt.grid(True)

# Set axis limits for a nice view
plt.xlim(-6, 6)
plt.ylim(-6, 6)

plt.savefig("figure.png", dpi=300)
# Show the plot
plt.show()
```

C code

```c
// main.c
#include <stdio.h>
#include <math.h>
double tangent_length(double R, double r) {
    if (R <= r) {
        printf("Invalid input: Outer radius must be greater than
            inner radius.\n");
        return -1;
    }
    return sqrt((R * R) - (r * r));
}
```

C code

```
    return k;
}

int main() {
    double k = find_k();
    printf("The value of k = %.2lf\n", k);
    return 0;
}
```

Python code with shared object

```python
# main.py
import ctypes
import numpy as np
import matplotlib.pyplot as plt
from math import acos, sqrt

# Load the shared object file (compiled C code)
lib = ctypes.CDLL("./main.so")

# Define argument and return types for the C function
lib.tangent_length.argtypes = [ctypes.c_double, ctypes.c_double]
lib.tangent_length.restype = ctypes.c_double

# Given radii
r = 3.0
R = 5.0
```

Python code with shared object

```python
# Call C function to get tangent length
tangent_len_c = lib.tangent_length(R, r)

if tangent_len_c < 0:
    raise ValueError("Invalid radii: R must be greater than r")

print(f"[C Function] Tangent length ((Rr)) = {tangent_len_c:.3f}
    cm")

# Now use geometry to verify and plot
theta = np.deg2rad(60)
O = np.array([0.0, 0.0])
P = np.array([R * np.cos(theta), R * np.sin(theta)])
```

Python code with shared object

```
# Compute tangent points
beta = acos(r / R)
T1 = np.array([r * np.cos(theta + beta), r * np.sin(theta + beta)
    ])
T2 = np.array([r * np.cos(theta - beta), r * np.sin(theta - beta)
    ])

# Verify length geometrically
L1 = np.linalg.norm(P - T1)
L2 = np.linalg.norm(P - T2)

print(f"[Python Geometry] PT1 = {L1:.3f} cm, PT2 = {L2:.3f} cm")

# Plot
fig, ax = plt.subplots(figsize=(6,6))
ax.set_aspect('equal', 'box')
ax.set_title("Tangents from a Point on Outer Circle to Inner
    Circle")
```

Python code with shared object

```
# Circles
outer = plt.Circle((0,0), R, fill=False, color='blue', linestyle='--', label='Outer Circle (R=5 cm)')
inner = plt.Circle((0,0), r, fill=False, color='green', linestyle='--', label='Inner Circle (r=3 cm)')
ax.add_patch(outer)
ax.add_patch(inner)

# Lines
ax.plot([0, P[0]], [0, P[1]], 'k-', label='OP')
ax.plot([P[0], T1[0]], [P[1], T1[1]], 'r-', label='Tangent 1')
ax.plot([P[0], T2[0]], [P[1], T2[1]], 'r-')
ax.plot([0, T1[0]], [0, T1[1]], 'gray', linestyle=':')
ax.plot([0, T2[0]], [0, T2[1]], 'gray', linestyle=':')
```

Python code with shared object

```
# Points
ax.plot(0, 0, 'ko')
ax.plot(P[0], P[1], 'ro')
ax.plot(T1[0], T1[1], 'go')
ax.plot(T2[0], T2[1], 'go')

ax.text(0.2, -0.3, 'O', fontsize=10)
ax.text(P[0]*1.05, P[1]*1.05, 'P', fontsize=10)
ax.text(T1[0]*1.05, T1[1]*1.05, 'T1', fontsize=10)
ax.text(T2[0]*1.05, T2[1]*1.05, 'T2', fontsize=10)

ax.legend()
ax.grid(True)
plt.show()
```