10.7.75

Harsha-EE25BTECH11026

September 12,2025

Question

Find the equations of tangents drawn from origin to the circle $x^2 + y^2 - 2rx - 2hy + h^2 = 0$,are

1. $x = 0$
2. $y = 0$
3. $\left(h^2 - r^2\right) x - 2rhy = 0$
4. $\left(h^2 - r^2\right) x + 2rhy = 0$

Theoretical Solution

Given the equation of circle,

$$\mathbf{x}^\top \mathbf{V}\mathbf{x} + 2\mathbf{u}^\top \mathbf{x} + f = 0 \tag{1}$$

where, $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \mathbf{V} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} -r \\ -h \end{pmatrix}$ and $f = h^2$.

It is given that the tangents pass through the origin.

$$\therefore \mathbf{n}^\top \mathbf{x} = 0 \tag{2}$$

where $\mathbf{n}$ is the direction vector of the tangent.

Theoretical Solution

It is known that for any conic , the condition of tangency is given by,

$$\mathbf{n}^\top \mathbf{\Sigma} \mathbf{n} = 0 \tag{3}$$

where,

$$\mathbf{n} = \begin{pmatrix} 1 \\ m \end{pmatrix} \text{(Direction vector of tangent)} \tag{4}$$

$$\mathbf{\Sigma} = (\mathbf{V}\mathbf{h} + \mathbf{u})(\mathbf{V}\mathbf{h} + \mathbf{u})^\top - S(\mathbf{h})\mathbf{V} \tag{5}$$

$\mathbf{h}$ is the point through which the tangent passes and
$S(\mathbf{h}) = \mathbf{h}^\top \mathbf{V}\mathbf{h} + 2\mathbf{u}^\top \mathbf{h} + f = 0.$

Theoretical Solution

From (2), (5) reduces to,

$$\boldsymbol{\Sigma} = \mathbf{u}\mathbf{u}^\top - f\mathbf{V} \tag{6}$$

yielding,

$$\mathbf{n}^\top \left(\mathbf{u}\mathbf{u}^\top - f\mathbf{V}\right)\mathbf{n} = 0 \tag{7}$$

$$\implies \mathbf{n}^\top\mathbf{u}\mathbf{u}^\top\mathbf{n} - f\mathbf{n}^\top\mathbf{V}\mathbf{n} = 0 \tag{8}$$

$$\therefore \left\|\mathbf{u}^\top\mathbf{n}\right\|^2 = f\mathbf{n}^\top\mathbf{V}\mathbf{n} \tag{9}$$

Theoretical Solution

Substituting $\mathbf{V}$ in (9),

$$\implies \|\mathbf{u}^\top \mathbf{n}\|^2 = f\|\mathbf{n}\|^2 \tag{10}$$

$$\implies (rm+h)^2 = h^2\left(1+m^2\right) \tag{11}$$

$$\therefore m\left(\left(r^2-h^2\right)m - 2rh\right) = 0 \tag{12}$$

$$\implies \mathbf{n} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \mathbf{n} = \begin{pmatrix} h^2-r^2 \\ -2rh \end{pmatrix} \tag{13}$$

C Code -Finding the equation of tangents

```c
#include <stdio.h>

void solve_tangents(double r, double h, double tangents[4]) {
    // First tangent: x = 0 -> line (1,0)
    tangents[0] = 1.0;
    tangents[1] = 0.0;
    // Second tangent: (h^2 - r^2)x - 2rh y = 0
    double a = h*h - r*r;
    double b = -2.0*r*h;

    if (a == 0 && b == 0) {
        // Special case -> y=0
        tangents[2] = 0.0;
        tangents[3] = 1.0;
    } else {
        tangents[2] = a;
        tangents[3] = b;
    }
}
```

Python+C code

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

# Load shared C library
lib = ctypes.CDLL("./libtangent_solver.so")

# Define C function signature
lib.solve_tangents.argtypes = [ctypes.c_double, ctypes.c_double,
                               ctypes.POINTER(ctypes.c_double)]
lib.solve_tangents.restype = None

def get_tangents_from_c(r, h):
    results = (ctypes.c_double * 4)()
    lib.solve_tangents(r, h, results)
    vals = list(results)
    tangents = [(vals[0], vals[1]), (vals[2], vals[3])]
```

Python+C code

```
 tangent_eqs = []
    for a, b in tangents:
        if b == 0: # x=0
            tangent_eqs.append("x = 0")
        elif a == 0: # y=0
            tangent_eqs.append("y = 0")
        else:
            tangent_eqs.append(f"{a:.0f}x {b:+.0f}y = 0")

    return tangents, tangent_eqs

def plot_tangents(r, h):
    tangents, tangent_eqs = get_tangents_from_c(r, h)

    print("Tangents from origin:")
    for eq in tangent_eqs:
        print(" ", eq)
```

Python+C code

```
# Circle
    theta = np.linspace(0, 2*np.pi, 400)
    x_circ = r + r*np.cos(theta)
    y_circ = h + r*np.sin(theta)

    plt.figure(figsize=(6,6))
    plt.plot(x_circ, y_circ, 'b', label="Circle")

    # Tangents
    x_vals = np.linspace(-2*r, 2*r, 400)
    for (a,b), eq in zip(tangents, tangent_eqs):
        if b == 0:
            plt.axvline(0, color='r', linestyle='--', label=eq)
        else:
            y_vals = -(a/b)*x_vals
            plt.plot(x_vals, y_vals, 'r--', label=eq)
```

Python+C code

```
# Origin & center
    plt.scatter([0],[0], color='k', marker='o', label="Origin
        (0,0)")
    plt.scatter([r],[h], color='g', marker='x', label=f"Center ({
        r},{h})")

    plt.gca().set_aspect('equal', adjustable='box')
    plt.axhline(0, color='gray', linewidth=0.5)
    plt.axvline(0, color='gray', linewidth=0.5)
    plt.legend()
    plt.title(f"Tangents from Origin to Circle (r={r}, h={h})")
    plt.savefig("/home/user/Matrix Theory: workspace/
        Matgeo_assignments/10.7.75/figs/figure_1.png")
    plt.show()

# Example
plot_tangents(r=3, h=4)
```

Python code

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

def tangents_from_origin(r, h):
    """
    Finds tangent lines from origin to the circle
    x^2 + y^2 - 2rx - 2hy + h^2 = 0.

    Returns:
        tangents: list of (a,b) representing ax+by=0
        tangent_eqs: list of pretty string equations
    """
    tangents = []
    tangent_eqs = []
    # Case 1: x=0
    tangents.append((1, 0))
    tangent_eqs.append("x = 0")
```

Python code

```
# Case 2
    if (h**2 - r**2) != 0:
        a, b = (h**2 - r**2, -2*r*h)
        tangents.append((a, b))
        tangent_eqs.append(f"{a}x {b:+}y = 0")
    else:
        tangents.append((0, 1))
        tangent_eqs.append("y = 0")

    return tangents, tangent_eqs

def plot_tangents(r, h):
    tangents, tangent_eqs = tangents_from_origin(r, h)

    # Print immediately
    print("Tangents from origin:")
    for eq in tangent_eqs:
        print(" ", eq)
```

Python code

```
# Circle: center = (r,h), radius = r
    theta = np.linspace(0, 2*np.pi, 400)
    x_circ = r + r*np.cos(theta)
    y_circ = h + r*np.sin(theta)

    plt.figure(figsize=(6,6))
    plt.plot(x_circ, y_circ, 'b', label="Circle")

    # Tangents
    x_vals = np.linspace(-2*r, 2*r, 400)
    for (a,b), eq in zip(tangents, tangent_eqs):
        if b == 0:
            plt.axvline(0, color='r', linestyle='--', label=eq)
        else:
            y_vals = -(a/b)*x_vals
            plt.plot(x_vals, y_vals, 'r--', label=eq)
```

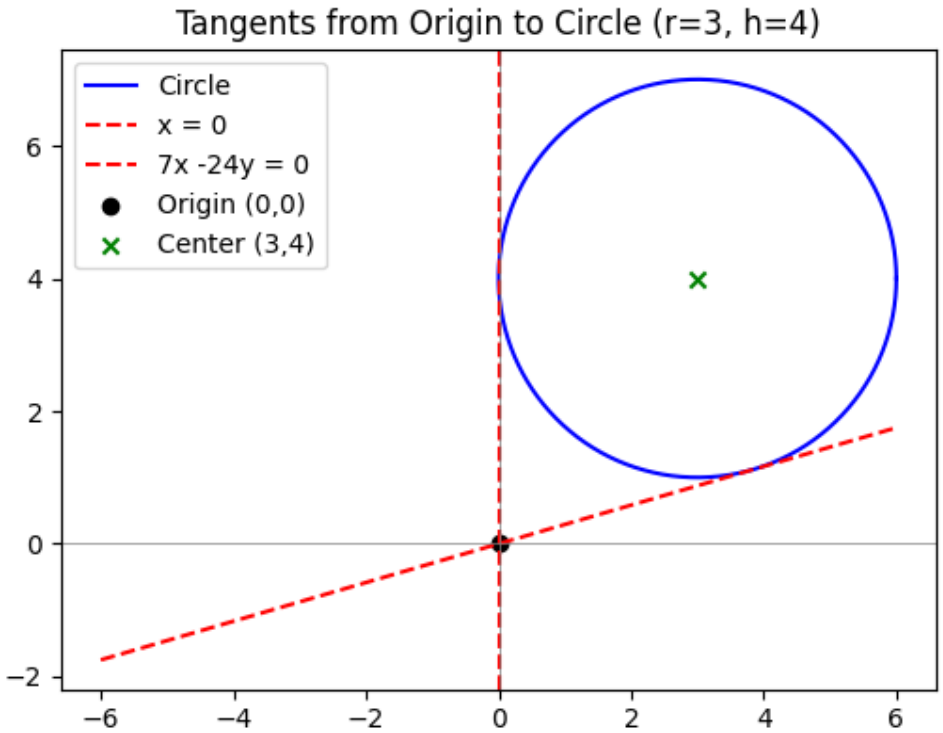
Python code

```
# Mark origin & center
    plt.scatter([0],[0], color='k', marker='o', label="Origin
        (0,0)")
    plt.scatter([r],[h], color='g', marker='x', label=f"Center ({
        r},{h})")

    plt.gca().set_aspect('equal', adjustable='box')
    plt.axhline(0, color='gray', linewidth=0.5)
    plt.axvline(0, color='gray', linewidth=0.5)
    plt.legend()
    plt.title(f"Tangents from Origin to Circle (r={r}, h={h})")
    plt.savefig("/home/user/Matrix Theory: workspace/
        Matgeo_assignments/10.7.75/figs/Figure_1.png")
    plt.show()

# Example usage
plot_tangents(r=3, h=4)
```

Plot


Tangents from Origin to Circle (r=3, h=4)
Circle
x = 0
7x -24y = 0
Origin (0,0)
Center (3,4)
6
4
2
0
−2
−6
−4
−2
0
2
4
6