Problem 10.3.26

ee25btech11023-Venkata Sai

September 30, 2025

1. Problem

2. Solution
   - Equation
   - Simplify
   - Finding the variables
   - Conclusion
   - Plot

3. C Code

4. Python Code

Problem

Find the point at which the tangent to the curve $y = \sqrt{4x-3} - 1$ has its slope $\frac{2}{3}$

Equation

Given curve

$$y = \sqrt{4x-3} - 1 \tag{3.1}$$

$$y + 1 = \sqrt{4x-3} \implies (y+1)^2 = 4x - 3 \tag{3.2}$$

$$y^2 + 2y + 1 = 4x - 3 \tag{3.3}$$

$$y^2 - 4x + 2y + 4 = 0 \tag{3.4}$$

Equation (4) in matrix form

$$y^2 + 2\left(-2x + y\right) + 4 = 0 \tag{3.5}$$

$$\mathbf{x}^\top \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{x} + 2\begin{pmatrix} -2 & 1 \end{pmatrix} \mathbf{x} + 4 = 0 \tag{3.6}$$

The general equation of conic

$$\mathbf{x}^\top \mathbf{V} \mathbf{x} + 2\mathbf{u}^\top \mathbf{x} + f = 0 \tag{3.7}$$

On comparing (6) with (7)

Simplify

$$\mathbf{V} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}, f = 4 \tag{3.8}$$

Given slope

$$m = \frac{2}{3} \tag{3.9}$$

The normal vector to the given tangent is

$$\mathbf{n} = \begin{pmatrix} -m \\ 1 \end{pmatrix} \implies \mathbf{n} = \begin{pmatrix} -\frac{2}{3} \\ 1 \end{pmatrix} \tag{3.10}$$

$$|\mathbf{V}-\lambda\mathbf{I}| = 0 \tag{3.11}$$

$$\left|\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right| = 0 \implies \left|\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right| = 0 \tag{3.12}$$

$$\left|\begin{pmatrix} -\lambda & 0 \\ 0 & 1-\lambda \end{pmatrix}\right| = 0 \implies (-\lambda)(1-\lambda) = 0 \tag{3.13}$$

Finding the variables

Finding eigen vector for $\lambda_1 = 0$

$$\left(\mathbf{V} - \lambda \mathbf{I}\right) \mathbf{p} = \mathbf{0} \tag{3.15}$$

$$\begin{pmatrix} -\lambda & 0 \\ 0 & 1-\lambda \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 0 \implies \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.16}$$

$$0 = 0, y = 0 \implies \mathbf{p_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{3.17}$$

For a given normal vector $\mathbf{n}$, the point of contact $\mathbf{q}$ for a given curve is given by the matrix equation

$$\begin{pmatrix} \left(\mathbf{u} + \kappa \mathbf{n}\right)^\top \\ \mathbf{V} \end{pmatrix} \mathbf{q} = \begin{pmatrix} -f \\ \kappa \mathbf{n} - \mathbf{u} \end{pmatrix} \quad \text{where } \kappa = \frac{\mathbf{p_1}^\top \mathbf{u}}{\mathbf{p_1}^\top \mathbf{n}} \tag{3.18}$$

$$\kappa = \frac{\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} -2 \\ 1 \end{pmatrix}}{\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} -\frac{2}{3} \\ 1 \end{pmatrix}} = \frac{-2}{-\frac{2}{3}} = 3 \tag{3.19}$$

Conclusion

From (8)

$$\begin{pmatrix} \left(\begin{pmatrix} -2 \\ 1 \end{pmatrix} + 3 \begin{pmatrix} -\frac{2}{3} \\ 1 \end{pmatrix} \right)^{\top} \\ \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix} \mathbf{q} = \begin{pmatrix} -4 \\ 3 \begin{pmatrix} -\frac{2}{3} \\ 1 \end{pmatrix} - \begin{pmatrix} -2 \\ 1 \end{pmatrix} \end{pmatrix} \tag{3.20}$$

$$\begin{pmatrix} \begin{pmatrix} -4 \\ 4 \end{pmatrix}^{\top} \\ \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix} \mathbf{q} = \begin{pmatrix} -4 \\ \begin{pmatrix} -2 \\ 3 \end{pmatrix} + \begin{pmatrix} 2 \\ -1 \end{pmatrix} \end{pmatrix} \implies \begin{pmatrix} \begin{pmatrix} -4 & 4 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix} \mathbf{q} = \begin{pmatrix} -4 \\ \begin{pmatrix} 0 \\ 2 \end{pmatrix} \end{pmatrix} \tag{3.21}$$

$$\begin{pmatrix} -4 & 4 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{q} = \begin{pmatrix} -4 \\ 0 \\ 2 \end{pmatrix} \tag{3.22}$$

Conclusion

Taking augmented matrix

$$\left(\begin{array}{cc|c} -4 & 4 & -4 \\ 0 & 0 & 0 \\ 0 & 1 & 2 \end{array}\right) \xrightarrow{R_1 \to R_1 - 4R_2} \left(\begin{array}{cc|c} -4 & 0 & -12 \\ 0 & 0 & 0 \\ 0 & 1 & 2 \end{array}\right) \xrightarrow{R_1 \to -\frac{1}{4}R_1} \left(\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & 1 & 2 \end{array}\right) \tag{3.23}$$

$$\mathbf{q} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \tag{3.24}$$

Hence the point of contact is $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$

Plot

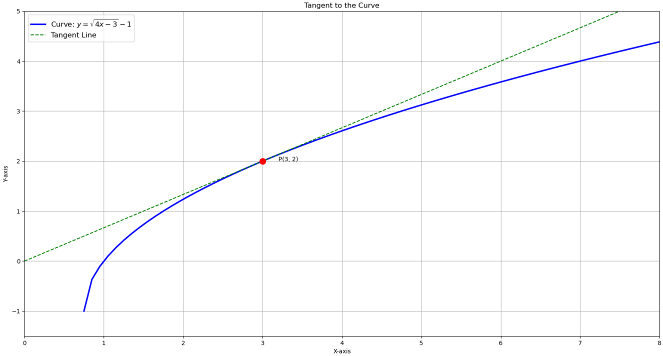
Tangent to the Curve
Curve: $y = \sqrt{4x-3} - 1$
Tangent Line
P(3, 2)
Y-axis
X-axis
5
4
3
2
1
0
−1
0
1
2
3
4
5
6
7
8

Figure:

C Code

```c
#include <math.h>

void get_tangent_data(double* out_data) {
       double tangent_x = 3.0;
    double tangent_y = 2.0;
        int num_points = 101;
    out_data[0] = tangent_x;
    out_data[1] = tangent_y;

    int index = 2;
    for (int i = 0; i < num_points; i++) {
        double x = 0.75 + (10.0 * i) / (num_points - 1);

        out_data[index] = x;
        out_data[index + 1] = sqrt(4 * x - 3) - 1;
        index += 2;
    }
}
```

Python Code for Solving

```
import ctypes
import numpy as np

def get_data_from_c():
    lib = ctypes.CDLL('./code.so')
    data_size = 2 + (101 * 2)
    double_array = ctypes.c_double * data_size
    lib.get_tangent_data.argtypes = [ctypes.POINTER(ctypes.
        c_double)]

    out_data_c = double_array()
    lib.get_tangent_data(out_data_c)

    all_data = np.array(out_data_c)
    tangent_point = all_data[0:2]
    curve_points = all_data[2:].reshape((-1, 2))
    return tangent_point, curve_points
```

Python Code for Plotting

```python
# Code by /sdcard/github/matgeo/codes/CoordGeoVV Sharma
# September 12, 2023
# Revised July 21, 2024
# Released under GNU GPL
# Section Formula
import sys
sys.path.insert(0, '/workspaces/urban-potato/matgeo/codes/CoordGeo/')
import numpy as np
import matplotlib.pyplot as plt

from call import get_data_from_c
P, curve_points = get_data_from_c()
slope = 2.0 / 3.0
x_tangent = np.array([0, 8])
y_tangent = slope * (x_tangent - P[0]) + P[1]
fig, ax = plt.subplots(figsize=(10, 8))
ax.plot(curve_points[:, 0], curve_points[:, 1], 'b-', linewidth=2.5, label='Curve: $y = \\sqrt{4x-3} - 1$')
```

Python Code for Plotting

```python
ax.plot(x_tangent, y_tangent, 'g--', label='Tangent Line')

ax.scatter(P[0], P[1], color='red', s=100, zorder=5)
ax.text(P[0] + 0.2, P[1] + 0.2, f'P({P[0]:.0f}, {P[1]:.0f})')

ax.set_title('Tangent to the Curve')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_xlim(0, 8)
ax.set_ylim(-1.5, 5)

ax.grid(True)
ax.legend(fontsize=12)

plt.show()
plt.savefig('fig.png')
```