10.4.3

INDHIRESH S - EE25BTECH11027

10 October, 2025

Question

The point at which the normal to the curve $y = x + \frac{1}{x}$, $x > 0$ is perpendicular to the line $3x - 4y - 7 = 0$

Equation I

The given curve be rearranged as:

$$x^2 - xy + 1 = 0. \tag{1}$$

This can be expressed in the form:

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

Where:

$$\mathbf{v} = \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{pmatrix} , \mathbf{u} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad and \quad f = 1 \tag{3}$$

Theoretical Solution

The required direction of normal which is perpendicular to the line $3x - 4y - 7 = 0$

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

$$\mathbf{m} = \begin{pmatrix} 3 \\ -4 \end{pmatrix} \tag{5}$$

Now the equation of normal to the conic at the point of contact $\mathbf{q}$ is given by:

$$(\mathbf{V}\mathbf{q} + \mathbf{u})^T \mathbf{R}(\mathbf{x} - \mathbf{q}) = 0 \tag{6}$$

Theoretical solution

In the normal equation $\mathbf{Vq}+\mathbf{u}$ is proportional to the direction vector of the normal.So,

$$\mathbf{Vq}+\mathbf{u}=k\mathbf{m} \tag{7}$$

$$\mathbf{q}=\mathbf{V}^{-1}(k\mathbf{m}-\mathbf{u}) \tag{8}$$

$\mathbf{q}$ lies on the curve.So substituting Eq.8 in Eq.2:

$$(\mathbf{V}^{-1}(k\mathbf{m}-\mathbf{u}))^T\mathbf{V}\mathbf{V}^{-1}(k\mathbf{m}-\mathbf{u})+2\mathbf{u}^T\mathbf{V}^{-1}(k\mathbf{m}-\mathbf{u})+f=0 \tag{9}$$

$$(\mathbf{V}^{-1}(k\mathbf{m}-\mathbf{u}))^T(k\mathbf{m}-\mathbf{u})+f=0 \tag{10}$$

$$\left(\begin{pmatrix}0 & -2\\ -2 & -4\end{pmatrix}\left(k\begin{pmatrix}3\\ -4\end{pmatrix}-\begin{pmatrix}0\\ 0\end{pmatrix}\right)\right)^T\left(k\begin{pmatrix}3\\ -4\end{pmatrix}-\begin{pmatrix}0\\ 0\end{pmatrix}\right)+1=0 \tag{11}$$

Theoretical solution

$$k^2 \begin{pmatrix} 3 & -4 \end{pmatrix} \begin{pmatrix} 0 & -2 \\ -2 & -4 \end{pmatrix} \begin{pmatrix} 3 \\ -4 \end{pmatrix} + 1 = 0 \tag{12}$$

$$k^2 = \frac{1}{16} \tag{13}$$

$$k = \frac{1}{4} \ \ and \ \ k = -\frac{1}{4} \tag{14}$$

Now substitute the corresponding values in the Eq.8 to get the point

$$\mathbf{q} = \begin{pmatrix} 0 & -2 \\ -2 & -4 \end{pmatrix} \left(k \begin{pmatrix} 3 \\ -4 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right) \tag{15}$$

$$\mathbf{q} = k \begin{pmatrix} 0 & -2 \\ -2 & -4 \end{pmatrix} \begin{pmatrix} 3 \\ -4 \end{pmatrix} \tag{16}$$

Theoretical solution

$$\mathbf{q} = k\begin{pmatrix} 8 \\ 10 \end{pmatrix} \tag{17}$$

When $k = \frac{1}{4}$

$$\mathbf{q} = \begin{pmatrix} 2 \\ \frac{5}{2} \end{pmatrix} \tag{18}$$

When $k = -\frac{1}{4}$

$$\mathbf{q} = \begin{pmatrix} -2 \\ -\frac{5}{2} \end{pmatrix} \tag{19}$$

Given that $x > 0$. So the point of contact is

$$\mathbf{q} = \begin{pmatrix} 2 \\ \frac{5}{2} \end{pmatrix} \tag{20}$$

C Code

```c
#include <math.h>
void solve_for_point(double line_A, double line_B, double* contact_x, double* contact_y) {
    // Slope of the given line
    double m_line = -line_A / line_B;

    // Slope of the normal to the curve (which is perpendicular to the line)
    double m_normal_req = -1.0 / m_line;


    double x_squared = m_normal_req / (1.0 + m_normal_req);

    double x = sqrt(x_squared); // Taking positive root since x > 0
    double y = x + (1.0 / x);
```

C Code

```
    // Store the results in the output pointers
    *contact_x = x;
    *contact_y = y;
}
```

Python Code

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

solver_lib = ctypes.CDLL('./normal.so')


# Define the function signature (argument types and return type).
solve_func = solver_lib.solve_for_point
solve_func.argtypes = [
    ctypes.c_double,
    ctypes.c_double,
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)
]
solve_func.restype = None
```

Python Code

```
# Prepare variables for the C function call for the line 3x - 4
    y - 7 = 0.
line_A = ctypes.c_double(3.0)
line_B = ctypes.c_double(-4.0)
contact_x_ptr = ctypes.c_double()
contact_y_ptr = ctypes.c_double()

# Call the C function from Python.
solve_func(line_A, line_B, ctypes.byref(contact_x_ptr), ctypes.
    byref(contact_y_ptr))

# [cite_start]Get the results from the pointers, which
    corresponds to your correct calculation. [cite: 1519, 1520,
    1521, 1522, 1523, 1524, 1525, 1526]
contact_x = contact_x_ptr.value
contact_y = contact_y_ptr.value
```

Python Code

```
  print(--- Python with C Library Solution ---)
print(fThe point of contact is ({contact_x:.1f}, {contact_y:.1f}))

# --- 2. Plotting ---
x_vals = np.linspace(0.1, 5, 400)
curve = x_vals + 1/x_vals
line = (3*x_vals - 7) / 4
# The normal passes through (contact_x, contact_y) and is perpendicular to the line.
# Slope of line is 3/4, so slope of normal is -4/3.
normal_line = (-4/3)*(x_vals - contact_x) + contact_y

plt.figure(figsize=(10, 8))
plt.plot(x_vals, curve, label='Curve: $y = x + 1/x$', color='blue')
plt.plot(x_vals, line, label='Line: $3x - 4y - 7 = 0$', color='red', linestyle='--')
```

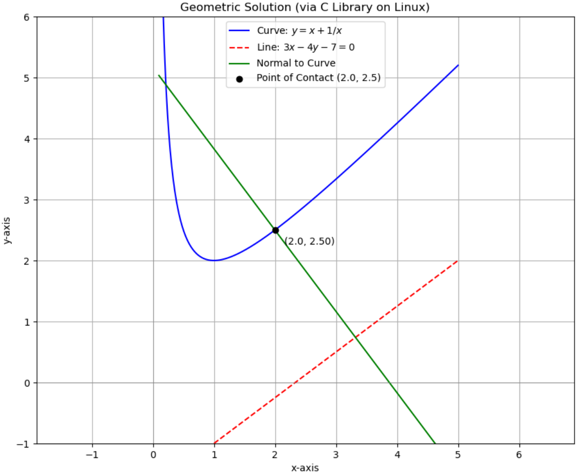
Python Code

```python
    plt.plot(x_vals, normal_line, label='Normal to Curve', color='green')
plt.scatter(contact_x, contact_y, color='black', zorder=5, label=f'Point of Contact ({contact_x:.1f}, {contact_y:.1f})')
plt.annotate(
    f'({contact_x:.1f}, {contact_y:.2f})',
    xy=(contact_x, contact_y),
    xytext=(10, -15),
    textcoords='offset points'
)
plt.title('Geometric Solution (via C Library on Linux)')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.axhline(0, color='grey', linewidth=0.5)
plt.axvline(0, color='grey', linewidth=0.5)
```

Python Code

```
    plt.grid(True)
plt.legend()
plt.axis('equal')
plt.ylim(-1, 6)
plt.xlim(-1, 6)
plt.savefig(/media/indhiresh-s/New Volume/Matrix/ee1030-2025/
    ee25btech11027/MATGEO/10.4.3/figs/figure1.png)
plt.show()
```

Plot



Geometric Solution (via C Library on Linux)
Curve: $y = x + 1/x$
Line: $3x - 4y - 7 = 0$
Normal to Curve
Point of Contact (2.0, 2.5)
(2.0, 2.50)
x-axis
y-axis