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
from matplotlib.lines import Line2D # Needed for custom legend

# --- 1. C Library Integration (No changes here) ---

try:
    solver_lib = ctypes.CDLL('./normal.so')
except OSError:
    print(Could not load 'libsolver.so'. Please compile solver.c
        first.)
    exit()

solve_func = solver_lib.solve_for_point
solve_func.argtypes = [
    ctypes.c_double,
    ctypes.c_double,
```

Python Code

```
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)
]
solve_func.restype = None

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()

solve_func(line_A, line_B, ctypes.byref(contact_x_ptr), ctypes.
    byref(contact_y_ptr))

contact_x = contact_x_ptr.value
contact_y = contact_y_ptr.value

print(--- Python with C Library Solution ---)
```

```
print(f"The point of contact is ({contact_x:.1f}, {contact_y:.2f
    })")


# --- 2. Plotting (Modified Section) ---

# Define a wider range to see the full conic
plot_range = np.linspace(-6, 6, 500)
X, Y = np.meshgrid(plot_range, plot_range)

# Define the implicit equation of the hyperbola: x^2 - xy + 1 = 0
hyperbola_eq = X**2 - X*Y + 1

# The tangent and normal lines, plotted over the new wider range
line = (3*plot_range - 7) / 4
normal_line = (-4/3)*(plot_range - contact_x) + contact_y
```

Python Code

```
# --- Create the Plot ---
plt.figure(figsize=(10, 10))

# Plot the complete hyperbola using a contour plot for the level
    where the equation is 0
plt.contour(X, Y, hyperbola_eq, levels=[0], colors='blue',
    linewidths=2)

# Plot the other geometric elements
plt.plot(plot_range, line, color='red', linestyle='--')
plt.plot(plot_range, normal_line, color='green')
plt.scatter(contact_x, contact_y, color='black', s=60, zorder=5)
    # Emphasize the point
```

Python Code

```
# --- Create a custom legend because plt.contour doesn't auto-
    label ---
legend_elements = [
    Line2D([0], [0], color='blue', lw=2, label='Hyperbola: $x^2 -
         xy + 1 = 0$'),
    Line2D([0], [0], color='red', linestyle='--', label='Line:
        $3x - 4y - 7 = 0$'),
    Line2D([0], [0], color='green', label='Normal to Curve'),
    Line2D([0], [0], marker='o', color='w', markerfacecolor='k',
        markersize=8,
           label=f'Point of Contact ({contact_x:.1f}, {contact_y
               :.2f})')
]

plt.title('Geometric Solution with Complete Hyperbola')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
```

Python Code

```
plt.axhline(0, color='grey', linewidth=0.5)
plt.axvline(0, color='grey', linewidth=0.5)
plt.grid(True)
plt.legend(handles=legend_elements)
plt.axis('equal')
plt.xlim(-6, 6)
plt.ylim(-6, 6)
plt.savefig(/media/indhiresh-s/New Volume/Matrix/ee1030-2025/
    ee25btech11027/MATGEO/10.4.3/figs/figure1.png)
plt.show()
```

Plot

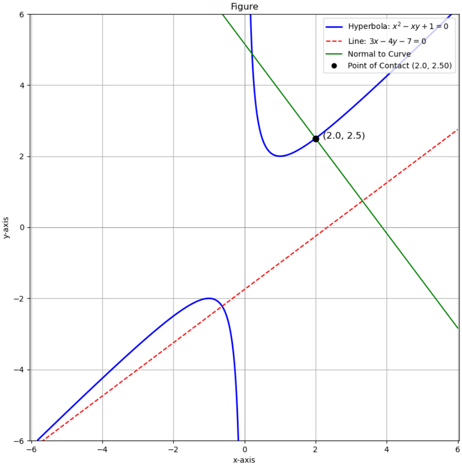
Figure
Hyperbola: $x^2 - xy + 1 = 0$
Line: $3x - 4y - 7 = 0$
Normal to Curve
Point of Contact (2.0, 2.50)
(2.0, 2.5)
x-axis
y-axis
−6
−4
−2
0
2
4
6