4.7.47

Hema Havil - EE25BTECH11050

October 10, 2025

Question

The foot of perpendiculars from the point (2, 3) on the line $y = 3x + 4$ is given by

Theoretical Solution

Let the given point be P=(2,3) and let the foot of perpendicular be Q and let the given line be written as,

$$\mathbf{n}^T\mathbf{x} = c \tag{1}$$

where

$$\mathbf{n} = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$$
$$c = -4$$

then Q is a point on the line, Hence it satisfies the line equation

$$\mathbf{n}^T\mathbf{Q} = c \tag{2}$$

Let $\mathbf{m} = (a, b)$ be the direction vector of the line

$$\mathbf{m}^T\mathbf{n} = 0 \tag{3}$$

$$\begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} 3 \\ -1 \end{pmatrix} = 0 \tag{4}$$

$$3a = b \tag{5}$$

$$\mathbf{m} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \tag{6}$$

Then m is perpendicular to direction vector along PQ

$$\mathbf{m}^T(\mathbf{Q} - \mathbf{P}) = 0 \tag{7}$$

$$\mathbf{m}^T\mathbf{Q} = \mathbf{m}^T\mathbf{P} \tag{8}$$

from equation (0.2) and (0.8) we can write

$$\begin{pmatrix}\mathbf{m} & \mathbf{n}\end{pmatrix}^T \mathbf{Q} = \begin{pmatrix}\mathbf{m}^T\mathbf{P} \\ c\end{pmatrix} \tag{9}$$

We can find the value of $\mathbf{m}^T\mathbf{P}$

$$\mathbf{m}^T\mathbf{P} = \begin{pmatrix}1 \ 3\end{pmatrix}\begin{pmatrix}2 \\ 3\end{pmatrix} = \begin{pmatrix}2+9\end{pmatrix} = \begin{pmatrix}11\end{pmatrix} \tag{10}$$

From this we can find Q, substitute values in (9)

$$\begin{pmatrix}1 & 3 \\ 3 & -1\end{pmatrix}^T \mathbf{Q} = \begin{pmatrix}11 \\ -4\end{pmatrix} \tag{11}$$

$$\begin{pmatrix}1 & 3 \\ 3 & -1\end{pmatrix} \mathbf{Q} = \begin{pmatrix}11 \\ -4\end{pmatrix} \tag{12}$$

This can be solved using augmented matrix and let the augmented matrix be A

Theoretical Solution

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 11 \\ 3 & -1 & -4 \end{pmatrix} \tag{13}$$

$R_2 \rightarrow R_2 - 3R_1$

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 11 \\ 0 & -10 & -37 \end{pmatrix} \tag{14}$$

$R_1 \rightarrow R_1 + \frac{3}{10}R_2$ $\qquad$ $R_2 \rightarrow \frac{-1}{10}R_2$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \frac{-1}{10} \\ 0 & 1 & \frac{-37}{10} \end{pmatrix} \tag{15}$$

Therefore from (0.15) the value of Q is

$$\mathbf{Q} = \begin{pmatrix} \frac{-1}{10} \\ \frac{-37}{10} \end{pmatrix} \tag{16}$$

C Code- Computing the unit vector

```c
#include <stdio.h>
void solve_system(double a00, double a01,
                  double a10, double a11,
                  double b0, double b1,
                  double *rx, double *ry)
{
    double det = a00 * a11 - a01 * a10;
    if (det == 0.0) {
        /* Singular matrix -- not expected here */
        *rx = 0.0;
        *ry = 0.0;
        return;
    }
    *rx = (b0 * a11 - b1 * a01) / det;
    *ry = (a00 * b1 - a10 * b0) / det;
}
```

Python Code using shared output

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

# Load shared library (adjust path if needed)
lib = ctypes.CDLL('./4.7.47.so')

# declare argtypes: 6 doubles for matrix and RHS, and two double*
    for outputs
lib.solve_system.argtypes = [
    ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double,
    ctypes.c_double, ctypes.c_double,
    ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.
        c_double)
]
```

Python Code using shared output

```
# Problem data
P = np.array([2.0, 3.0]) # given point P
# Line: y = 3x + 4
# We form the 2x2 linear system derived in the method:
# [1 3] Q = [m^T P] where m = (1,3)
# [3 -1] [ c ] where c = -4 (because n = (3,-1), n^T x = c -> 3x
    - y = -4)
A = np.array([[1.0, 3.0],
              [3.0, -1.0]])
b = np.array([ (1.0*P[0] + 3.0*P[1]), # m^T P = [1 3] dot P
               -4.0 ]) # c = -4

# Prepare output doubles
rx = ctypes.c_double()
ry = ctypes.c_double()
```

Python Code using shared output

```
        # Call C function
lib.solve_system(A[0,0], A[0,1],
                 A[1,0], A[1,1],
                 b[0], b[1],
                 ctypes.byref(rx), ctypes.byref(ry))

Q = np.array([rx.value, ry.value])

print(Foot of perpendicular Q =, Q)

# Plotting
x = np.linspace(-3, 4, 400)
y_line = 3*x + 4 # the line y = 3x + 4

plt.figure(figsize=(7,6))
plt.plot(x, y_line, label='Line: y = 3x + 4', linewidth=2)
```

Python Code using shared output

```
        # Plot the given point P and foot Q
plt.scatter([P[0]], [P[1]], marker='o', s=80, label='P (2,3)')
plt.scatter([Q[0]], [Q[1]], marker='o', s=80, label=f'Q ({Q[0]:.3
    f}, {Q[1]:.3f})', color='red')

# Draw the perpendicular segment PQ
plt.plot([P[0], Q[0]], [P[1], Q[1]], '--', linewidth=1.8, label='
    Perpendicular PQ')

# For visual reference, draw direction vector of the line at Q (
    scaled)
dir_vec = np.array([1.0, 3.0])
plt.arrow(Q[0], Q[1], 0.6*dir_vec[0], 0.6*dir_vec[1], head_width
    =0.12, head_length=0.18, length_includes_head=True)
```

Python Code using shared output

```
        plt.gca().set_aspect('equal', adjustable='box')
plt.xlim(min(-3, P[0]-2), max(4, P[0]+2))
plt.ylim(min(-1, P[1]-2), max(6, P[1]+3))
plt.xlabel('x')
plt.ylabel('y')
plt.title('Foot of Perpendicular from P')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Plot by python using shared output from c

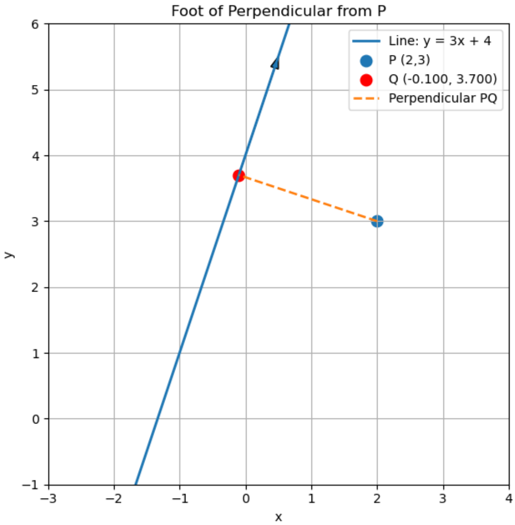

Figure: Plot of the foot of perpendicular of P

Python code for the plot

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

# Given point P and line y = 3x + 4
P = np.array([2.0, 3.0])

# Normal vector (from line 3x - y + 4 = 0) and direction vector
n = np.array([3.0, -1.0]) # normal vector
m = np.array([1.0, 3.0]) # direction vector, since m.T n = 0
c = -4.0 # constant term (3x - y = -4)

# Construct linear system [m^T; n^T] Q = [m^T P; c]
A = np.array([[1.0, 3.0],
              [3.0, -1.0]])
b = np.array([np.dot(m, P), c])
# Solve for Q using numpy
Q = np.linalg.solve(A, b)
print(fFoot of perpendicular Q = ({Q[0]:.3f}, {Q[1]:.3f}))
```

Python code for the plot

```
# Prepare line for plotting
x = np.linspace(-3, 4, 400)
y_line = 3*x + 4 # line: y = 3x + 4

# Plot setup (same style as before)
plt.figure(figsize=(7,6))
plt.plot(x, y_line, label='Line: y = 3x + 4', linewidth=2)
# Plot points P and Q
plt.scatter(P[0], P[1], color='blue', s=80, label='P (2,3)')
plt.scatter(Q[0], Q[1], color='red', s=80, label=f'Q ({Q[0]:.2f},
    {Q[1]:.2f})')
# Draw perpendicular PQ
plt.plot([P[0], Q[0]], [P[1], Q[1]], 'k--', linewidth=1.8, label=
    'Perpendicular PQ')
```

Python code for the plot

```
# Draw a small arrow showing the line direction at Q
dir_vec = m / np.linalg.norm(m)
plt.arrow(Q[0], Q[1], 0.6*dir_vec[0], 0.6*dir_vec[1],
          head_width=0.12, head_length=0.18,
          length_includes_head=True, color='green')
          # Axes formatting
plt.gca().set_aspect('equal', adjustable='box')
plt.xlim(min(-3, P[0]-2), max(4, P[0]+2))
plt.ylim(min(-1, P[1]-2), max(6, P[1]+3))
plt.xlabel('x')
plt.ylabel('y')
plt.title('Foot of Perpendicular from P')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

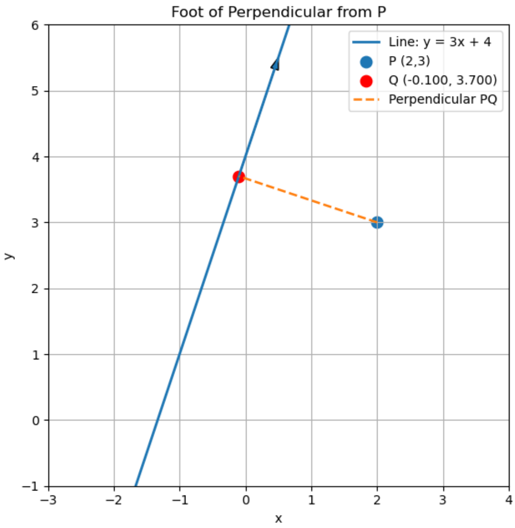

Figure: Plot of foot of perpendicular of point P