Matgeo Presentation - Problem 4.3.38

Revanth Siva Kumar.D – EE25BTECH11048

September 30, 2025

Problem Statement

Find the equation of the line joining the points $(3, 1)$ and $(9, 3)$.

Solution:

Solution :

Given

$$\mathbf{A} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \mathbf{B} = \begin{pmatrix} 9 \\ 3 \end{pmatrix} \tag{0.1}$$

Let us assume line equation to be:

$$\mathbf{n}^T \mathbf{x} = c \tag{0.2}$$

We get the line equation on solving

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \end{pmatrix}^T \mathbf{n} = c \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The line passes through the points from (0.1) substituting, we get:

$$\begin{pmatrix} 3 & 9 \\ 1 & 3 \end{pmatrix}^T \mathbf{n} = c \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{0.3}$$

$$\begin{pmatrix} 3 & 1 \\ 9 & 3 \end{pmatrix} \mathbf{n} = c \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{0.4}$$

Solution:

Now by Gaussian Elimination solve:

$$\left(\begin{array}{cc|c} 3 & 1 & 1 \\ 9 & 3 & 1 \end{array}\right) \tag{0.5}$$

$$R_1 \leftarrow \frac{1}{3}R_1$$
$$\implies \left(\begin{array}{cc|c} 1 & \frac{1}{3} & \frac{1}{3} \\ 9 & 3 & 1 \end{array}\right) \tag{0.6}$$

$$R_2 \leftarrow R_2 - 9R_1$$
$$\implies \left(\begin{array}{cc|c} 1 & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & -2 \end{array}\right) \tag{0.7}$$

By the assumption that line equation is $\mathbf{n}^T\mathbf{x} = 1$ which doesn't pass through origin we are not getting any solution.So our assumption is wrong and origin lies on the line . So consider

$$\mathbf{n}^T\mathbf{x} = 0 \tag{0.8}$$

Solution:

$c = 0$ because origin lies on the line and solving: so now,Assume the line equation:

$$\mathbf{n}^T\mathbf{x} = 0, \quad \mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

Line passes through points **A** and **B**

$$\mathbf{n}^T\mathbf{A} = 0 \implies 3n_1 + 1n_2 = 0 \tag{0.9}$$

$$\mathbf{n}^T\mathbf{B} = 0 \implies 9n_1 + 3n_2 = 0 \tag{0.10}$$

Matrix form:

$$\begin{pmatrix} 3 & 1 \\ 9 & 3 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{0.11}$$

Augmented matrix:

$$\left(\begin{array}{cc|c} 3 & 1 & 0 \\ 9 & 3 & 0 \end{array}\right) \tag{0.12}$$

Solution:

$$R_1 \leftarrow \frac{1}{3} R_1$$

$$\Rightarrow \left(\begin{array}{cc|c} 1 & \frac{1}{3} & 0 \\ 9 & 3 & 0 \end{array}\right) \tag{0.13}$$

$$R_2 \leftarrow R_2 - 9R_1$$

$$\Rightarrow \left(\begin{array}{cc|c} 1 & \frac{1}{3} & 0 \\ 0 & 0 & 0 \end{array}\right) \tag{0.14}$$

From first row:

$$n_1 + \frac{1}{3} n_2 = 0 \implies n_1 = -\frac{1}{3} n_2 \tag{0.15}$$

$$\tag{0.16}$$

Solution:

Let,

$$n_2 = 3 \implies n_1 = -1 \tag{0.17}$$

$$\mathbf{n} = \begin{pmatrix} -1 \\ 3 \end{pmatrix} \tag{0.18}$$

$$\mathbf{n}^T\mathbf{x} = 0 \implies \begin{pmatrix} -1 & 3 \end{pmatrix} \mathbf{x} = 0 \tag{0.19}$$

Final Answer Required Equation is

$$\boxed{\begin{pmatrix} -1 & 3 \end{pmatrix} \mathbf{x} = 0}$$

C Source Code: points.c

```
#include <stdio.h>

// Function to compute normal vector for line through A(3,1) and B(9,3)
void get_normal_vector(double *n1, double *n2) {
    int x1 = 3, y1 = 1;
    int x2 = 9, y2 = 3;

    // Normal vector = [y2-y1, -(x2-x1)]
    *n1 = y2 - y1; // 3 - 1 = 2
    *n2 = -(x2 - x1); // -(9 - 3) = -6
}

int main() {
    double n1, n2;
    get_normal_vector(&n1, &n2);
    printf("Normal␣vector:␣n1=%lf,␣n2=%lf\n", n1, n2);
    return 0;
}
```

Python Script: call c.py

```
import ctypes

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

# Prepare ctypes doubles
n1 = ctypes.c_double()
n2 = ctypes.c_double()

# Call C function
lib.get_normal_vector(ctypes.byref(n1), ctypes.byref(n2))
print("Normal vector from C:", n1.value, n2.value)

# Save normal vector for plotting
normal_vector = (n1.value, n2.value)
```

Python Script: plot.py

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

# ------------------------
# Load C library to get normal vector
# ------------------------
lib = ctypes.CDLL("./points.so")
n1 = ctypes.c_double()
n2 = ctypes.c_double()
lib.get_normal_vector(ctypes.byref(n1), ctypes.byref(n2))
print("Normal␣vector␣from␣C:", n1.value, n2.value)

# ------------------------
# Points A and B
# ------------------------
A = np.array([3, 1])
B = np.array([9, 3])

# Direction vector along line
D = B - A

# Parameter t for plotting line
t = np.linspace(-1, 2, 100)
line_points = A[:, None] + D[:, None]*t
```

Python Script: plot.py

```
# ------------------------
# Plot line and points in 2D
# ------------------------
plt.figure(figsize=(6,6))
plt.plot(line_points[0], line_points[1], color='r', label='Line␣through␣A␣and␣B')
plt.scatter([A[0], B[0]], [A[1], B[1]], color='b', s=50, label='Points␣A␣and␣B')

# Optional: plot normal vector from origin
origin = np.array([0,0])
plt.quiver(*origin, n1.value, n2.value, angles='xy', scale_units='xy', scale=1,
          color='g', label='Normal␣vector')

plt.xlabel('X')
plt.ylabel('Y')
plt.title('Line␣and␣Normal␣Vector␣for␣Points␣(3,1)␣&␣(9,3)')
plt.grid(True)
plt.axis('equal')
plt.legend()
plt.savefig("line_normal_2d.png")
plt.show()
```

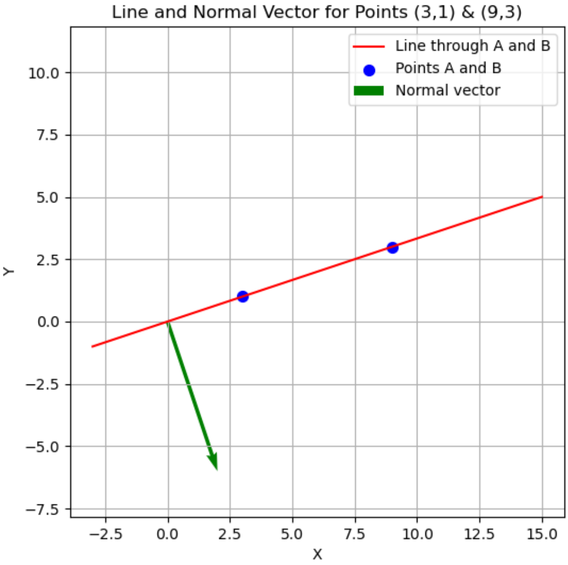

Figure: Plot for the unit vector along PQ