4.2.13

Direction and normal vector

EE25BTECH11010 - Arsh Dhoke

Question

Question:

Find the direction and normal vector for the line $y = x$.

Solution

The line can be written as:

$$x - y = 0 \tag{1}$$

Let

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{n^T} = \begin{pmatrix} 1 & -1 \end{pmatrix}, \quad c = 0 \tag{2}$$

Then:

$$\mathbf{n^T x} = c \tag{3}$$

Where $\mathbf{n}$ is the normal vector of the given line.

Direction Vector

The direction vector of the line is:

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

If the direction vector is given by

$$\mathbf{m} = \begin{pmatrix} 1 \\ m \end{pmatrix} \tag{5}$$

then the normal vector can be written as

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

We can prove this using

$$\mathbf{n}^{\mathsf{T}}\mathbf{m} = 0 \tag{7}$$

Final Vectors

$$\begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0 \tag{8}$$

The normal vector of the line is $\mathbf{n} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

The direction vector of the line is $\mathbf{m} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Graph

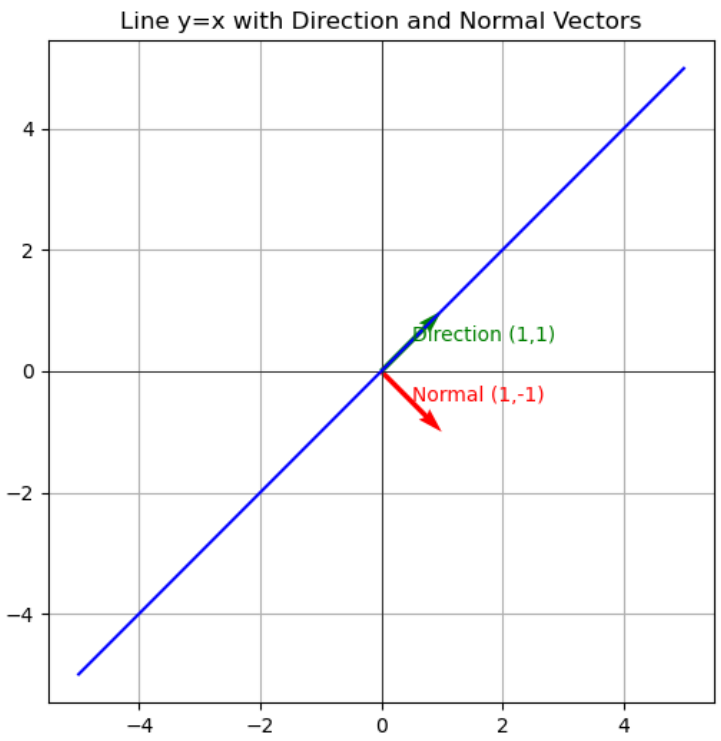

Figure: Graph of the line with direction and normal vectors

C Code

```c
typedef struct {
    double nx;
    double ny;
    double dx;
    double dy;
} LineVectors;

LineVectors find_vectors(double a, double b, double c) {
    LineVectors result;
    result.nx = a;
    result.ny = b;
    result.dx = -b;
    result.dy = a;
    return result;
}
```

Python Code

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

# Define the line y = x
x_vals = np.linspace(-5, 5, 100)
y_vals = x_vals

# Normal and direction vectors
normal_vec = np.array([1, -1])
direction_vec = np.array([1, 1])

# Choose a point on the line (origin for simplicity)
point = np.array([0, 0])

# Create the plot
plt.figure(figsize=(6,6))
plt.axhline(0, color='black', linewidth=0.5) # x-axis
plt.axvline(0, color='black', linewidth=0.5) # y-axis
```

Python Code

```
# Plot the line
plt.plot(x_vals, y_vals, 'b', label='y = x')

# Plot the direction vector (along the line)
plt.quiver(point[0], point[1],
           direction_vec[0], direction_vec[1],
           angles='xy', scale_units='xy', scale=1, color='green')
plt.text(direction_vec[0]/2, direction_vec[1]/2, 'Direction (1,1)
    ', color='green')

# Plot the normal vector (perpendicular to the line)
plt.quiver(point[0], point[1],
           normal_vec[0], normal_vec[1],
           angles='xy', scale_units='xy', scale=1, color='red')
plt.text(normal_vec[0]/2, normal_vec[1]/2, 'Normal (1,-1)', color
    ='red')
```

Python Code

```
# Formatting
plt.xlim(-5,5)
plt.ylim(-5,5)
plt.axis('equal')
plt.grid(True)
plt.title('Line y=x with Direction and Normal Vectors')
plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /4.2.13/figs/q6.png")
plt.show()
```

Python+ C Code

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

# Load shared library
lib = ctypes.CDLL('./code.so')

# Define the struct in Python
class LineVectors(ctypes.Structure):
    _fields_ = [
        ('nx', ctypes.c_double),
        ('ny', ctypes.c_double),
        ('dx', ctypes.c_double),
        ('dy', ctypes.c_double)
    ]

# Function signature
```

Python+ C Code

```
lib.find_vectors.restype = LineVectors
lib.find_vectors.argtypes = [ctypes.c_double, ctypes.c_double,
    ctypes.c_double]

# Call the function for y = x -> x - y = 0 -> a=1, b=-1, c=0
v = lib.find_vectors(1.0, -1.0, 0.0)

# Print results
print("Normal vector:", (v.nx, v.ny))
print("Direction vector:", (v.dx, v.dy))

# Plotting the line and direction vector
# Line equation: a*x + b*y + c = 0 -> y = (-a*x - c)/b
a, b, c = 1.0, -1.0, 0.0
x_vals = np.linspace(-5, 5, 100)
y_vals = (-a * x_vals - c) / b

plt.figure(figsize=(6,6))
```

Python+ C Code

```
plt.plot(x_vals, y_vals, label='Line y = x', color='blue')

# Plot direction vector as an arrow from origin
plt.quiver(0, 0, v.dx, v.dy, angles='xy', scale_units='xy', scale
    =1, color='red', label='Direction vector')

# Plot normal vector as arrow from origin
plt.quiver(0, 0, v.nx, v.ny, angles='xy', scale_units='xy', scale
    =1, color='green', label='Normal vector')

plt.xlim(-6,6)
plt.ylim(-6,6)
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.legend()
plt.title("Line y = x with Normal and Direction Vectors")
plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /4.2.13/figs/q6.png")
plt.show()
```