2.10.3

AI25BTECH11014 - Gooty Suhas

September 28, 2025

Question

Find the unit vector perpendicular to the plane determined by the points

$$\mathbf{P} = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$$

Theoretical Solution

We begin by computing two vectors that lie on the plane:

$$\mathbf{Q} - \mathbf{P} = \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix}, \quad \mathbf{R} - \mathbf{P} = \begin{pmatrix} -1 \\ 3 \\ -1 \end{pmatrix}$$

We now block the orthogonality conditions into a matrix system:

$$\begin{bmatrix} 1 & 1 & -3 \\ -1 & 3 & -1 \end{bmatrix} \mathbf{N} = \mathbf{0}$$

Theoretical Solution

Apply row operations:

$$R_2 \leftarrow R_1 + R_2 \Rightarrow \begin{bmatrix} 1 & 1 & -3 \\ 0 & 4 & -4 \end{bmatrix} \Rightarrow \mathbf{N} = \begin{pmatrix} 8 \\ 2 \\ 4 \end{pmatrix}$$

Compute magnitude:

$$\|\mathbf{N}\| = \sqrt{8^2 + 2^2 + 4^2} = \sqrt{84}$$

Unit vector:

$$\hat{n} = \frac{1}{\sqrt{84}} \begin{pmatrix} 8 \\ 2 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{8}{\sqrt{84}} \\ \frac{2}{\sqrt{84}} \\ \frac{4}{\sqrt{84}} \end{pmatrix}$$

Final Answer

$$\boxed{\hat{n} = \begin{pmatrix} \frac{8}{\sqrt{84}} \\ \frac{2}{\sqrt{84}} \\ \frac{4}{\sqrt{84}} \end{pmatrix}}$$

This is the unit vector perpendicular to the plane defined by points $\mathbf{P}, \mathbf{Q}, \mathbf{R}$.

Python Code (Part 1)

```
import numpy as np

P = np.array([1, -1, 2])
Q = np.array([2, 0, -1])
R = np.array([0, 2, 1])

A = Q - P
B = R - P

M = np.array([A, B])
U, S, Vt = np.linalg.svd(M)
```

Python Code (Part 2)

```
N = Vt[-1] # Null space vector

unit_vector = N / np.linalg.norm(N)
print("Unit vector:", unit_vector)
```

C Code for .so File (Part 1)

```c
#include <math.h>

void normal_vector(float* A,
                   float* B,
                   float* out) {

  float z = 1.0;

  float denom = A[0]*B[1] - B[0]*A[1];
  float x = (B[1]*A[2] - A[1]*B[2]) / denom;
```

C Code for .so File (Part 2)

```c
    float y = (A[0]*B[2] - B[0]*A[2]) / denom;

    out[0] = x;
    out[1] = y;
    out[2] = z;

    float mag = sqrt(out[0]*out[0] +
                     out[1]*out[1] +
                     out[2]*out[2]);

    for(int i=0;i<3;i++) out[i]/=mag;
}
```

Python Code Using .so File (Part 1)

```
import ctypes
import numpy as np

lib = ctypes.CDLL('./libnormal.so')
lib.normal_vector.argtypes = [
  ctypes.POINTER(ctypes.c_float),
  ctypes.POINTER(ctypes.c_float),
  ctypes.POINTER(ctypes.c_float)
]

P = np.array([1, -1, 2], np.float32)
```

Python Code Using .so File (Part 2)

```
Q = np.array([2, 0, -1], np.float32)
R = np.array([0, 2, 1], np.float32)

A = Q - P
B = R - P
out = np.zeros(3, np.float32)

lib.normal_vector(
  A.ctypes.data_as(ctypes.POINTER(ctypes.c_float)),
  B.ctypes.data_as(ctypes.POINTER(ctypes.c_float)),
  out.ctypes.data_as(ctypes.POINTER(ctypes.c_float))
)

print("Unit vector:", out)
```

Plot

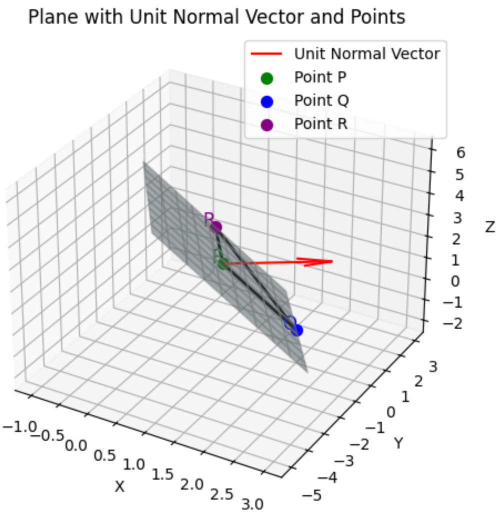

Figure: Plane and its normal