2.10.3

AI25BTECH11014 - Gooty Suhas

September 17, 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:

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

To find a vector perpendicular to the plane, we take the cross product:

$$\mathbf{N} = \mathbf{PQ} \times \mathbf{PR}$$

Theoretical Solution

Using the formula:

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}$$

We compute:

$$\mathbf{N} = \begin{pmatrix} (1)(-1) - (-3)(3) \\ (-3)(-1) - (1)(1) \\ (1)(3) - (1)(-1) \end{pmatrix} = \begin{pmatrix} 8 \\ 2 \\ 4 \end{pmatrix}$$

Theoretical Solution

Now compute the magnitude of **N**:

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

To get the 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

```python
import numpy as np

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

PQ = Q - P
PR = R - P

N = np.cross(PQ, PR)
mag = np.linalg.norm(N)
unit = N / mag

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

C Code for .so File

```c
#include <math.h>

void normal_vector(float* PQ, float* PR, float* out) {
  out[0] = PQ[1]*PR[2] - PQ[2]*PR[1];
  out[1] = PQ[2]*PR[0] - PQ[0]*PR[2];
  out[2] = PQ[0]*PR[1] - PQ[1]*PR[0];

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

  out[0] /= mag;
  out[1] /= mag;
  out[2] /= mag;
}
```

Python Code Using .so (Part 1)

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

Python Code Using .so (Part 2)

```python
PQ = Q - P
PR = R - P
out = np.zeros(3, np.float32)

lib.normal_vector(
  PQ.ctypes.data_as(ctypes.POINTER(ctypes.c_float)),
  PR.ctypes.data_as(ctypes.POINTER(ctypes.c_float)),
  out.ctypes.data_as(ctypes.POINTER(ctypes.c_float))
)

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

Plot

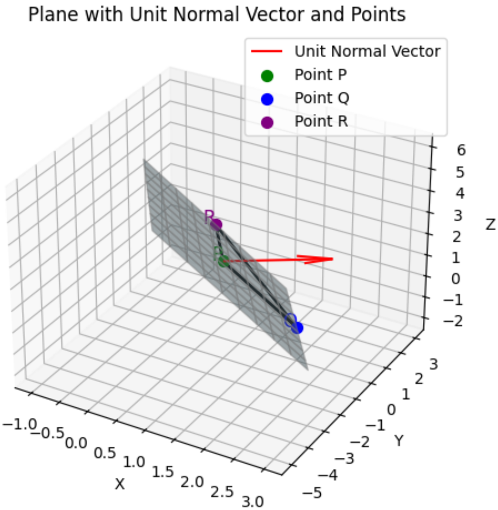

Figure: Plane and its normal