12.249

INDHIRESH S - EE25BTECH11027

10 October, 2025

Question

A plane contains the following three points: $\mathbf{P}(2, 1, 5), \mathbf{Q}(-1, 3, 4)$ and $\mathbf{R}(3, 0, 6)$. The vector perpendicular to the above plane can be represented as

Given points are:

$$\mathbf{P} = \begin{pmatrix} 2 \\ 1 \\ 5 \end{pmatrix} \;, \mathbf{Q} = \begin{pmatrix} -1 \\ 3 \\ 4 \end{pmatrix} \;\; and \;\; \mathbf{R} = \begin{pmatrix} 3 \\ 0 \\ 6 \end{pmatrix} \tag{1}$$

The equation of plane through the points **P**,**Q** and **R** can be given as

$$\begin{pmatrix} \mathbf{P} & \mathbf{Q} & \mathbf{R} \end{pmatrix}^T \mathbf{n} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{2}$$

Theoretical Solution

Where $\mathbf{n}$ is the normal to the plane

$$\begin{pmatrix} 2 & -1 & 3 \\ 1 & 3 & 0 \\ 5 & 4 & 6 \end{pmatrix}^T \mathbf{n} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{3}$$

$$\begin{pmatrix} 2 & 1 & 5 \\ -1 & 3 & 4 \\ 3 & 0 & 6 \end{pmatrix} \mathbf{n} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{4}$$

Now forming the augmented matrix and performing row operations

$$\left(\begin{array}{ccc|c} 2 & 1 & 5 & 1 \\ -1 & 3 & 4 & 1 \\ 3 & 0 & 6 & 1 \end{array}\right) \xleftrightarrow{R_2 \leftarrow -R_2} \left(\begin{array}{ccc|c} 2 & 1 & 5 & 1 \\ 1 & -3 & -4 & -1 \\ 3 & 0 & 6 & 1 \end{array}\right) \tag{5}$$

Theoretical solution

$$\left(\begin{array}{ccc|c} 2 & 1 & 5 & 1 \\ 1 & -3 & -4 & -1 \\ 3 & 0 & 6 & 1 \end{array}\right) \xleftrightarrow[R_1 \leftarrow R_1 - 2R_2]{R_3 \leftarrow R_3 - 3R_2} \left(\begin{array}{ccc|c} 0 & 7 & 13 & 3 \\ 1 & -3 & -4 & -1 \\ 0 & 9 & 18 & 4 \end{array}\right) \tag{6}$$

$$\left(\begin{array}{ccc|c} 0 & 7 & 13 & 3 \\ 1 & -3 & -4 & -1 \\ 0 & 9 & 18 & 4 \end{array}\right) \xleftrightarrow{R_3 \leftarrow 7R_3 - 9R_1} \left(\begin{array}{ccc|c} 0 & 7 & 13 & 3 \\ 1 & -3 & -4 & -1 \\ 0 & 0 & 9 & 1 \end{array}\right) \tag{7}$$

$$\left(\begin{array}{ccc|c} 0 & 7 & 13 & 3 \\ 1 & -3 & -4 & -1 \\ 0 & 0 & 9 & 1 \end{array}\right) \xleftrightarrow[R_1 \leftarrow \frac{1}{7}R_1]{R_3 \leftarrow \frac{1}{9}R_3} \left(\begin{array}{ccc|c} 0 & 1 & \frac{13}{7} & \frac{3}{7} \\ 1 & -3 & -4 & -1 \\ 0 & 0 & 1 & \frac{1}{9} \end{array}\right) \tag{8}$$

Theoretical solution

$$\left(\begin{array}{ccc|c} 0 & 1 & \frac{13}{7} & \frac{3}{7} \\ 1 & -3 & -4 & -1 \\ 0 & 0 & 1 & \frac{1}{9} \end{array}\right) \xleftrightarrow[R_2 \leftarrow R_2 + 4R_3]{R_1 \leftarrow R_1 - \frac{13}{7} R_3} \left(\begin{array}{ccc|c} 0 & 1 & 0 & \frac{2}{9} \\ 1 & -3 & 0 & -\frac{5}{9} \\ 0 & 0 & 1 & \frac{1}{9} \end{array}\right) \tag{9}$$

$$\left(\begin{array}{ccc|c} 0 & 1 & 0 & \frac{2}{9} \\ 1 & -3 & 0 & -\frac{5}{9} \\ 0 & 0 & 1 & \frac{1}{9} \end{array}\right) \xleftrightarrow{R_2 \leftarrow R_2 + 3R_1} \left(\begin{array}{ccc|c} 0 & 1 & 0 & \frac{2}{9} \\ 1 & 0 & 0 & \frac{1}{9} \\ 0 & 0 & 1 & \frac{1}{9} \end{array}\right) \tag{10}$$

$$\left(\begin{array}{ccc|c} 0 & 1 & 0 & \frac{2}{9} \\ 1 & 0 & 0 & \frac{1}{9} \\ 0 & 0 & 1 & \frac{1}{9} \end{array}\right) \xleftrightarrow{R_2 \longleftrightarrow R_1} \left(\begin{array}{ccc|c} 1 & 0 & 0 & \frac{1}{9} \\ 0 & 1 & 0 & \frac{2}{9} \\ 0 & 0 & 1 & \frac{1}{9} \end{array}\right) \tag{11}$$

The equation of plane can be given as:

$$\begin{pmatrix} 1 & 2 & 1 \end{pmatrix} \mathbf{x} = 9 \tag{12}$$

Theoretical solution

Normal of the plane is:

$$\mathbf{n} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \tag{13}$$

This normal vector is perpendicular to the plane.

C Code

```c
typedef struct {
    double x, y, z;
} Vector3D;

void find_normal_vector_lib(Vector3D p, Vector3D q, Vector3D r,
    Vector3D* normal_out) {
    Vector3D vec_pq = {q.x - p.x, q.y - p.y, q.z - p.z};
    Vector3D vec_pr = {r.x - p.x, r.y - p.y, r.z - p.z};

    normal_out->x = (vec_pq.y * vec_pr.z) - (vec_pq.z * vec_pr.y)
        ;
    normal_out->y = (vec_pq.z * vec_pr.x) - (vec_pq.x * vec_pr.z)
        ;
    normal_out->z = (vec_pq.x * vec_pr.y) - (vec_pq.y * vec_pr.x)
        ;
}
```

Python Code

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

def plot_3d(p, q, r, normal):
    Visualizes the plane, points, and normal vector with coordinate labels.
    fig = plt.figure(figsize=(10, 8))
    ax = fig.add_subplot(111, projection='3d')

    points = np.array([p, q, r])
    ax.scatter(points[:,0], points[:,1], points[:,2], color='red', s=100)

    # Add text labels with coordinates
    ax.text(p[0], p[1], p[2] + 0.2, f' P({p[0]}, {p[1]}, {p[2]})', color='darkred')
```

```python
ax.text(q[0], q[1], q[2] + 0.2, f' Q({q[0]}, {q[1]}, {q[2]})',
    color='darkred')
ax.text(r[0], r[1], r[2] + 0.2, f' R({r[0]}, {r[1]}, {r[2]})'
    , color='darkred')

# Create and plot the plane
d = np.dot(normal, p)
x_range = np.linspace(min(points[:,0])-2, max(points[:,0])+2,
    10)
y_range = np.linspace(min(points[:,1])-2, max(points[:,1])+2,
    10)
xx, yy = np.meshgrid(x_range, y_range)
zz = (d - normal[0] * xx - normal[1] * yy) / normal[2]
ax.plot_surface(xx, yy, zz, alpha=0.4, color='cyan')
```

Python Code

```
# Plot the normal vector
 ax.quiver(p[0], p[1], p[2], normal[0], normal[1], normal[2],
           length=4, normalize=True, color='black',
               arrow_length_ratio=0.2, label='Normal Vector')

 ax.set_xlabel('X-axis'); ax.set_ylabel('Y-axis'); ax.
     set_zlabel('Z-axis')
 ax.set_title('Plane and Perpendicular Vector')
 ax.legend()
 plt.savefig(/media/indhiresh-s/New Volume/Matrix/ee1030-2025/
     ee25btech11027/MATGEO/12.249/figs/figure1.png)
 plt.show()

class Vector3D(ctypes.Structure):
    _fields_ = [(x, ctypes.c_double), (y, ctypes.c_double), (z,
        ctypes.c_double)]
```

Python Code

```python
c_lib_file = 'plane.so'

c_lib = ctypes.CDLL(os.path.abspath(c_lib_file))
c_lib.find_normal_vector_lib.argtypes = [Vector3D, Vector3D,
    Vector3D, ctypes.POINTER(Vector3D)]
c_lib.find_normal_vector_lib.restype = None

p_pt, q_pt, r_pt = Vector3D(2, 1, 5), Vector3D(-1, 3, 4),
    Vector3D(3, 0, 6)
normal_out = Vector3D()
c_lib.find_normal_vector_lib(p_pt, q_pt, r_pt, ctypes.byref(
    normal_out))
```

Python Code

```
    normal_vector = np.array([normal_out.x, normal_out.y,
        normal_out.z])
p_np, q_np, r_np = np.array([p_pt.x, p_pt.y, p_pt.z]), np.array([
    q_pt.x, q_pt.y, q_pt.z]), np.array([r_pt.x, r_pt.y, r_pt.z])

print(fPerpendicular vector (from C): {normal_vector})
plot_3d(p_np, q_np, r_np, normal_vector)
```