4.7.44

RAVULA SHASHANK REDDY - EE25BTECH11047

September 28, 2025

Question

Find the distance of the plane from the origin.

$$\mathbf{r}^T \begin{pmatrix} \frac{2}{7} \\ \frac{3}{7} \\ -\frac{6}{7} \end{pmatrix} = 1$$

Equation

Plane Equation

$$\mathbf{n}^T\mathbf{x} = 1$$

Distance of a point from a plane

$$d = \frac{|\mathbf{n}^T\mathbf{x} - 1|}{\|\mathbf{n}\|}.$$

Solution

Equation of plane

$$\mathbf{n}^T\mathbf{x} = 1, \tag{1}$$

$$\mathbf{n} = \begin{pmatrix} \frac{2}{7} \\ \frac{3}{7} \\ -\frac{6}{7} \end{pmatrix}. \tag{2}$$

$$d = \frac{|\mathbf{n}^T\mathbf{x}_0 - 1|}{\|\mathbf{n}\|}. \tag{3}$$

$$d = \frac{|\mathbf{n}^T\mathbf{0} - 1|}{\|\mathbf{n}\|} \tag{4}$$

$$= \frac{|0 - 1|}{\|\mathbf{n}\|} \tag{5}$$

$$= \frac{1}{\|\mathbf{n}\|}. \tag{6}$$

$$\tag{7}$$

Solution

$$\|\mathbf{n}\| = \sqrt{\mathbf{n}^T\mathbf{n}} \tag{8}$$

$$= \sqrt{\left(\frac{2}{7}\right)^2 + \left(\frac{3}{7}\right)^2 + \left(-\frac{6}{7}\right)^2} \tag{9}$$

$$= \sqrt{\frac{49}{49}} \tag{10}$$

$$= 1. \tag{11}$$

Therefore, the required distance is

$$d = \frac{1}{1} = \boxed{1}. \tag{12}$$

C Code

```c
           #include <stdio.h>
#include <math.h>

int main() {
    // Plane normal vector components
    double n[3] = {2.0/7.0, 3.0/7.0, -6.0/7.0};

    // Point (origin)
    double x[3] = {0.0, 0.0, 0.0};

    // Compute n^T x
    double dot = 0.0;
    for (int i = 0; i < 3; i++) {
        dot += n[i] * x[i];
    }
```

C Code

```
    // Numerator = |n^T x - 1|
    double numerator = fabs(dot - 1.0);

    // Denominator = ||n||
    double norm = sqrt(n[0]*n[0] + n[1]*n[1] + n[2]*n[2]);

    // Distance
    double d = numerator / norm;

    printf(The distance of the plane from the origin is: %.2f\n,
        d);

    return 0;
}
```

Python Direct Code

```
    import numpy as np
import matplotlib.pyplot as plt
from libs.line.funcs import *
from libs.triangle.funcs import *

# --- Plane normal and constant ---
n = np.array([2/7, 3/7, -6/7])
c = 1

# --- Origin ---
origin = np.array([0, 0, 0])

# --- Closest point from origin to plane ---
closest_point = (c / np.dot(n, n)) * n
d = np.linalg.norm(closest_point)

print(Closest point:, closest_point)
print(Distance:, d)
```

Python Direct Code

```
# --- Basis vectors lying in the plane ---
u = np.array([n[1], -n[0], 0])
if np.allclose(u, 0):
    u = np.array([0, n[2], -n[1]])
v = np.cross(n, u)

u = u / np.linalg.norm(u) * 2
v = v / np.linalg.norm(v) * 2

# Define four corners of a plane patch (parallelogram)
p1 = closest_point + u + v
p2 = closest_point - u + v
p3 = closest_point - u - v
p4 = closest_point + u - v

# --- Plotting ---
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection=3d)
```

Python Direct Code

```
# Plot plane patch (semi-transparent)
ax.plot_trisurf([p1[0], p2[0], p3[0], p4[0]],
                [p1[1], p2[1], p3[1], p4[1]],
                [p1[2], p2[2], p3[2], p4[2]],
                color=cyan, alpha=0.4)

# Line (Origin  Closest Point)
ax.plot([origin[0], closest_point[0]],
        [origin[1], closest_point[1]],
        [origin[2], closest_point[2]],
        color=black, linestyle=--)

# Origin (red with label)
ax.scatter(*origin, color=red, s=60)
ax.text(*origin, (0,0,0), color=red, fontsize=9, weight=bold)
```

Python Direct Code

```
# Closest Point (blue with fraction coords as label)
ax.scatter(*closest_point, color=blue, s=60)
ax.text(*closest_point,
        (2/7, 3/7, -6/7),
        color=blue, fontsize=9, weight=bold)

# Axes labels
ax.set_xlabel(X-axis)
ax.set_ylabel(Y-axis)
ax.set_zlabel(Z-axis)
ax.set_title(Plane with Closest Point from Origin)

plt.show()
```

Python Shared

```python
import ctypes
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Define a simple vector class using ctypes
class MyVec(ctypes.Structure):
    _fields_ = [(x, ctypes.c_double),
                (y, ctypes.c_double),
                (z, ctypes.c_double)]

    def __repr__(self):
        return f({self.x}, {self.y}, {self.z})

# Plane normal n and constant d for n^T x = 1
n = MyVec(2/7, 3/7, -6/7)
d = 1
```

Python Shared

```
# Distance from origin formula: |d| / ||n||
norm_n = (n.x**2 + n.y**2 + n.z**2)**0.5
distance = abs(d) / norm_n
print(Distance of plane from origin:, distance)

# Plotting the plane
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Create a grid
xx, yy = np.meshgrid(range(-10, 11), range(-10, 11))
# Solve for z using plane equation: n^T x = 1 -> z = (1 - n_x*x -
    n_y*y)/n_z
zz = (d - n.x*xx - n.y*yy) / n.z
```

Python Shared

```
# Plot the plane
ax.plot_surface(xx, yy, zz, alpha=0.5, color='cyan')

# Plot the origin and normal vector
ax.scatter(0, 0, 0, color='red', s=100, label='Origin')
ax.quiver(0, 0, 0, n.x, n.y, n.z, length=5, color='blue', label='
    Normal n')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Plane and its normal vector')
ax.legend()
plt.show()
```

Plot

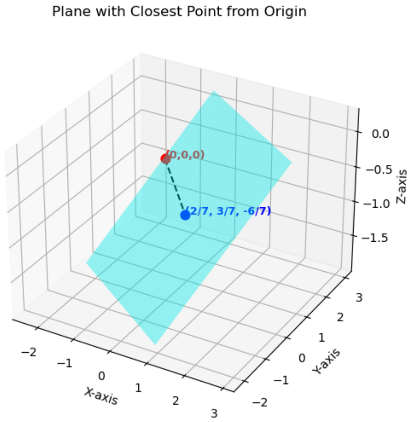
Plane with Closest Point from Origin
(0,0,0)
(2/7, 3/7, -6/7)
0.0
−0.5
−1.0
−1.5
Z-axis
−2
−1
0
1
2
3
X-axis
−2
−1
0
1
2
3
Y-axis

Figure: