4.6.1

Bhargav - EE25BTECH11013

September 8, 2025

Question

The distance between the parallel planes

$$2x + y - 2z - 6 = 0 \tag{1}$$

$$4x + 2y - 4z = 0 \tag{2}$$

Solution

The 2 given planes are parallel since their normal vectors are the same

The normal vector of the planes $\mathbf{n}$ is

$$\mathbf{n} = \begin{pmatrix} 2 \\ 1 \\ -2 \end{pmatrix} \tag{3}$$

Solution

The distance between the planes is given by this formula

$$\text{Distance} = \frac{\left|\mathbf{n}^{\mathbf{T}}\mathbf{p} - d\right|}{\|\mathbf{n}\|} \tag{4}$$

$$\|\mathbf{n}\| = \left(\sqrt{(2)^2 + (1)^2 + (-2)^2}\right) = 3 \tag{5}$$

Where **p** represents a point on the second plane

$$\mathbf{p} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, d = 6 \tag{6}$$

Solution

Substituting these values in the Distance formula, we get

$$\therefore \text{Distance} = \frac{\left| \begin{pmatrix} 2 & 1 & -2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - 6 \right|}{3} \tag{7}$$

$$\text{Distance} = \frac{|-6|}{3} \tag{8}$$

$$\text{Distance} = 2 \tag{9}$$

C Code

```c
#include <math.h>
double norm(double *n) {
    return sqrt(n[0]*n[0] + n[1]*n[1] + n[2]*n[2]);
}
double plane_distance(double *n, double a, double b) {
    double num = fabs(a - b);
    double denom = norm(n);
    return num / denom;
}
```

Python + C Code

```
import numpy as np
import matplotlib.pyplot as plt
import ctypes
lib = ctypes.CDLL(./libdistance.so)
lib.plane_distance.argtypes = [ctypes.POINTER(ctypes.c_double),
    ctypes.c_double, ctypes.c_double]
lib.plane_distance.restype = ctypes.c_double
n = np.array([2.0, 1.0, -2.0], dtype=np.double)
a = -6.0
b = 0.0
n_ptr = n.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
dist = lib.plane_distance(n_ptr, a, b)
print(Distance =, dist)
fig = plt.figure()
ax3d = fig.add_subplot(projection=3d)
x, y = np.meshgrid(range(-10, 11), range(-10, 11))
z1 = (2*x + y - 6) / 2
z2 = (2*x + y) / 2
```

Python + C Code

```
ax3d.plot_surface(x, y, z1, alpha=0.5, color='blue')
ax3d.plot_surface(x, y, z2, alpha=0.5, color='red')
ax3d.plot([], [], [], color='blue', label=2x + y - 2z - 6 = 0)
ax3d.plot([], [], [], color='red', label=2x + y - 2z = 0)
ax3d.set_xlabel(X)
ax3d.set_ylabel(Y)
ax3d.set_zlabel(Z)
ax3d.set_title(fTwo Parallel Planes\nDistance = {dist})
ax3d.legend(loc='upper right')
plt.savefig(/Users/bhargavkrish/Desktop/BackupMatrix/
    ee25btech11013/matgeo/4.6.1/figs/Figure_1.png)
plt.show()
```

Python Code

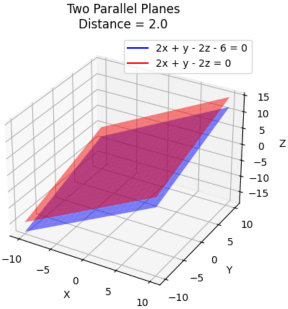
```
import numpy as np
import matplotlib.pyplot as plt
def distance(n, a, b):
    n = np.array(n, dtype=float)
    return abs(a - b) / np.linalg.norm(n)
n = np.array([2, 1, -2])
a = -6
b = 0
dist = distance(n, a, b)
print(Distance =, dist)
fig = plt.figure()
ax3d = fig.add_subplot(projection=3d)
x, y = np.meshgrid(range(-10, 11), range(-10, 11))
z1 = (2*x + y - 6) / 2
z2 = (2*x + y) / 2
ax3d.plot_surface(x, y, z1, alpha=0.5, color='blue')
ax3d.plot_surface(x, y, z2, alpha=0.5, color='red')
```

Python Code

```

ax3d.plot([], [], [], color='blue', label=2x + y - 2z - 6 = 0)
ax3d.plot([], [], [], color='red', label=2x + y - 2z = 0)
ax3d.set_xlabel(X)
ax3d.set_ylabel(Y)
ax3d.set_zlabel(Z)
ax3d.set_title(fTwo Parallel Planes\nDistance = {dist})
ax3d.legend(loc='upper right')
plt.savefig(/Users/bhargavkrish/Desktop/BackupMatrix/
    ee25btech11013/matgeo/4.6.1/figs/Figure_1.png)
plt.show()
```

Figure


Two Parallel Planes
Distance = 2.0
2x + y - 2z - 6 = 0
2x + y - 2z = 0
15
10
5
0
−5
−10
−15
Z
10
5
0
−5
−10
Y
−10
−5
0
5
10
X