4.10.10

Distance between 2 points

EE25BTECH11010 - Arsh Dhoke

Question

Find the distance of the point $(-1, -5, -10)$ from the point of intersection of the line $\mathbf{r} = 2\mathbf{i} - \mathbf{j} - 2\mathbf{k} + \lambda(3\mathbf{i} + 4\mathbf{j} + 2\mathbf{k})$ and the plane $\mathbf{r} \cdot (\mathbf{i} - \mathbf{j} + \mathbf{k}) = 5$.

Input parameters

Description	Vector
P	$\begin{pmatrix} -1 \\ -5 \\ -10 \end{pmatrix}$
a	$\begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix}$
b	$\begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix}$
n	$\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$

Intersection

At intersection:

$$\mathbf{n}^T(\mathbf{a} + \lambda \mathbf{b}) = c \tag{1}$$

$$\lambda = \frac{c - \mathbf{n}^T\mathbf{a}}{\mathbf{n}^T\mathbf{b}} \tag{2}$$

$$\mathbf{x} = \mathbf{a} + \left(\frac{c - \mathbf{n}^T\mathbf{a}}{\mathbf{n}^T\mathbf{b}} \right) \mathbf{b} \tag{3}$$

Computation

$$\mathbf{n}^T\mathbf{a} = \begin{pmatrix} 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix} = 1 \tag{4}$$

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

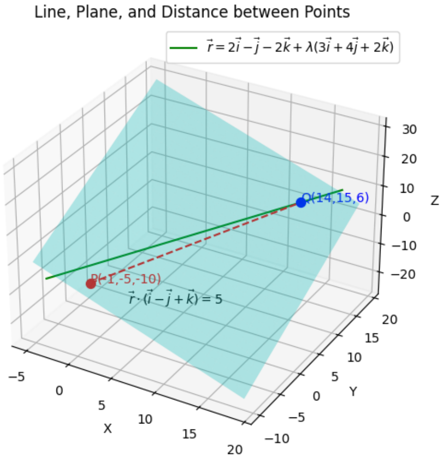
$$\mathbf{x} = \begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix} + \frac{(5-1)}{1} \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 14 \\ 15 \\ 6 \end{pmatrix} \tag{6}$$

Distance

$$\|\mathbf{P}-\mathbf{x}\| = \sqrt{(14-(-1))^2+(15-(-5))^2+(6-(-10))^2} \tag{7}$$

$$= \sqrt{15^2+20^2+16^2} = \sqrt{881} \tag{8}$$

Graph



Line, Plane, and Distance between Points
$\vec{r} = 2\vec{i} - \vec{j} - 2\vec{k} + \lambda(3\vec{i} + 4\vec{j} + 2\vec{k})$
Q(14,15,6)
P(-1,-5,-10)
$\vec{r} \cdot (\vec{i} - \vec{j} + \vec{k}) = 5$
X
Y
Z

C Code

```c
#include <math.h>

double point_line_plane_distance(
    double Px, double Py, double Pz,
    double ax, double ay, double az,
    double bx, double by, double bz,
    double nx, double ny, double nz,
    double c
) {
    double dot_na = nx*ax + ny*ay + nz*az;
    double dot_nb = nx*bx + ny*by + nz*bz;

    double lambda = (c - dot_na) / dot_nb;

    double Xx = ax + lambda * bx;
    double Xy = ay + lambda * by;
    double Xz = az + lambda * bz;
```

C Code

```c
    double dx = Px - Xx;
    double dy = Py - Xy;
    double dz = Pz - Xz;

    return sqrt(dx*dx + dy*dy + dz*dz);
}
```

Python Code

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

# Given data
a = np.array([2, -1, -2]) # point on line
b = np.array([3, 4, 2]) # direction vector of line
n = np.array([1, -1, 1]) # normal vector of plane
P = np.array([-1, -5, -10]) # external point

# Intersection point with plane
lam = (5 - np.dot(n, a)) / np.dot(n, b)
Q = a + lam * b

# Distance
dist = np.linalg.norm(Q - P)
print("Intersection Point Q =", Q)
print("Distance =", dist)
```

Python Code

```
# Create plot
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')

# Plot line with legend
lambdas = np.linspace(-2, 5, 100)
line_points = np.array([a + t*b for t in lambdas])
ax.plot(line_points[:,0], line_points[:,1], line_points[:,2], 'g'
    , label=r"$\vec{r}=2\vec{i}-\vec{j}-2\vec{k}+\lambda(3\vec{i
    }+4\vec{j}+2\vec{k})$")

# Plot plane
xx, yy = np.meshgrid(range(-5, 20), range(-10, 20))
zz = (5 - n[0]*xx - n[1]*yy) / n[2]
ax.plot_surface(xx, yy, zz, alpha=0.3, color='cyan')

# Plot points
ax.scatter(*P, color='red', s=50)
```

Python Code

```
ax.scatter(*Q, color='blue', s=50)

# Annotate points
ax.text(P[0], P[1], P[2], "P(-1,-5,-10)", color='red')
ax.text(Q[0], Q[1], Q[2], "Q(14,15,6)", color='blue')

# Draw distance vector PQ
ax.plot([P[0], Q[0]], [P[1], Q[1]], [P[2], Q[2]], 'r--')

# Annotate plane equation
ax.text(5, -8, (5 - n[0]*5 - n[1]*(-8))/n[2],
        r"$\vec{r}\cdot(\vec{i}-\vec{j}+\vec{k})=5$",
        color='black')

# Labels
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
```

Python Code

```
ax.set_title("Line, Plane, and Distance between Points")

# Only legend for the line
ax.legend()
 plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /4.10.10/figs/q8.png")
plt.show()
```

Python+ C Code

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

lib = ctypes.CDLL("./code.so")
lib.point_line_plane_distance.restype = ctypes.c_double
lib.point_line_plane_distance.argtypes = [ctypes.c_double]*13

def distance(P, a, b, n, c):
    return lib.point_line_plane_distance(
        P[0], P[1], P[2],
        a[0], a[1], a[2],
        b[0], b[1], b[2],
        n[0], n[1], n[2],
        c
    )
```

```
# Input
P = [-1, -5, -10]
a = [2, -1, -2]
b = [3, 4, 2]
n = [1, -1, 1]
c = 5

# Compute distance
d = distance(P, a, b, n, c)
print("Distance =", d)

# Compute intersection point in Python (needed for plotting)
dot_na = np.dot(n, a)
dot_nb = np.dot(n, b)
lam = (c - dot_na) / dot_nb
X = np.array(a) + lam * np.array(b)
```

Python+ C Code

```
# Plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Line (parametric points)
t = np.linspace(-5, 5, 50)
line_pts = np.array([a + np.array(b)*ti for ti in t])
ax.plot(line_pts[:,0], line_pts[:,1], line_pts[:,2], 'b-', label=
    'Line')

# Plane (mesh grid)
xx, yy = np.meshgrid(range(-10, 11), range(-10, 11))
zz = (c - n[0]*xx - n[1]*yy)/n[2]
ax.plot_surface(xx, yy, zz, alpha=0.3, color='cyan')

# Plot P and intersection X
ax.scatter(*P, color='red', s=50, label='Point P')
ax.scatter(*X, color='green', s=50, label='Intersection X')
```

```
# Distance line (P-X)
ax.plot([P[0], X[0]], [P[1], X[1]], [P[2], X[2]], 'r--', label='
    Distance')

# Labels
ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.legend()
 plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
     /4.10.10/figs/q8.png")
plt.show()
```