4.11.25

Revanth Siva Kumar.D - EE25BTECH11048

October 1, 2025

Question

Find the distance of the point $\left(1, -2, 9\right)$ from the point of intersection of the line

$$\mathbf{r} = 4\hat{i} + 2\hat{j} + 7\hat{k} + \lambda\left(3\hat{i} + 4\hat{j} + 2\hat{k}\right)$$

and the plane

$$\mathbf{r} \cdot \left(\hat{i} - \hat{j} + \hat{k}\right) = 10.$$

Theoretical Solution

The line is

$$\mathbf{r} = \mathbf{r}_0 + \lambda \mathbf{d}, \tag{1}$$

$$\mathbf{r}_0 = \begin{pmatrix} 4 \\ 2 \\ 7 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix}. \tag{2}$$

The plane has normal

$$\mathbf{n} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \quad \mathbf{r}^T \mathbf{n} = 10. \tag{3}$$

Substitute $\mathbf{r} = \mathbf{r}_0 + \lambda \mathbf{d}$ into the plane equation:

$$\mathbf{n}^T \left(\mathbf{r}_0 + \lambda \mathbf{d} \right) = 10 \tag{4}$$

$$\implies \mathbf{n}^T \mathbf{d} \, \lambda = 10 - \mathbf{n}^T \mathbf{r}_0. \tag{5}$$

Theoretical Solution

Now,

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

$$\mathbf{n}^T\mathbf{r}_0 = \begin{pmatrix} 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ 7 \end{pmatrix} = 9. \tag{7}$$

Thus,

$$\lambda = \frac{10 - 9}{1} = 1. \tag{8}$$

Hence, the intersection point is

$$P = \mathbf{r}_0 + \lambda\mathbf{d} \tag{9}$$

$$= \begin{pmatrix} 7 \\ 6 \\ 9 \end{pmatrix}. \tag{10}$$

Theoretical Solution

Given point is

$$A = \begin{pmatrix} 1 \\ -2 \\ 9 \end{pmatrix}. \tag{11}$$

The displacement vector is

$$\mathbf{v} = P - A = \begin{pmatrix} 6 \\ 8 \\ 0 \end{pmatrix}. \tag{12}$$

Therefore, the distance is

$$\begin{aligned} d = \|\mathbf{v}\| &= \sqrt{\mathbf{v}^T\mathbf{v}} && (13)\\ &= \sqrt{6^2 + 8^2 + 0^2} && (14)\\ &= \sqrt{100} = 10. && (15) \end{aligned}$$

Conclusion

Final Answer: The required distance is

$$\boxed{10}$$

C Code

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

// Function to compute distance between A(1,-2,9) and P(7,6,9)
double compute_distance() {
    double A[3] = {1, -2, 9};
    double P[3] = {7, 6, 9};
    double v[3];
    double sum = 0.0;

    for(int i=0; i<3; i++) {
        v[i] = P[i] - A[i];
        sum += v[i]*v[i];
    }

    return sqrt(sum);
}
```

Python Code using shared output

```
import ctypes

# Load the shared C library
lib = ctypes.CDLL(./points.so)

# Specify the return type of the C function
lib.compute_distance.restype = ctypes.c_double

# Call the C function
distance = lib.compute_distance()

# Store the distance in another variable
final_distance = distance

# Print the distance
print(Distance from point (1,-2,9) to intersection point (7,6,9):
    , final_distance)
```

Python Code using shared output

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

# Given point
A = np.array([1, -2, 9])

# Line: r = r0 + lambda*d
r0 = np.array([4, 2, 7])
d = np.array([3, 4, 2])
lambda_vals = np.linspace(-1, 2, 100)
line_points = r0.reshape(3,1) + d.reshape(3,1) * lambda_vals

# Intersection point
P = r0 + d

# Plane: x - y + z = 10 => z = 10 - x + y
x_plane = np.linspace(-5, 10, 20)
y_plane = np.linspace(-5, 10, 20)
```

Python Code using shared output

```
X, Y = np.meshgrid(x_plane, y_plane)
Z = 10 - X + Y

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

# Plot plane
ax.plot_surface(X, Y, Z, alpha=0.3, color='cyan', rstride=1,
    cstride=1, edgecolor='none')

# Plot line
ax.plot(line_points[0,:], line_points[1,:], line_points[2,:],
    color='blue', label=Line r=r0+d)
```

Python Code using shared output

```
# Plot points
ax.scatter(A[0], A[1], A[2], color='red', s=50, label=Point A
    (1,-2,9))
ax.scatter(P[0], P[1], P[2], color='green', s=50, label=
    Intersection P(7,6,9))

# Dotted line from A to P
ax.plot([A[0], P[0]], [A[1], P[1]], [A[2], P[2]], color='magenta'
    , linestyle='--', label=Distance d)

# Labels
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title(Distance from Point to Line-Plane Intersection)
ax.legend()
plt.show()
```

Plot

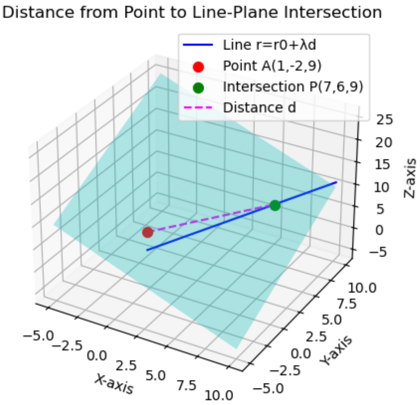

Figure: PLOT