Problem Statement

Question

Find the distance of the point

$$\mathbf{P} = \begin{bmatrix} 2 \\ 4 \\ -1 \end{bmatrix}$$

from the line

$$\frac{x+5}{1} = \frac{y+3}{4} = \frac{z-6}{-9}.$$

Line in Vector Form

$$\mathbf{r} = \mathbf{A} + \lambda \begin{pmatrix} 1 \\ 4 \\ -9 \end{pmatrix}, \qquad \mathbf{A} = \begin{pmatrix} -5 \\ -3 \\ 6 \end{pmatrix}, \ \lambda \in \mathbb{R}. \tag{1}$$

We seek the foot $\mathbf{Q} = (x, y, z)^T$ on the line such that

$(\mathbf{P} - \mathbf{Q}) \cdot \begin{pmatrix} 1 \\ 4 \\ -9 \end{pmatrix} = 0.$

Matrix Form

From the perpendicularity condition we obtain the linear system

$$\begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & 9 \\ -1 & -4 & 9 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \lambda \end{bmatrix} = \begin{bmatrix} -5 \\ -3 \\ 6 \\ -27 \end{bmatrix}. \tag{2}$$

Write the augmented matrix for elimination:

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & -1 & -5 \\ 0 & 1 & 0 & -4 & -3 \\ 0 & 0 & 1 & 9 & 6 \\ -1 & -4 & 9 & 0 & -27 \end{array}\right] \tag{3}$$

Row-Reduction — Steps

$$R_4 \leftarrow R_4 + R_1 \quad \Longrightarrow \quad \left[\begin{array}{cccc|c} 1 & 0 & 0 & -1 & -5 \\ 0 & 1 & 0 & -4 & -3 \\ 0 & 0 & 1 & 9 & 6 \\ 0 & -4 & 9 & -1 & -32 \end{array}\right] \tag{4}$$

$$R_4 \leftarrow R_4 + 4R_2 \quad \Longrightarrow \quad \left[\begin{array}{cccc|c} 1 & 0 & 0 & -1 & -5 \\ 0 & 1 & 0 & -4 & -3 \\ 0 & 0 & 1 & 9 & 6 \\ 0 & 0 & 9 & -17 & -44 \end{array}\right] \tag{5}$$

$$R_4 \leftarrow R_4 - 9R_3 \quad \Longrightarrow \quad \left[\begin{array}{cccc|c} 1 & 0 & 0 & -1 & -5 \\ 0 & 1 & 0 & -4 & -3 \\ 0 & 0 & 1 & 9 & 6 \\ 0 & 0 & 0 & -98 & -98 \end{array}\right] \tag{6}$$

Final Steps to RREF

$$R_4 \leftarrow \frac{1}{-98}R_4 \qquad \Longrightarrow \qquad \left[\begin{array}{cccc|c} 1 & 0 & 0 & -1 & -5 \\ 0 & 1 & 0 & -4 & -3 \\ 0 & 0 & 1 & 9 & 6 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] \tag{7}$$

Now eliminate the λ-entries above:

$$R_1 \leftarrow R_1 + R_4, \quad R_2 \leftarrow R_2 + 4R_4, \quad R_3 \leftarrow R_3 - 9R_4$$

which yields the RREF:

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & -4 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -3 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]. \tag{8}$$

Solution and Distance

Read off the solution from (**??**):

$$x = -4, \quad y = 1, \quad z = -3, \quad \lambda = 1, \qquad \mathbf{Q} = \begin{pmatrix} -4 \\ 1 \\ -3 \end{pmatrix}. \tag{9}$$

Compute the distance:

$$\begin{aligned} d = \|\mathbf{P} - \mathbf{Q}\| &= \sqrt{(2-(-4))^2 + (4-1)^2 + (-1-(-3))^2} \\ &= \sqrt{6^2 + 3^2 + 2^2} = \sqrt{49} = 7. \end{aligned} \tag{10}$$

$$\boxed{d = 7}$$

C Code (Part 1)

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

#define ROWS 4
#define COLS 5 // 4 variables + 1 RHS

// Function to perform Gaussian elimination to RREF
void gaussJordan(double mat[ROWS][COLS]) {
    int i, j, k;
    for (i = 0; i < ROWS; i++) {
        // Make the pivot element = 1
        double pivot = mat[i][i];
        if (pivot != 0) {
            for (j = 0; j < COLS; j++) {
                mat[i][j] /= pivot;
            }
        }
```

C Code (Part 2)

```
        // Eliminate all other entries in column i
        for (k = 0; k < ROWS; k++) {
            if (k != i) {
                double factor = mat[k][i];
                for (j = 0; j < COLS; j++) {
                    mat[k][j] -= factor * mat[i][j];
                }
            }
        }
    }
}
int main() {
    // Augmented matrix for system in (x,y,z,lambda)
    double mat[ROWS][COLS] = {
        { 1, 0, 0, -1, -5},
        { 0, 1, 0, -4, -3},
        { 0, 0, 1, 9, 6},
        {-1, -4, 9, 0, -27}
    };
```

C Code (Part 3)

```c
    gaussJordan(mat);

    double x = mat[0][4], y = mat[1][4];
    double z = mat[2][4], lambda = mat[3][4];

    printf("Solution: x = %.2f, y = %.2f, z = %.2f,  = %.2f\n",
           x, y, z, lambda);

    // Given point P(2,4,-1)
    double px = 2, py = 4, pz = -1;
    double dist = sqrt((px-x)*(px-x) + (py-y)*(py-y) + (pz-z)*(pz
        -z));

    printf("Distance = %.2f\n", dist);
    return 0;
}
```

Python Code (Part 1)

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

# Load shared library
lib = ctypes.CDLL("gj.so")

# Define function signature
lib.distance.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double
]
lib.distance.restype = ctypes.c_double

# Point P
px, py, pz = 2.0, 4.0, -1.0
```

Python Code (Part 2)

```
# Call C function
dist = lib.distance(px, py, pz)
print("Distance =", dist)

# Foot of perpendicular (from Gauss-Jordan result)
Q = np.array([-1/7, 19/7, 45/7])
P = np.array([px, py, pz])

# Line definition
A_line = np.array([-5, -3, 6])
d = np.array([1, 4, -9])

# Generate line points
t_vals = np.linspace(-2, 3, 100)
line_points = A_line.reshape(3,1) + d.reshape(3,1)*t_vals
```

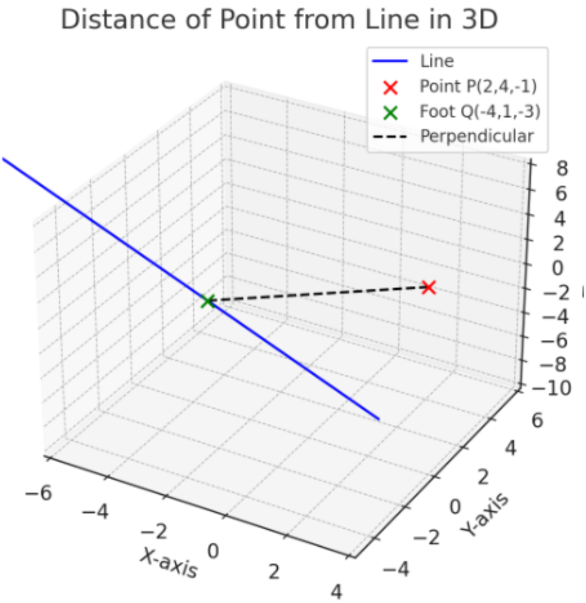
Python Code (Plot)

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

# Line, points, perpendicular
ax.plot(line_points[0], line_points[1], line_points[2],
        'b-', label="Line")
ax.scatter(P[0], P[1], P[2], color='r', s=60, label="Point P")
ax.scatter(Q[0], Q[1], Q[2], color='g', s=60, label="Foot Q")
ax.plot([P[0], Q[0]], [P[1], Q[1]], [P[2], Q[2]],
        'k--', label="Perpendicular")

ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
ax.legend()
plt.show()
```

Plot



Distance of Point from Line in 3D
Line
Point P(2,4,-1)
Foot Q(-4,1,-3)
Perpendicular
X-axis
Y-axis
−6
−4
−2
0
2
4
6
8
−10