Problem

Question

Find the distance of the point

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

from the line

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

Solution (1/3)

The distance formula using projection matrices is

$$d = \left\| \left(I - \frac{\mathbf{m}\mathbf{m}^T}{\mathbf{m}^T\mathbf{m}} \right) (\mathbf{p} - \mathbf{a}) \right\|. \tag{1}$$

Let

$$\mathbf{w} = \mathbf{p} - \mathbf{a} = \begin{bmatrix} 7 \\ 7 \\ -7 \end{bmatrix}. \tag{2}$$

Solution (2/3)

Compute

$$\mathbf{m}^T\mathbf{m} = 1^2 + 4^2 + (-9)^2 = 98, \tag{3}$$

$$\mathbf{m}^T\mathbf{w} = 1 \cdot 7 + 4 \cdot 7 + (-9)(-7) = 98. \tag{4}$$

So

$$(I - P)\mathbf{w} = \mathbf{w} - \frac{\mathbf{m}^T\mathbf{w}}{\mathbf{m}^T\mathbf{m}}\mathbf{m} = \begin{bmatrix} 7 \\ 7 \\ -7 \end{bmatrix} - \begin{bmatrix} 1 \\ 4 \\ -9 \end{bmatrix}. \tag{5}$$

Solution (3/3)

$$(I - P)\mathbf{w} = \begin{bmatrix} 6 \\ 3 \\ 2 \end{bmatrix}. \tag{6}$$

Therefore,

$$d = \sqrt{6^2 + 3^2 + 2^2} = \sqrt{49} = 7. \tag{7}$$

$$\boxed{d = 7}$$

C Code (1/3)

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

// Function to compute squared norm of a vector
double norm_sq(double v[3]) {
    return v[0]*v[0] + v[1]*v[1] + v[2]*v[2];
}

// Function to compute dot product
double dot(double v1[3], double v2[3]) {
    return v1[0]*v2[0] + v1[1]*v2[1] + v1[2]*v2[2];
}
```

C Code (2/3)

```c
int main() {
    // Point p
    double p[3] = {2, 4, -1};
    // Point a on line
    double a[3] = {-5, -3, 6};
    // Direction vector of line
    double m[3] = {1, 4, -9};

    // Compute w = p - a
    double w[3];
    for (int i = 0; i < 3; i++) {
        w[i] = p[i] - a[i];
    }
```

C Code (3/3)

```c
    // Compute distance
    double d2 = norm_sq(w) -
              (dot(m, w) * dot(m, w)) / norm_sq(m);
    double d = sqrt(d2);

    printf("Distance from point to line = %.2f\n", d);
    return 0;
}
```

Python(1/3)

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

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

# Define argument and return types
lib.distance_point_line.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)
]
lib.distance_point_line.restype = ctypes.c_double
```

Python(2/3)

```python
# Define arrays
p = np.array([2.0, 4.0, -1.0], dtype=np.double)
a = np.array([-5.0, -3.0, 6.0], dtype=np.double)
m = np.array([1.0, 4.0, -9.0], dtype=np.double)

# Call C function
d = lib.distance_point_line(
    p.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    a.ctypes.data_as(ctypes.POINTER(ctypes.c_double)),
    m.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
)

print(f"Distance from point to line = {d:.2f}")
# Projection of p onto line
w = p - a
proj = (np.dot(m, w) / np.dot(m, m)) * m
foot = a + proj
```

Python(3/3)

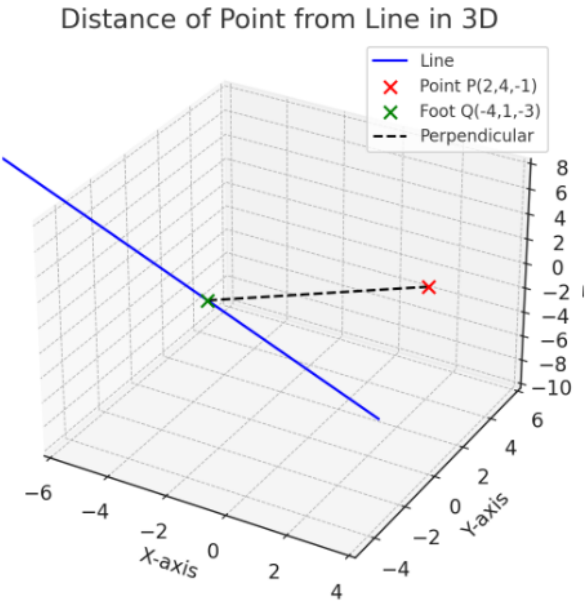
```python
# Line points
t_vals = np.linspace(-2, 2, 100)
line_points = np.array([a + t*m for t in t_vals])

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

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=50, label="Point P")
ax.plot([p[0], foot[0]], [p[1], foot[1]], [p[2], foot[2]],
       'r--', label="Perpendicular")
ax.scatter(foot[0], foot[1], foot[2],
          color='k', s=60, marker='x', label="Foot")

ax.legend()
plt.show()
```

Plot



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