Matgeo Presentation - 6.3.3

ee25btech11063 - Vejith

September 17, 2025

Question

Find the shortest distance between the lines given by
$\mathbf{r} = (8 + 3\lambda)\hat{i} - (9 + 16\lambda)\hat{j} + (10 + 7\lambda)\hat{k}$ and
$\mathbf{r} = 15\hat{i} + 29\hat{j} + 5\hat{k} + \mu(3\hat{i} + 8\hat{j} - 5\hat{k})$.

Solution

The given lines can be written in vector form as

$$\mathbf{X} = \begin{pmatrix} 8 \\ -9 \\ 10 \end{pmatrix} + k \begin{pmatrix} 3 \\ -16 \\ 7 \end{pmatrix} \tag{0.1}$$

$$\mathbf{X} = \begin{pmatrix} 15 \\ 29 \\ 5 \end{pmatrix} + k \begin{pmatrix} 3 \\ 8 \\ -5 \end{pmatrix} \tag{0.2}$$

$$\tag{0.3}$$

which are of the form

$$\mathbf{X_1} = \mathbf{A} + k_1 \mathbf{m_1} \tag{0.4}$$

$$\mathbf{X_2} = \mathbf{B} + k_2 \mathbf{m_2} \tag{0.5}$$

let $\mathbf{M} = \begin{pmatrix} \mathbf{m_1} & \mathbf{m_2} \end{pmatrix}$ and $\mathbf{K} = \begin{pmatrix} k_1 \\ -k_2 \end{pmatrix}$ be the values of k for which shortest distance between the two lines occurs

Solution

$$\implies \mathbf{M} = \begin{pmatrix} 3 & 3 \\ -16 & 8 \\ 7 & -5 \end{pmatrix} \text{ and } \mathbf{B} - \mathbf{A} = \begin{pmatrix} 7 \\ 38 \\ -5 \end{pmatrix} \quad (0.6)$$

$$(\mathbf{M} \quad \mathbf{B} - \mathbf{A}) = \begin{pmatrix} 3 & 3 & 7 \\ -16 & 8 & 38 \\ 7 & -5 & -5 \end{pmatrix} \xleftrightarrow{R_2 \to R_2 + \frac{16}{3} \times R_1} \begin{pmatrix} 3 & 3 & 7 \\ 0 & 24 & \frac{226}{3} \\ 7 & -5 & -5 \end{pmatrix} \quad (0.7)$$

$$\xleftrightarrow{R_3 \to R_3 - \frac{7}{3} \times R_1} \begin{pmatrix} 3 & 3 & 7 \\ 0 & 24 & \frac{226}{3} \\ 0 & -12 & -\frac{64}{3} \end{pmatrix} \quad (0.8)$$

$$\xleftrightarrow{R_3 \to R_3 + \frac{1}{2} \times R_2} \begin{pmatrix} 3 & 3 & 7 \\ 0 & 24 & \frac{226}{3} \\ 0 & 0 & -\frac{49}{3} \end{pmatrix} \quad (0.9)$$

The above matrix now is in row echelon form.Rank of a matix in echelon form is number of non zero rows.so,The rank of the above matrix is 3

Solution

$\implies$ given lines are skew.

$$\implies \mathbf{M}^T\mathbf{M}\mathbf{K} = \mathbf{M}^T(\mathbf{B}-\mathbf{A}) \tag{0.10}$$

$$\begin{pmatrix} 3 & -16 & 7 \\ 3 & 8 & -5 \end{pmatrix} \begin{pmatrix} 3 & 3 \\ -16 & 8 \\ 7 & -5 \end{pmatrix} \mathbf{K} = \begin{pmatrix} 3 & -16 & 7 \\ 3 & 8 & -5 \end{pmatrix} \begin{pmatrix} 7 \\ 38 \\ -5 \end{pmatrix} \tag{0.11}$$

$$\implies \begin{pmatrix} 314 & -154 \\ -154 & 98 \end{pmatrix} \mathbf{K} = \begin{pmatrix} -622 \\ 350 \end{pmatrix} \tag{0.12}$$

The augmented matrix of above equation is given by

$$\left(\begin{array}{cc|c} 314 & -154 & -622 \\ -154 & 98 & 350 \end{array}\right) \xleftrightarrow{R_1 \to R_1 + 2R_2} \left(\begin{array}{cc|c} 6 & 42 & 78 \\ -154 & 98 & 350 \end{array}\right) \tag{0.13}$$

$$\xleftrightarrow{R_2 \to R_2 + \frac{77}{3} \times R_2} \left(\begin{array}{cc|c} 6 & 42 & 78 \\ 0 & 1176 & 2352 \end{array}\right) \tag{0.14}$$

$$\xleftrightarrow{R_1 \to \frac{1}{6} \times R_1} \left(\begin{array}{cc|c} 1 & 7 & 13 \\ 0 & 1176 & 2352 \end{array}\right) \tag{0.15}$$

On back substitution we get, (0.16)

Conclusion

$$\mathbf{K} = \begin{pmatrix} k_1 \\ -k_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \tag{0.17}$$

$$\implies \mathbf{X_1} = \begin{pmatrix} 5 \\ 7 \\ 3 \end{pmatrix} \text{ and } \mathbf{X_2} = \begin{pmatrix} 9 \\ 13 \\ 15 \end{pmatrix} \tag{0.18}$$

$$\tag{0.19}$$

The minimum distance between the lines is given by

$$\|\mathbf{X_2} - \mathbf{X_1}\| = \|\begin{pmatrix} 4 \\ 6 \\ 12 \end{pmatrix}\| = 14 \tag{0.20}$$

Plot

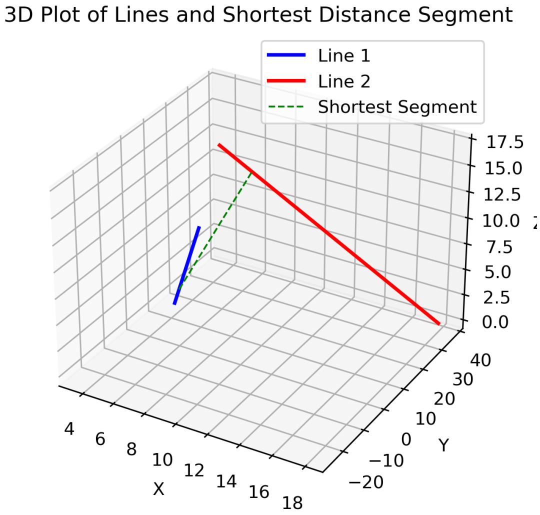

Figure: Caption

C Code: line.c

```c
#include <stdio.h>

int main() {
    #include <stdio.h>
#include <math.h>

int main() {
    // Define P1, d1, P2, d2
    double P1[3] = {8, -9, 10};
    double d1[3] = {3, -16, 7};
    double P2[3] = {15, 29, 5};
    double d2[3] = {3, 8, -5};

    // Compute c = P1 - P2
    double c[3];
    for (int i = 0; i < 3; i++) {
        c[i] = P1[i] - P2[i];
    }

    // Matrix M = [d1 -d2]
    double M[3][2] = {
        { d1[0], -d2[0] },
        { d1[1], -d2[1] },
        { d1[2], -d2[2] }
    };

    // Compute M^T * M (2x2 matrix)
    double MTM[2][2] = {0};
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            for (int k = 0; k < 3; k++) {
                MTM[i][j] += M[k][i] * M[k][j];
            }}}
```

C Code: line.c

```
    // Compute -M^T * c (2x1 vector)
    double rhs[2] = {0};
    for (int i = 0; i < 2; i++) {
        for (int k = 0; k < 3; k++) {
            rhs[i] -= M[k][i] * c[k];
        }
    }

    // Solve 2x2 linear system MTM * x = rhs
    double det = MTM[0][0]*MTM[1][1] - MTM[0][1]*MTM[1][0];
    double lambda = ( rhs[0]*MTM[1][1] - rhs[1]*MTM[0][1] ) / det;
    double mu = ( MTM[0][0]*rhs[1] - MTM[1][0]*rhs[0] ) / det;

    // Closest points Q1 and Q2
    double Q1[3], Q2[3];
    for (int i = 0; i < 3; i++) {
        Q1[i] = P1[i] + lambda * d1[i];
        Q2[i] = P2[i] + mu * d2[i];
    }
    // Distance = ||Q1 - Q2||
    double dx = Q1[0] - Q2[0];
    double dy = Q1[1] - Q2[1];
    double dz = Q1[2] - Q2[2];
    double distance = sqrt(dx*dx + dy*dy + dz*dz);

    // Write result to file "line.dat"
    FILE *fp = fopen("line.dat", "w");
    if (fp == NULL) {
        printf("Error opening file!\n");
        return 1;}
    fprintf(fp, "Shortest distance between the lines = %.2f\n", distance);
    fclose(fp);
return 0;}
```

Python: plot.py

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

# Parameter ranges
lambda_vals = np.linspace(-1.5, 1, 100)
mu_vals = np.linspace(-2.5, 1, 100)

# Line 1: r = (8 + 3) i - (9 + 16) j + (10 + 7) k
x1 = 8 + 3 * lambda_vals
y1 = -9 - 16 * lambda_vals
z1 = 10 + 7 * lambda_vals

# Line 2: r = 15i + 29j + 5k + (3i + 8j - 5k)
x2 = 15 + 3 * mu_vals
y2 = 29 + 8 * mu_vals
z2 = 5 - 5 * mu_vals

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

# Plot Line 1
ax.plot(x1, y1, z1, label='Line␣1', color='blue', linewidth=2)

# Plot Line 2
ax.plot(x2, y2, z2, label='Line␣2', color='red', linewidth=2)

# Shortest distance segment as thin dashed green line
S1 = np.array([5., 7., 3.])
S2 = np.array([9., 13., 15.])
ax.plot([S1[0], S2[0]], [S1[1], S2[1]], [S1[2], S2[2]],
       color='green', linestyle='--', linewidth=1, label='Shortest␣Segment')
```

Python: plot.py

```
# Axes labels and legend
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()
ax.set_title('3D Plot of Lines and Shortest Distance Segment')
plt.savefig('shortest_distance_3d.png', dpi=300, bbox_inches='tight')
plt.show()
```