MatGeo Assignment - Problem 1.9.13

EE25BTECH11024

IIT Hyderabad

September 18, 2025

Problem Statement

If $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are the points with position vectors $\hat{i} + \hat{j} - \hat{k}$, $2\hat{i} - \hat{j} + 3\hat{k}$, $2\hat{i} - 3\hat{k}$, $3\hat{i} - 2\hat{j} + \hat{k}$ respectively, find the projection of AB along CD.

Given Data:

Symbol	Value	Description
A	$\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$	First Point
B	$\begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}$	Second Point
C	$\begin{pmatrix} 2 \\ 0 \\ -3 \end{pmatrix}$	Third Point
D	$\begin{pmatrix} 3 \\ -2 \\ 1 \end{pmatrix}$	Fourth Point

Formula Used

The Projection of A along B is given by

$$\mathbf{C} = \frac{\mathbf{A}^\top \mathbf{B}}{\|\mathbf{B}\|^2}\mathbf{B} \tag{1}$$

Solution:

From the the given points we find AB and CD

$$\mathbf{B}-\mathbf{A}=\begin{pmatrix}1\\2\\-4\end{pmatrix},\mathbf{D}-\mathbf{C}=\begin{pmatrix}1\\2\\-4\end{pmatrix} \tag{2}$$

Let P be the projection of AB along CD. We know that

$$\mathbf{P}=(\frac{(\mathbf{B}-\mathbf{A})^{\top}(\mathbf{D}-\mathbf{C})}{\|\mathbf{D}-\mathbf{C}\|^{2}})(\mathbf{D}-\mathbf{C}) \tag{3}$$

Substituting (**??**) in (**??**) we get

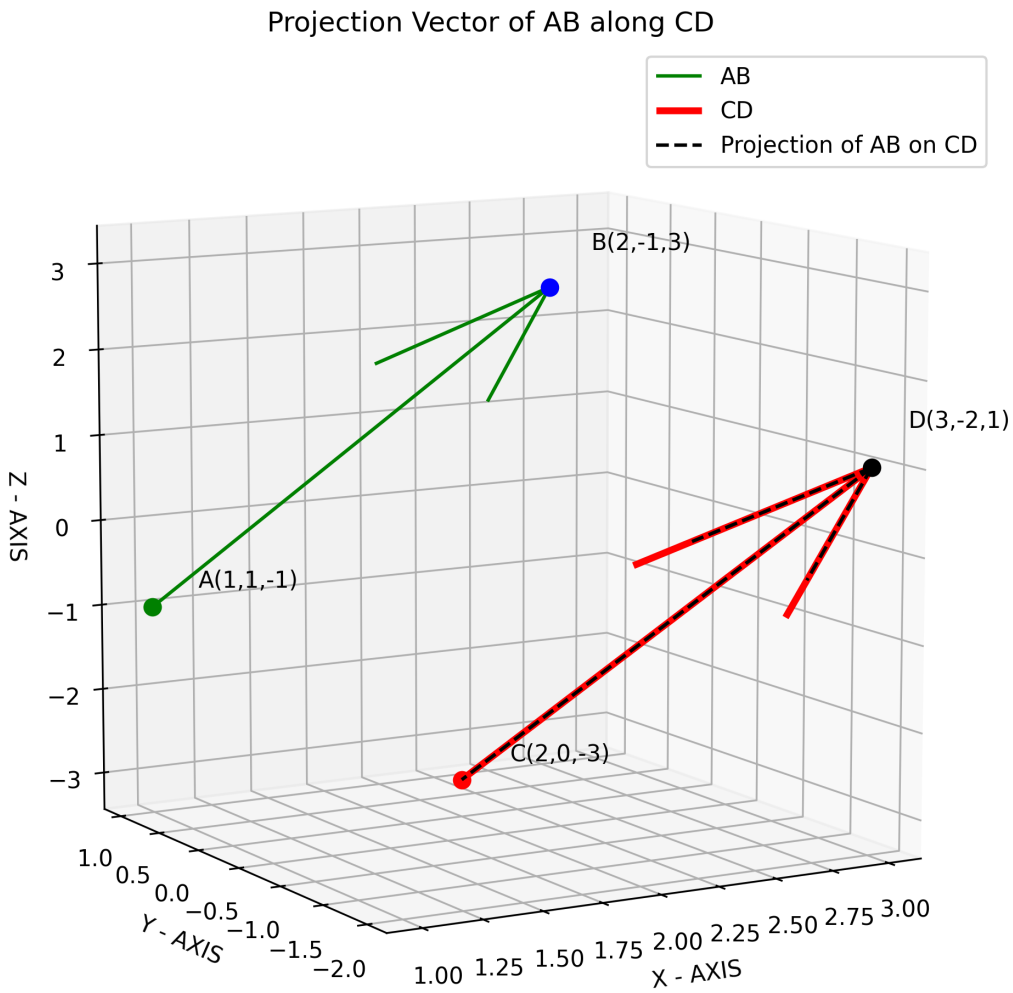
$$P=(\mathbf{D}-\mathbf{C})=\begin{pmatrix}1\\2\\-4\end{pmatrix} \tag{4}$$

Final Answer

Thus, the projection of AB along CD $= \begin{pmatrix} 1 \\ 2 \\ -4 \end{pmatrix}$

See Figure **??**.

Figure


Projection Vector of AB along CD
AB
CD
Projection of AB on CD
B(2,-1,3)
D(3,-2,1)
A(1,1,-1)
C(2,0,-3)
Z - AXIS
Y - AXIS
X - AXIS

Python Code: plot.py (Native)

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

A = np.array([1,1,-1])
B = np.array([2,-1,3])
C = np.array([2,0,-3])
D = np.array([3,-2,1])

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

ax.quiver(A[0], A[1], A[2], B[0]-A[0], B[1]-A[1], B[2]-A[2], color = 'g'
    , label = 'AB')
ax.quiver(C[0], C[1], C[2], D[0]-C[0], D[1]-C[1], D[2]-C[2], color = 'r'
    , label = 'CD')
```

Python Code (Native Implementation – plot.py)

```
ax.scatter(*A, color='green', s=50)
ax.scatter(*B, color= 'blue', s=50)
ax.scatter(*C, color= 'red', s=50)
ax.scatter(*D, color= 'black', s=50)

ax.text(A[0]+0.1, A[1]-0.3, A[2]+0.3, "A(1,1,-1)")
ax.text(B[0]+0.3, B[1]+0.3, B[2]+0.3, "B(2,-1,3)")
ax.text(C[0]+0.1, C[1]-0.3, C[2]+0.3, "C(2,0,-3)")
ax.text(D[0]+0.3, D[1]+0.3, D[2]+0.3, "D(3,-2,1)")


ax.set_xlabel("X - AXIS")
ax.set_ylabel("Y - AXIS")
ax.set_zlabel("Z - AXIS")
ax.set_title("Vectors AB and CD in 3D Space")
```

Python Code (Native Implementation – plot.py)

```
ax.legend()
ax.set_box_aspect([1, 1, 1])
ax.view_init(elev=10, azim=-120)
plt.savefig("fig3.png", dpi=300)
plt.show()
```

C Code (Shared Library – findprojection.c)

```c
#include <stdio.h>

void find_projection(double *A, double *B, double *C, double *D, double
    *projection){

double AB[3];
for (int i = 0; i < 3; i++) {
    AB[i] = B[i] - A[i];
}
double CD[3];
for (int i=0; i<3; i++){
    CD[i] = D[i] - C[i];
}

double dot_product = 0.0;
for (int i=0; i<3; i++) {
    dot_product += AB[i]*CD[i];
}
```

C Code (Shared Library – findprojection.c)

```
double mag_square = 0.0;
for (int i=0; i<3; i++) {
mag_square += CD[i]*CD[i];
}

if (mag_square == 0) {
        projection[0] = 0;
        projection[1] = 0;
        projection[2] = 0;
        return;
    }
for (int i=0; i<3; i++){
    projection[i] = (dot_product/mag_square)*CD[i];
}

}
```

Python Code: call.py (C + Python)

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

so = ctypes.CDLL("./find_projection.so")

so.find_projection.argtypes = [ctypes.POINTER(ctypes.c_double), ctypes.
    POINTER(ctypes.c_double), ctypes.POINTER(ctypes.c_double), ctypes.
    POINTER(ctypes.c_double), ctypes.POINTER(ctypes.c_double)]
so.find_projection.restype = None

A = np.array([1, 1, -1], dtype=np.double)
B = np.array([2, -1, 3], dtype=np.double)
C = np.array([2, 0, -3], dtype=np.double)
D = np.array([3, -2, 1], dtype=np.double)
```

Python Code (C Integrated – call.py)

```
A_ptr = A.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
B_ptr = B.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
C_ptr = C.ctypes.data_as(ctypes.POINTER(ctypes.c_double))
D_ptr = D.ctypes.data_as(ctypes.POINTER(ctypes.c_double))

proj_result = (ctypes.c_double*3)()

so.find_projection(A_ptr, B_ptr, C_ptr, D_ptr, proj_result)
proj_vec = np.array(proj_result)

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

ax.quiver(A[0], A[1], A[2], B[0]-A[0], B[1]-A[1], B[2]-A[2], color = 'g'
    , label = 'AB')
ax.quiver(C[0], C[1], C[2], D[0]-C[0], D[1]-C[1], D[2]-C[2], color = 'r'
    , label = 'CD', linewidth=3, arrow_length_ratio=0.4)
ax.quiver(C[0], C[1], C[2], proj_vec[0], proj_vec[1], proj_vec[2], color
    ='black', linestyle = '--', label='Projection of AB on CD')
```

Python Code (C Integrated – call.py)

```
ax.scatter(*A, color='green', s=50)
ax.scatter(*B, color= 'blue', s=50)
ax.scatter(*C, color= 'red', s=50)
ax.scatter(*D, color= 'black', s=50)

ax.text(A[0]+0.1, A[1]-0.3, A[2]+0.3, "A(1,1,-1)")
ax.text(B[0]+0.3, B[1]+0.3, B[2]+0.3, "B(2,-1,3)")
ax.text(C[0]+0.1, C[1]-0.3, C[2]+0.3, "C(2,0,-3)")
ax.text(D[0]+0.3, D[1]+0.3, D[2]+0.3, "D(3,-2,1)")
```

Python Code (C Integrated – call.py)

```
ax.set_xlabel("X - AXIS")
ax.set_ylabel("Y - AXIS")
ax.set_zlabel("Z - AXIS")
ax.set_title("Projection Vector of AB along CD")

ax.legend()
ax.set_box_aspect([1, 1, 1])
ax.view_init(elev=10, azim=-120)
plt.savefig("fig3'.png", dpi=300)
plt.show()
```