

# Problem 12.765

ee25btech11023-Venkata Sai

October 10, 2025

- 1 Problem
- 2 Solution
  - Formula
  - Conclusion
- 3 Plot
- 4 C code
- 5 Python code

## Problem

Let  $\mathbf{v}_1 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$  and  $\mathbf{v}_2 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$  be two vectors. The value of the coefficient  $\alpha$  in the expression  $\mathbf{v}_1 = \alpha \mathbf{v}_2 + \mathbf{e}$ , which minimizes the length of the error vector  $\mathbf{e}$ , is

# Formula

Given expression

$$\mathbf{v}_1 = \alpha \mathbf{v}_2 + \mathbf{e} \quad (3.1)$$

where  $\mathbf{e}$  is the error vector

For any linear system  $\mathbf{Ax} = \mathbf{B}$ , the least squares solution formula is given by

$$(\mathbf{A}^\top \mathbf{A}) \mathbf{x} = \mathbf{A}^\top \mathbf{B} \quad (3.2)$$

$$\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{B} \quad (3.3)$$

On writing the given expression as a linear system

$$\mathbf{v}_2 \alpha = \mathbf{v}_1 \quad (3.4)$$

where  $\alpha$  being an  $1 \times 1$  vector

# Conclusion

$$\mathbf{A} = \mathbf{v}_2, \mathbf{B} = \mathbf{v}_1 \quad (3.5)$$

$$\alpha = \left( \mathbf{v}_2^\top \mathbf{v}_2 \right)^{-1} \mathbf{v}_2^\top \mathbf{v}_1 \quad (3.6)$$

$$= \left( \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}^\top \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} \right)^{-1} \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}^\top \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \quad (3.7)$$

$$= \left( \begin{pmatrix} 2 & 1 & 3 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} \right)^{-1} \begin{pmatrix} 2 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \quad (3.8)$$

$$= (4 + 1 + 9)^{-1} (2 + 2 + 0) \quad (3.9)$$

$$= \frac{1}{14} (4) \quad (3.10)$$

$$= \frac{2}{7} \quad (3.11)$$

# Plot

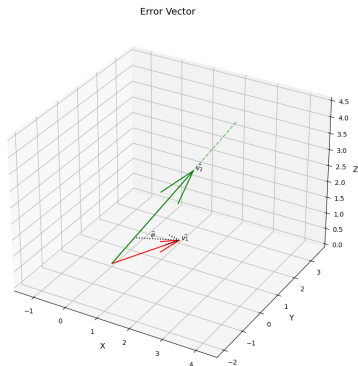


Figure:

## C code

```
void get_vectors(double* data) {  
    data[0] = 1.0;  
    data[1] = 2.0;  
    data[2] = 0.0;  
    data[3] = 2.0;  
    data[4] = 1.0;  
    data[5] = 3.0;  
}
```

# Python code for calling

```
import ctypes
import numpy as np

def solve_least_squares():
    lib = ctypes.CDLL('./code.so')

    out_data = (ctypes.c_double * 6)()
    lib.get_vectors.argtypes = [ctypes.POINTER(ctypes.c_double)]
    lib.get_vectors(out_data)
    data = np.array(list(out_data))
    v1 = data[0:3]
    v2 = data[3:6]
    v2_dot_v2 = np.dot(v2, v2)
    v2_dot_v1 = np.dot(v2, v1)
    alpha = v2_dot_v1 / v2_dot_v2
    error_vec = v1 - (alpha * v2)
    return v1, v2, error_vec, alpha
```



# Python code for plotting

```
import matplotlib.pyplot as plt
import numpy as np
from call import solve_least_squares

v1, v2, e, alpha = solve_least_squares()

fig = plt.figure(figsize=(9, 9))
ax = fig.add_subplot(111, projection='3d')

ax.quiver(0, 0, 0, v1[0], v1[1], v1[2], color='r')
ax.text(v1[0], v1[1], v1[2], '  $\vec{v_1}$  ')

line_v2 = np.array([np.zeros(3), 1.5 * v2])
ax.plot(line_v2[:, 0], line_v2[:, 1], line_v2[:, 2], 'g--', alpha
        =0.5)
ax.quiver(0, 0, 0, v2[0], v2[1], v2[2], color='g')
ax.text(v2[0], v2[1], v2[2], '  $\vec{v_2}$  ')
projection_point = alpha * v2
```

## Python code for plotting

```
ax.quiver(projection_point[0], projection_point[1],
          projection_point[2],
          e[0], e[1], e[2],
          color='k', linestyle=':')
ax.text(projection_point[0]+0.8, projection_point[1]-0.5,
        projection_point[2]+0.5, '$\\vec{e}$')
ax.set_title('Error Vector', fontsize=14)
ax.set_xlabel('X', fontsize=12); ax.set_ylabel('Y', fontsize=12);
    ax.set_zlabel('Z', fontsize=12)
ax.legend()
ax.grid(True)
ax.axis('equal')
plt.show()
```