2.4.20

Vishwambhar - EE25BTECH11025

31st August, 2025

Question

Find the value of λ such that the vectors $\mathbf{a} = 2\mathbf{i} + \lambda\mathbf{j} + \mathbf{k}$ and $\mathbf{b} = \mathbf{i} + 2\mathbf{j} + 3\mathbf{k}$ are orthogonal.

Given Vectors

$$\mathbf{a} = \begin{pmatrix} 2 \\ \lambda \\ 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \tag{1}$$

Finding Lambda(λ)

For two vectors to be orthogonal their dot product should be equal to zero which is equal to product of transpose of column matrix **a** and column matrix **b**:

$$\mathbf{a}^T\mathbf{b} = 0 \tag{2}$$

$$\begin{pmatrix} 2 & \lambda & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 0 \tag{3}$$

$$2 + 2\lambda + 3 = 0 \tag{4}$$

$$\lambda = \begin{pmatrix} \frac{-5}{2} \end{pmatrix} \tag{5}$$

Final vectors

Therefore, the final vectors are:

$$\mathbf{a} = \begin{pmatrix} 2 \\ \left(\frac{-5}{2}\right) \\ 1 \end{pmatrix} \tag{6}$$

$$\mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \tag{7}$$

C Code

```c
#include <stdio.h>

// Function to calculate dot product of two vectors
double dot_product(double a[], double b[], int size) {
    double result = 0.0;
    for(int i = 0; i < size; i++) {
        result += a[i] * b[i];
    }
    return result;
}
```

C Code

```c
// Wrapper for Python to access
double solve_lambda(double b[]) {
    // a = [2, lambda, 1]
    double a[3];
    double lambda;
    // Equation: 2*b[0] + lambda*b[1] + 1*b[2] = 0
    // => lambda = -(2*b[0] + 1*b[2]) / b[1]
    lambda = -(2*b[0] + 1*b[2]) / b[1];
    return lambda;
}
```

Python Code 1

```
import ctypes
import numpy as np

# Load the shared object file
lib = ctypes.CDLL('./problem.so')

# Set argument types for solve_lambda
lib.solve_lambda.argtypes = [ctypes.POINTER(ctypes.c_double)]
```

Python Code 1

```
# Set return type
lib.solve_lambda.restype = ctypes.c_double

b = np.array([1.0, 2.0, 3.0], dtype=np.double)

b_ptr = b.ctypes.data_as(ctypes.POINTER(ctypes.c_double))

lambda_val = lib.solve_lambda(b_ptr)
print(f"Solved lambda: {lambda_val}")
```

Python Code 2

```
import sys
sys.path.insert(0, '/home/ganachari-vishwmabhar/Downloads/codes/
    CoordGeo')
import numpy as np
import matplotlib.pyplot as plt

from line.funcs import *
from triangle.funcs import *

# Load lambda from previous result
lambda_val = -2.5
a = np.array([2, lambda_val, 1]) # vector a
b = np.array([1, 2, 3]) # vector b
```

Python Code 2

```
# Prepare 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Plot vectors from origin
ax.quiver(0, 0, 0, a[0], a[1], a[2], color='r', label='Vector a')
ax.quiver(0, 0, 0, b[0], b[1], b[2], color='b', label='Vector b')

# Mark points
ax.text(a[0], a[1], a[2], 'a', fontsize=12)
ax.text(b[0], b[1], b[2], 'b', fontsize=12)
```

Python Code 2

```
# Labels
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title("Orthogonal Vectors a and b")
ax.legend()
plt.savefig("../figs/plot.png")
plt.show()
```

Plot

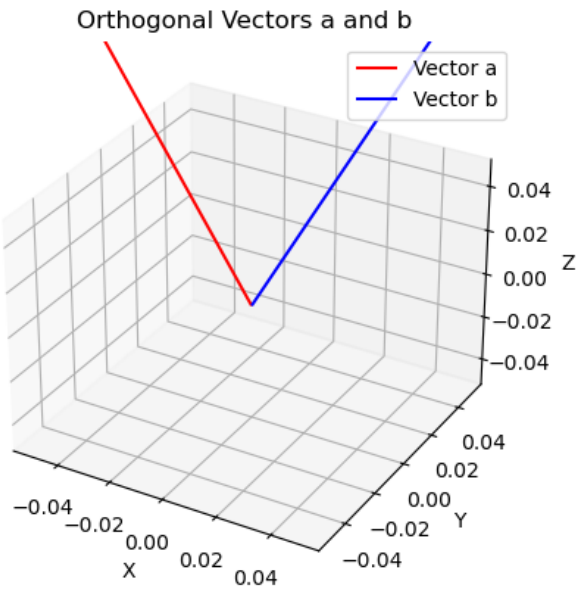

Figure: Plot of orthogonal vectors **a** and **b**.