1.10.2

RAVULA SHASHANK REDDY - EE25BTECH11047

September 12, 2025

Question

Find the unit vector in the direction of the sum of the vectors
$\mathbf{a} = 2\hat{i} - \hat{j} + \hat{k}$, $\mathbf{b} = 2\hat{j} + \hat{k}$.

Equation

The formula for unit vector of x is :

$$\frac{\mathbf{x}}{\|\mathbf{x}\|}$$

Theoretical Solution

Given :

$$\mathbf{a} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \tag{1}$$

$$\mathbf{b} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}. \tag{2}$$

Sum of the vectors:

$$\mathbf{a} + \mathbf{b} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}. \tag{3}$$

Norm of $\mathbf{a} + \mathbf{b}$:

$$\|\mathbf{a} + \mathbf{b}\| = \sqrt{\begin{pmatrix} 2 & 1 & 2 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}} = \sqrt{2 \cdot 2 + 1 \cdot 1 + 2 \cdot 2} = \sqrt{9} = 3. \tag{4}$$

Using the unit vector formula :

$$\mathbf{u} = \frac{\mathbf{a} + \mathbf{b}}{\|\mathbf{a} + \mathbf{b}\|} \tag{5}$$

$$\mathbf{u} = \frac{1}{3}\begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}. \tag{6}$$

$$\therefore \mathbf{u} = \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \\ \frac{2}{3} \end{pmatrix} \tag{7}$$

C Code - Unit vector function

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

// Function to compute unit vector of (x1, x2, x3)
void unitVector(double x1, double x2, double x3, double unit[3])
    {
    double mag = sqrt(x1*x1 + x2*x2 + x3*x3); // |x|
    unit[0] = x1 / mag;
    unit[1] = x2 / mag;
    unit[2] = x3 / mag;

   }
```

Python Code through shared output

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

# Load the shared C library
lib = ctypes.CDLL(./libunit.so)

# Define the function signature for unitVector
lib.unitVector.argtypes = [ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double)]
lib.unitVector.restype = None

# Input vectors
a = np.array([2, -1, 1], dtype=np.double)
b = np.array([0, 2, 1], dtype=np.double)
s = a + b
```

```
# Prepare C arrays
x = (ctypes.c_double * 3)(*s)
result = (ctypes.c_double * 3)()

# Call the C function
lib.unitVector(x, result)

# Convert result to numpy array
u = np.array([result[i] for i in range(3)])

# Print result
print(fUnit vector = ({u[0]:.4f})i + ({u[1]:.4f})j + ({u[2]:.4f})k)

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

origin = [0, 0, 0]
```

```
# Plot vectors
ax.quiver(*origin, *a, color='r', label='a')
ax.quiver(*origin, *b, color='g', label='b')
ax.quiver(*origin, *s, color='b', label='a+b')
ax.quiver(*origin, *u, color='m', label='unit vector (from C)',
    linewidth=2)

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()

plt.show()
```

Python code Direct

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

# Define vectors
a = np.array([2, -1, 1])
b = np.array([0, 2, 1])
result = a + b # a + b
unit_result = result / np.linalg.norm(result) # unit vector

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

Python code Direct

```
# Plot the resultant vector
ax.quiver(0, 0, 0, result[0], result[1], result[2], color='b',
    label='a+b', arrow_length_ratio=0.1)

# Plot the unit vector (scaled to length 1)
ax.quiver(0, 0, 0, unit_result[0], unit_result[1], unit_result
    [2], color='r', label='Unit vector', arrow_length_ratio=0.1)

# Labels
ax.set_xlim([0, 3])
ax.set_ylim([0, 3])
ax.set_zlim([0, 3])
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()

plt.show()
```

Plot