1.11.12

Dhanush Kumar A - AI25BTECH11010

September 9, 2025

Question

If the sum of two unit vectors is a unit vector, prove that the magnitude of their difference is $\sqrt{3}$.

Solution

Let $\mathbf{u}$ and $\mathbf{v}$ be unit vectors.
Hence,

$$\mathbf{u}^T\mathbf{u} = 1, \qquad \mathbf{v}^T\mathbf{v} = 1. \tag{1}$$

$$\begin{aligned} ||\mathbf{u}+\mathbf{v}||^2 &= (\mathbf{u}+\mathbf{v})^T(\mathbf{u}+\mathbf{v}) && (2)\\ &= \mathbf{u}^T\mathbf{u} + \mathbf{v}^T\mathbf{v} + 2\mathbf{u}^T\mathbf{v} && (3)\\ 1^2 &= 1 + 1 + 2\mathbf{u}^T\mathbf{v} && (4)\\ 1 &= 2 + 2\mathbf{u}^T\mathbf{v} && (5)\\ \Rightarrow \quad \mathbf{u}^T\mathbf{v} &= -\tfrac{1}{2}. && (6) \end{aligned}$$

Solution

Now,

$$\begin{aligned} \|\mathbf{u}-\mathbf{v}\|^2 &= (\mathbf{u}-\mathbf{v})^T(\mathbf{u}-\mathbf{v}) && (7)\\ &= \mathbf{u}^T\mathbf{u}+\mathbf{v}^T\mathbf{v}-2\mathbf{u}^T\mathbf{v} && (8)\\ &= 1+1-2\left(-\tfrac{1}{2}\right) && (9)\\ &= 2+1=3. && (10) \end{aligned}$$

Therefore,

$$\|\mathbf{u}-\mathbf{v}\| = \sqrt{3}. \quad (11)$$

Python code - Verfiying

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

def check_unit_sum_and_plot(a, b):
    # Ensure vectors are unit vectors
    a = a / np.linalg.norm(a)
    b = b / np.linalg.norm(b)

    # Compute vectors
    sum_vec = a + b
    diff_vec = a - b
    neg_b = -b

    # Magnitudes
    sum_mag = np.linalg.norm(sum_vec)
    diff_mag = np.linalg.norm(diff_vec)
    neg_b_mag = np.linalg.norm(neg_b) # should be 1 since b is
        unit
```

Python code - Verfiying

```
    print(f"|a+b| = {sum_mag:.3f}")
    print(f"|a-b| = {diff_mag:.3f}")
    print(f"|b| = {neg_b_mag:.3f}")
```

Python code - Plotting the Vectors

```
# Plot
    fig, ax = plt.subplots(figsize=(6,6))

    # Function to plot vectors
    def plot_vec(v, color, label):
        ax.arrow(0, 0, v[0], v[1], head_width=0.1, head_length
            =0.1,
                 fc=color, ec=color, length_includes_head=True)
        ax.text(v[0]*1.1, v[1]*1.1, f"{label}\n|{label}|={np.
            linalg.norm(v):.2f}",
                color=color, fontsize=10, ha="center")

    # Plot all vectors
    plot_vec(a, "blue", "a")
    plot_vec(b, "green", "b")
    plot_vec(neg_b, "orange", "-b")
    plot_vec(sum_vec, "red", "a+b")
    plot_vec(diff_vec, "purple", "a-b")
```

Python code - Plotting the Vectors

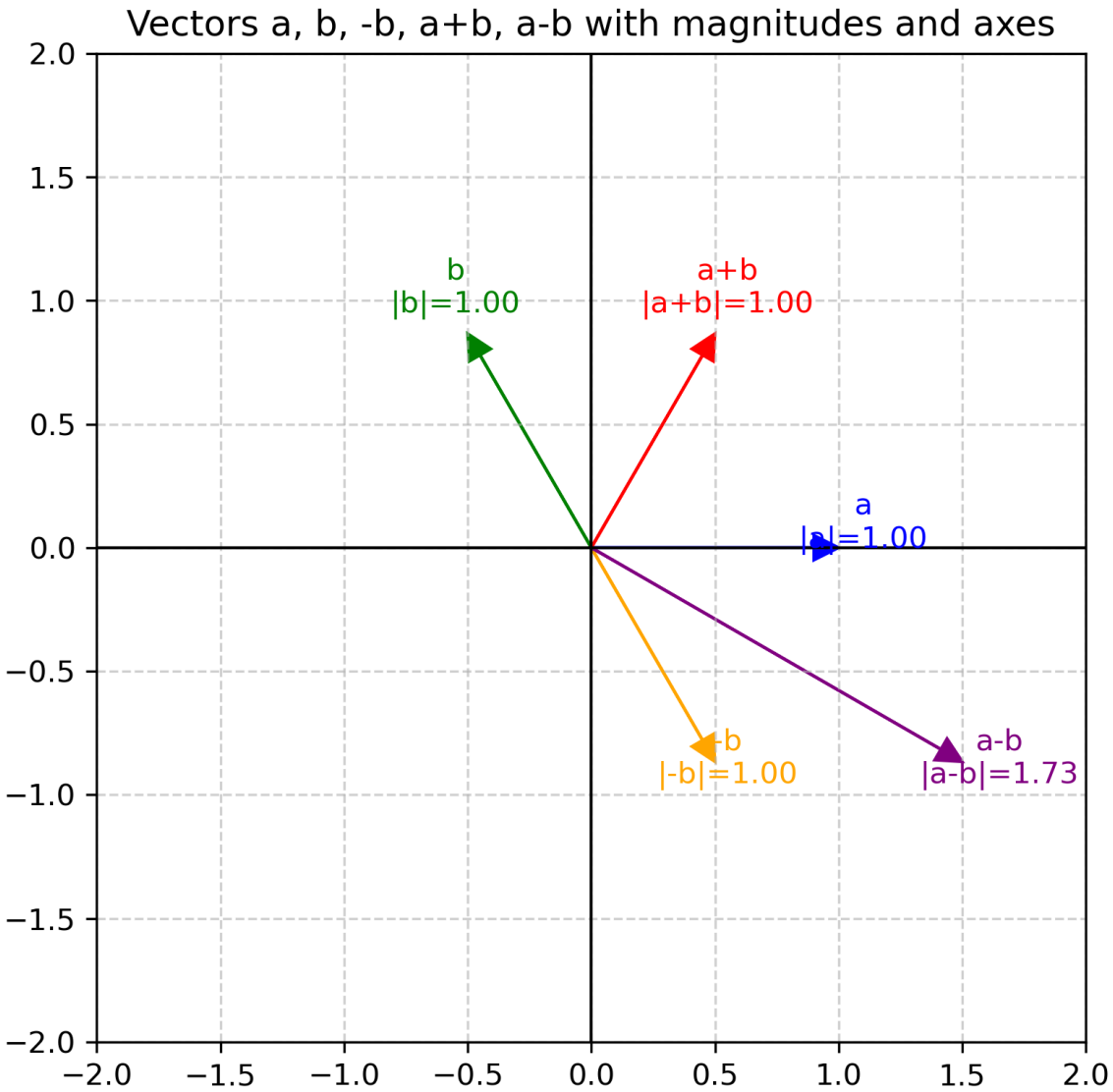
```
    # Plot X and Y axes
    ax.axhline(0, color="black", linewidth=1) # X-axis
    ax.axvline(0, color="black", linewidth=1) # Y-axis

    ax.set_xlim(-2, 2)
    ax.set_ylim(-2, 2)
    ax.set_aspect("equal")
    ax.grid(True, linestyle="--", alpha=0.6)
    ax.set_title("Vectors a, b, -b, a+b, a-b with magnitudes and
        axes")

    # Save figure
    plt.savefig('../figs/vectors_plot.png', dpi=300, bbox_inches=
        "tight")
    plt.show()

# Example: pick a along x-axis, b rotated by 120
theta = 120 * np.pi / 180
```

Plot-Using Python



C code - To Verify and Save vectors

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

// Structure for 2D vector
typedef struct {
    double x;
    double y;
} Vector;

// Function to normalize a vector
Vector normalize(Vector v) {
    double mag = sqrt(v.x * v.x + v.y * v.y);
    Vector result = {v.x / mag, v.y / mag};
    return result;
}
```

C code - To Verify and Save Vectors

```c
// Function to add two vectors
Vector add(Vector a, Vector b) {
    Vector result = {a.x + b.x, a.y + b.y};
    return result;
}

// Function to subtract two vectors
Vector subtract(Vector a, Vector b) {
    Vector result = {a.x - b.x, a.y - b.y};
    return result;
}

// Function to negate a vector
Vector negate(Vector v) {
    Vector result = {-v.x, -v.y};
    return result;
}
```

C code - To Verify and Save Vectors

```c
// Function to compute magnitude
double magnitude(Vector v) {
    return sqrt(v.x * v.x + v.y * v.y);
}

// Function to save vectors to .dat file
void save_to_dat(Vector a, Vector b, Vector neg_b, Vector sum,
    Vector diff, const char *filename) {
    FILE *fp = fopen(filename, "w");
    if (fp == NULL) {
        printf("Error opening file!\n");
        return;
    }
    fprintf(fp, "Vector\tX\tY\tMagnitude\n");
    fprintf(fp, "a\t%.4f\t%.4f\t%.4f\n", a.x, a.y, magnitude(a));
    fprintf(fp, "b\t%.4f\t%.4f\t%.4f\n", b.x, b.y, magnitude(b));
    fprintf(fp, "-b\t%.4f\t%.4f\t%.4f\n", neg_b.x, neg_b.y,
```

C code - To Verify and Save Vectors

```
    fprintf(fp, "a+b\t%.4f\t%.4f\t%.4f\n", sum.x, sum.y,
        magnitude(sum));
    fprintf(fp, "a-b\t%.4f\t%.4f\t%.4f\n", diff.x, diff.y,
        magnitude(diff));
    fclose(fp);
    printf("Data saved to %s\n", filename);
}

int main() {
    // Define vectors: a = (1,0), b = (cos120, sin120)
    Vector a = {1.0, 0.0};
    Vector b = {cos(M_PI * 120.0 / 180.0), sin(M_PI * 120.0 /
        180.0)};
```

C code - To Verify and Save Vectors

```
// Normalize to ensure unit vectors
a = normalize(a);
b = normalize(b);

// Compute required vectors
Vector sum = add(a, b);
Vector diff = subtract(a, b);
Vector neg_b = negate(b);

// Print to console
printf("|a+b| = %.3f\n", magnitude(sum));
printf("|a-b| = %.3f\n", magnitude(diff));

printf("|-b| = %.3f\n", magnitude(neg_b));
```

C code - To Verify and Save Vectors

```
    // Save results in same folder as code
    save_to_dat(a, b, neg_b, sum, diff, "vectors_data.dat");

    return 0;
}
```

Python code -Ploting the Vector using c function

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

# Step 1: Compile the C program
os.system("gcc c.c -o vectors -lm")

# Step 2: Run the compiled C program
os.system("./vectors")

# Step 3: Load data (skip header row, read mixed types)
data = np.genfromtxt("vectors_data.dat", skip_header=1, dtype=
    None, encoding="utf-8")

# Separate columns
labels = [row[0] for row in data] # first column is text
x_vals = np.array([float(row[1]) for row in data])
y_vals = np.array([float(row[2]) for row in data])
mags = np.array([float(row[3]) for row in data])
```

Python code -Ploting the Vector using c function

```
# Step 4: Plot vectors
fig, ax = plt.subplots(figsize=(6,6))

for label, x, y, mag in zip(labels, x_vals, y_vals, mags):
    ax.arrow(0, 0, x, y, head_width=0.1, head_length=0.1,
             fc="blue", ec="blue", alpha=0.7, length_includes_head
                 =True)
    ax.text(x*1.1, y*1.1, f"{label}\n|{label}|={mag:.2f}",
        fontsize=10, ha="center")

# Plot x and y axes
ax.axhline(0, color="black", linewidth=1.0, linestyle="--") # X-
    axis
ax.axvline(0, color="black", linewidth=1.0, linestyle="--") # Y-
    axis
```

Python code -Ploting the Vector using c function

```
# Formatting
ax.set_xlim(-2, 2)
ax.set_ylim(-2, 2)
ax.set_aspect("equal")
ax.grid(True)
ax.set_title("Vectors from C Program with X and Y Axes")

# Save and show plot
plt.savefig("../figs/vectors_from_c.png", dpi=300, bbox_inches="
    tight")
plt.show()
```

Plot-Using Python and C