2.5.20

Rushil Shanmukha Srinivas
EE25BTECH11057
Electrical Engginneering ,
IIT Hyderabad.

September 2, 2025

Problem Statement

Let $\mathbf{a}, \mathbf{b}, \mathbf{c}$ be three vectors with $\|\mathbf{a}\| = 1,\ \|\mathbf{b}\| = 2,\ \|\mathbf{c}\| = 3$. If the projection of $\mathbf{b}$ on $\mathbf{a}$ equals the projection of $\mathbf{c}$ on $\mathbf{a}$, and $\mathbf{b} \perp \mathbf{c}$, find $\left\|3\mathbf{a} - 2\mathbf{b} + 2\mathbf{c}\right\|$.

Dot Product of vectors

Let us denote the scalar products

$$x = \mathbf{a} \cdot \mathbf{b}, \qquad y = \mathbf{a} \cdot \mathbf{c}, \qquad z = \mathbf{b} \cdot \mathbf{c}. \tag{3.1}$$

Given: projection of $\mathbf{b}$ on $\mathbf{a}$ equals projection of $\mathbf{c}$ on $\mathbf{a}$. Since $\|\mathbf{a}\| = 1$, this implies

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a} \cdot \mathbf{c} \quad \Rightarrow \quad x = y. \tag{3.2}$$

Also $\mathbf{b} \perp \mathbf{c} \Rightarrow z = 0$. Using the magnitudes,

$$\mathbf{a} \cdot \mathbf{a} = 1, \quad \mathbf{b} \cdot \mathbf{b} = 4, \quad \mathbf{c} \cdot \mathbf{c} = 9. \tag{3.3}$$

Gram Matrix

Form the Gram (inner-product) matrix of $(\mathbf{a}, \mathbf{b}, \mathbf{c})$:

$$G = \begin{pmatrix} \mathbf{a}\cdot\mathbf{a} & \mathbf{a}\cdot\mathbf{b} & \mathbf{a}\cdot\mathbf{c} \\ \mathbf{b}\cdot\mathbf{a} & \mathbf{b}\cdot\mathbf{b} & \mathbf{b}\cdot\mathbf{c} \\ \mathbf{c}\cdot\mathbf{a} & \mathbf{c}\cdot\mathbf{b} & \mathbf{c}\cdot\mathbf{c} \end{pmatrix} = \begin{pmatrix} 1 & x & x \\ x & 4 & 0 \\ x & 0 & 9 \end{pmatrix}, \tag{3.4}$$

where we used $x = y$ and $z = 0$.
Now denote the coefficient vector of $3\mathbf{a} - 2\mathbf{b} + 2\mathbf{c}$ relative to $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ by

$$\mathbf{u} = \begin{pmatrix} 3 \\ -2 \\ 2 \end{pmatrix}. \tag{3.5}$$

Then the squared norm is the quadratic form

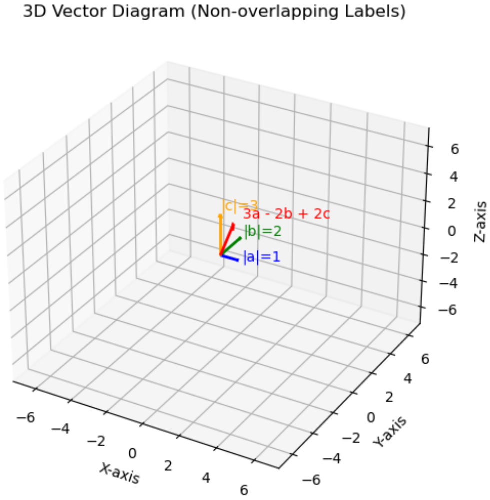
$$\left\|3\mathbf{a} - 2\mathbf{b} + 2\mathbf{c}\right\|^2 = \mathbf{u}^\top G\,\mathbf{u}. \tag{3.6}$$

$$\mathbf{u}^\top G\,\mathbf{u} = \begin{pmatrix}3 & -2 & 2\end{pmatrix}\begin{pmatrix}1 & x & x\\ x & 4 & 0\\ x & 0 & 9\end{pmatrix}\begin{pmatrix}3\\ -2\\ 2\end{pmatrix} = 9 - 6x + 16 + 6x + 36 = 61 \tag{3.7}$$

Therefore

$$\left\|3\mathbf{a} - 2\mathbf{b} + 2\mathbf{c}\right\|^2 = 61 \quad \Longrightarrow \quad \boxed{\left\|3\mathbf{a} - 2\mathbf{b} + 2\mathbf{c}\right\| = \sqrt{61}}. \tag{3.8}$$

Plots



3D Vector Diagram (Non-overlapping Labels)
|c|=3
3a - 2b + 2c
|b|=2
|a|=1
X-axis
Y-axis
Z-axis

C Code

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

// Function to compute magnitude of (3a - 2b + 2c)
double compute_magnitude() {
    // Given norms
    double norm_a = 1.0;
    double norm_b = 2.0;
    double norm_c = 3.0;


    double result_squared =
        9 * (norm_a * norm_a) + // 9|a|^2
        4 * (norm_b * norm_b) + // 4|b|^2
```

```
        4 * (norm_c * norm_c) + // 4|c|^2
        -12 * (0) + 12 * (0) + -8 * (0);

    // Simplified: 9*1 + 4*4 + 4*9 = 61
    return sqrt(result_squared);
}

int main() {
    double magnitude = compute_magnitude();
    printf("The magnitude of (3a - 2b + 2c) is: %.5f\n", magnitude);
    return 0;
}
```

Python : call_c.py

```
import ctypes
import os
import math
# Load the shared library (ensure libvector.so is in the same directory)
lib_path = os.path.abspath("./libvector.so")
lib = ctypes.CDLL(lib_path)
# Tell Python the return type of the C function
lib.compute_magnitude.restype = ctypes.c_double
# Call the function from the shared object
result = lib.compute_magnitude()
# Print result
print("The magnitude of (3a - 2b + 2c) is:", result)

# (Optional) Verify using Python math
print("Verification using Python math.sqrt(61):", math.sqrt(61))
```

Python Code for Plotting

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

# Step 1: Define vectors
a = np.array([1, 0, 0]) # |a| = 1
b = np.array([0, 2, 0]) # |b| = 2
c = np.array([0, 0, 3]) # |c| = 3

# Required vector
v = 3*a - 2*b + 2*c

# Step 2: Setup 3D figure
fig = plt.figure(figsize=(8,6))
ax = fig.add_subplot(111, projection='3d')
```

```
# Helper function to draw vectors with shifted labels
def draw_vector(ax, origin, vec, color, label, shift):
    ax.quiver(
        origin[0], origin[1], origin[2],
        vec[0], vec[1], vec[2],
        color=color, arrow_length_ratio=0.1, linewidth=2
    )
    ax.text(
        vec[0] + shift[0],
        vec[1] + shift[1],
        vec[2] + shift[2],
        label,
        fontsize=10, color=color
    )

# Step 3: Plot vectors with shifted labels
draw_vector(ax, [0,0,0], a, "blue", "|a|=1", shift=[0.2,0,0])
draw_vector(ax, [0,0,0], b, "green", "|b|=2", shift=[0,0.2,0])
```

```
draw_vector(ax, [0,0,0], c, "orange", "|c|=3", shift=[0,0,0.3])
draw_vector(ax, [0,0,0], v, "red", "3a - 2b + 2c", shift=[0.3,0.3,0.3])

# Step 4: Axis settings
max_range = np.max(np.abs([a, b, c, v])) + 1
ax.set_xlim([-max_range, max_range])
ax.set_ylim([-max_range, max_range])
ax.set_zlim([-max_range, max_range])

ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")
ax.set_zlabel("Z-axis")
ax.set_title("3D Vector Diagram (Non-overlapping Labels)")

plt.show()
```