2.10.39

Vector Geometry

EE25BTECH11010 - Arsh Dhoke

Question

Let $\mathbf{a} = 2\mathbf{i} + \mathbf{j} + \mathbf{k}$, $\mathbf{b} = \mathbf{i} + 2\mathbf{j} - \mathbf{k}$ and a unit vector $\mathbf{c}$ be coplanar. If $\mathbf{c}$ is perpendicular to $\mathbf{a}$, then $\mathbf{c} =$

1. $\frac{1}{\sqrt{2}}(-\mathbf{j} + \mathbf{k})$
2. $\frac{1}{\sqrt{3}}(-\mathbf{i} - \mathbf{j} - \mathbf{k})$
3. $\frac{1}{\sqrt{5}}(\mathbf{i} - 2\mathbf{j})$
4. $\frac{1}{\sqrt{3}}(\mathbf{i} - \mathbf{j} - \mathbf{k})$

Solution

Vector	Point
$\mathbf{a}$	$\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$
$\mathbf{b}$	$\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$

$$\mathbf{c} = \mathbf{a} + k\mathbf{b} \tag{1}$$

$$\mathbf{c} = \begin{pmatrix} 2+k \\ 1+2k \\ 1-k \end{pmatrix} \tag{2}$$

$$\mathbf{a}^T\mathbf{c} = 0 \tag{3}$$

$$2(2+k) + 1(1+2k) + 1(1-k) = 0 \tag{4}$$

$$6 + 3k = 0 \tag{5}$$

Solution (continued)

$$k = -2 \tag{6}$$

$$\mathbf{c} = \begin{pmatrix} 0 \\ -3 \\ 3 \end{pmatrix} \tag{7}$$

$$\|\mathbf{c}\| = \sqrt{0^2 + (-3)^2 + 3^2} = 3\sqrt{2} \tag{8}$$

$$\mathbf{c} = \frac{1}{3\sqrt{2}} \begin{pmatrix} 0 \\ -3 \\ 3 \end{pmatrix} \tag{9}$$

$$\mathbf{c} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \tag{10}$$

Thus, option 1 is correct.

Graph

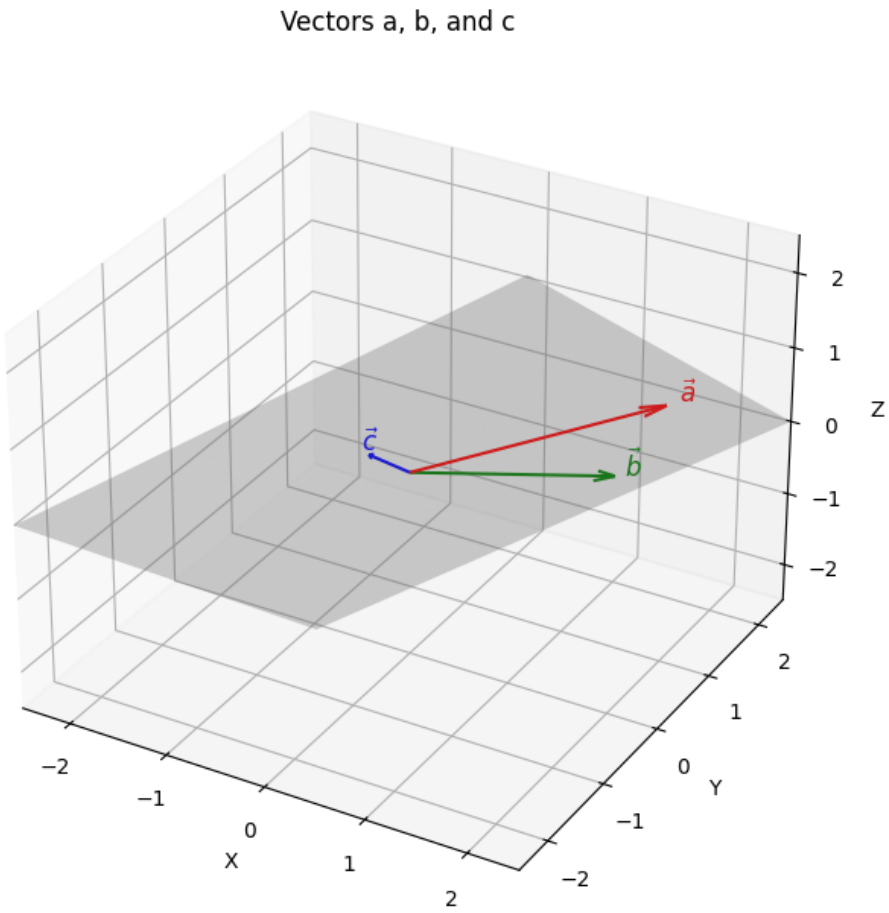

Figure: Graph of vectors

C Code

```c
#include <math.h>

typedef struct {
    double x;
    double y;
    double z;
} Vec3;

Vec3 find_unit_vector_c(void) {
    double a[3] = {2, 1, 1};
    double b[3] = {1, 2, -1};

    double dot_ab = a[0]*b[0] + a[1]*b[1] + a[2]*b[2];
    double dot_aa = a[0]*a[0] + a[1]*a[1] + a[2]*a[2];
    double k = - (double)dot_aa / dot_ab;

    double x = a[0] + k*b[0];
    double y = a[1] + k*b[1];
```

C Code

```
    double z = a[2] + k*b[2];
    double mag = sqrt(x*x + y*y + z*z);

    Vec3 result;
    result.x = x / mag;
    result.y = y / mag;
    result.z = z / mag;

    return result;
}
```

Python Code

```
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([1, 2, -1])
c = np.array([0, -1, 1]) / np.sqrt(2) # unit vector

# Compute normal to plane (a x b)
n = np.cross(a, b)
A, B, C = n # plane coefficients
# Plane equation: A*x + B*y + C*z = 0 (through origin)

# Create a grid for x and y
x_vals = np.linspace(-2.5, 2.5, 10)
y_vals = np.linspace(-2.5, 2.5, 10)
X, Y = np.meshgrid(x_vals, y_vals)
```

Python Code

```
# Solve for Z from plane equation (avoid divide by zero)
Z = (-A * X - B * Y) / C

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

origin = np.zeros(3)

# Plot the plane as a translucent surface
ax.plot_surface(X, Y, Z, alpha=0.3, color='gray')

# Plot vectors
ax.quiver(*origin, *a, color='r', arrow_length_ratio=0.1)
ax.quiver(*origin, *b, color='g', arrow_length_ratio=0.1)
ax.quiver(*origin, *c, color='b', arrow_length_ratio=0.1)
```

Python Code

```
# Add text labels near arrowheads
ax.text(*(a * 1.05), r'$\vec{a}$', color='r', fontsize=12)
ax.text(*(b * 1.05), r'$\vec{b}$', color='g', fontsize=12)
ax.text(*(c * 1.2), r'$\vec{c}$', color='b', fontsize=12)

# Set axes limits and labels
ax.set_xlim([-2.5, 2.5])
ax.set_ylim([-2.5, 2.5])
ax.set_zlim([-2.5, 2.5])
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

ax.grid(True)
plt.title("Vectors a, b, and c")
plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /2.10.39/figs/q5.png")
plt.show()
```

Python+ C Code

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D # needed for 3D plotting

# --- Mirror the C struct ---
class Vec3(ctypes.Structure):
    _fields_ = [("x", ctypes.c_double),
                ("y", ctypes.c_double),
                ("z", ctypes.c_double)]

# --- Load the shared library ---
lib = ctypes.CDLL("./code.so")
lib.find_unit_vector_c.restype = Vec3 # C returns a Vec3 struct

# --- Call the C function ---
result = lib.find_unit_vector_c()
unit_vector = np.array([result.x, result.y, result.z])
```

Python+ C Code

```
print("Unit vector from C:", unit_vector)

# --- Plot the unit vector ---
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')

# Draw axes
ax.quiver(0, 0, 0, unit_vector[0], unit_vector[1], unit_vector
    [2],
          color='blue', arrow_length_ratio=0.1, linewidth=2)

# Mark the tip of the vector
ax.scatter(unit_vector[0], unit_vector[1], unit_vector[2],
           color='red', s=50, label='Unit Vector')

# Label axes
ax.set_xlim([0,1]); ax.set_ylim([0,1]); ax.set_zlim([0,1])
ax.set_xlabel('X')
```

Python+ C Code

```
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Unit Vector from C Library')
ax.legend()

plt.tight_layout()
plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /2.10.39/figs/q5.png")
plt.show()
```