Matgeo Presentation - Direction Cosines Problem

EE25BTECH11048 - Revanth Siva Kumar D

September 28, 2025

Problem Statement

Question:
If the direction cosines of a line are (k, k, k), then find the value of k.
Goal: Determine k and visualize the direction vectors in 3D.

Theoretical Solution

The direction cosines of a line are denoted by k, k, k. So, the direction cosine vector becomes

$$\vec{d} = \begin{pmatrix} k \\ k \\ k \end{pmatrix} \tag{0.1}$$

since d is a unit vector

$$\|d\| = 1 \tag{0.2}$$

Applying condition (0.1),

$$(from(0.2)\|d\| = 1) \tag{0.3}$$

$$\| \begin{pmatrix} k \\ k \\ k \end{pmatrix} \| = 1 \tag{0.4}$$

$$\sqrt{3k^2} = 1 \tag{0.5}$$

$$3k^2 = 1 \implies k^2 = \frac{1}{3} \tag{0.6}$$

Theoretical Solution

Hence,

$$k = \pm \frac{1}{\sqrt{3}} \tag{0.7}$$

So, the line vectors are

$$\vec{v}_1 = \begin{pmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{pmatrix}, \quad \vec{v}_2 = \begin{pmatrix} -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \end{pmatrix}$$

Answer:

$$k = \frac{1}{\sqrt{3}} \quad \text{or} \quad k = -\frac{1}{\sqrt{3}}$$

C Code: points.c

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

// Function to compute direction cosine k
// Returns positive root; negative handled in Python
double compute_k() {
    double k = 1.0 / sqrt(3.0);
    return k;
}

int main() {
    double k = compute_k();
    printf("k = %lf\n", k);
    printf("-k = %lf\n", -k);
    return 0;
}
```

Python: call_c.py

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

# Load shared object
lib = ctypes.CDLL("./points.so")
lib.compute_k.restype = ctypes.c_double

# Call C function to get positive k
k_pos = lib.compute_k()
k_neg = -k_pos

print("Possible␣values␣of␣k␣from␣C:␣k␣=", k_pos, "or", k_neg)

# Vectors for plotting
vec_pos = np.array([k_pos, k_pos, k_pos])
vec_neg = np.array([k_neg, k_neg, k_neg])
```

Python: call_c.py

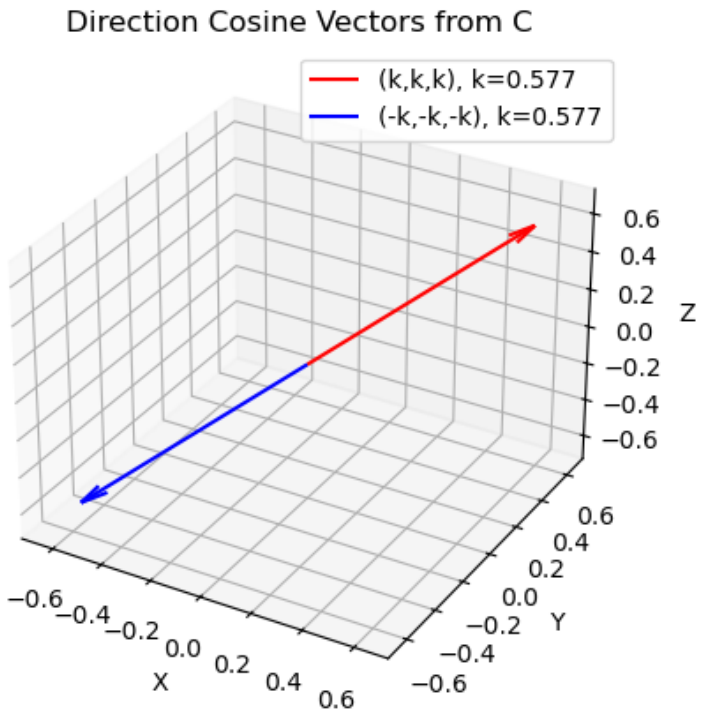
```
# 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
origin = np.array([0,0,0])

# Plot both vectors
ax.quiver(*origin, *vec_pos, color='r', arrow_length_ratio=0.1,
          label=f"(k,k,k), k={k_pos:.3f}")
ax.quiver(*origin, *vec_neg, color='b', arrow_length_ratio=0.1,
          label=f"(-k,-k,-k), k={k_pos:.3f}")

ax.set_xlim([-0.7,0.7])
ax.set_ylim([-0.7,0.7])
ax.set_zlim([-0.7,0.7])

ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Direction Cosine Vectors from C")
ax.legend()
plt.show()
```

3D Plot of Direction Cosines



3D plot of direction cosine vectors (k, k, k) and $(-k, -k, -k)$ using shared output

Python: plot.py

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

# Direction cosine values
k1 = 1 / np.sqrt(3)
k2 = -k1

print("Possible␣values:␣k␣=", k1, "or", k2)

# Vectors for plotting
vec_pos = np.array([k1, k1, k1])
vec_neg = np.array([k2, k2, k2])

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
origin = np.array([0,0,0])
```

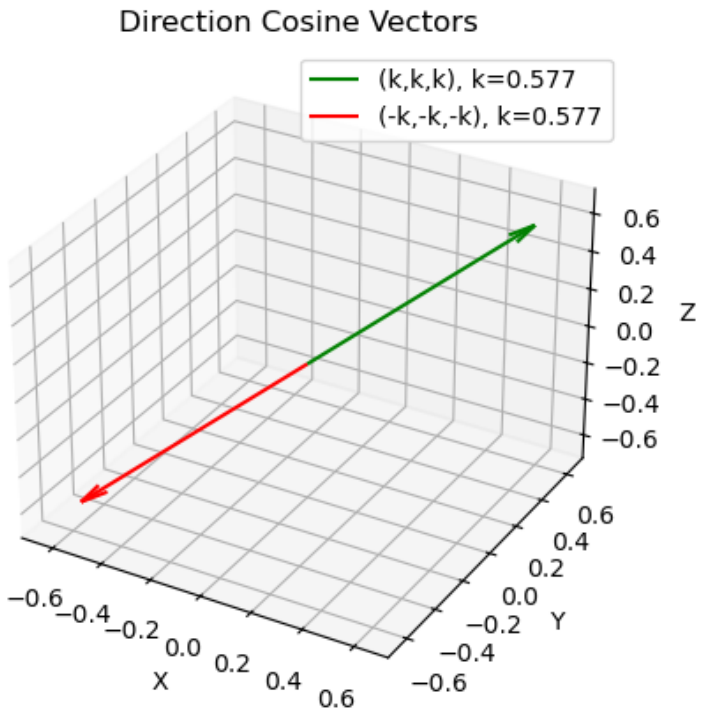
Python: plot.py

```
# Plot both vectors
ax.quiver(*origin, *vec_pos, color='g', arrow_length_ratio=0.1,
          label=f"(k,k,k), k={k1:.3f}")
ax.quiver(*origin, *vec_neg, color='r', arrow_length_ratio=0.1,
          label=f"(-k,-k,-k), k={k1:.3f}")

ax.set_xlim([-0.7,0.7])
ax.set_ylim([-0.7,0.7])
ax.set_zlim([-0.7,0.7])

ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Direction Cosine Vectors")
ax.legend()
plt.savefig("dir_cosines_both.png")
plt.show()
```

3D Plot of Direction Cosines



3D plot of direction cosine vectors (k, k, k) and $(-k, -k, -k)$ by direct python code