2.4.37

Naman Kumar-EE25BTECH11041

6 September,2025

Question

If a variable line in two adjacent positions has directions cosines l, m, n and $l+\delta l, m+\delta m, n+\delta n$, show that the small angle $\delta\theta$ between the two positions is given by

$$\delta\theta^2 = (\delta l)^2 + (\delta m)^2 + (\delta n)^2$$

Solution

We know about direction cosine of any vector,

$$l^2 + m^2 + n^2 = 1 \tag{1}$$

and angle between two vectors

$$\cos\theta = \frac{\mathbf{A}^T\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{2}$$

We also know expansion of $\cos\delta x$ (δx represents very small x)

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!}.... \tag{3}$$

Given Two direction cosine

$$l, m, n \text{ and } l + \delta l, m + \delta m, n + \delta n \tag{4}$$

Solution

Using (1) for both direction cosines

$$l^2 + m^2 + n^2 = 1 \tag{5}$$

and

$$(l + \delta l)^2 + (m + \delta m)^2 + (n + \delta n)^2 = 1 \tag{6}$$

$$l^2 + m^2 + n^2 + (\delta l)^2 + (\delta m)^2 + (\delta n)^2 + 2(l\delta l + m\delta m + n\delta n) = 1 \tag{7}$$

from (1)

$$1 + (\delta l)^2 + (\delta m)^2 + (\delta n)^2 + 2(l\delta l + m\delta m + n\delta n) = 1 \tag{8}$$

$$(\delta l)^2 + (\delta m)^2 + (\delta n)^2 = -2(l\delta l + m\delta m + n\delta n) \tag{9}$$

Solution

Using equation (2)

$$\cos\theta = \frac{\begin{pmatrix} l \\ m \\ n \end{pmatrix}^T \begin{pmatrix} l+\delta l \\ m+\delta m \\ n+\delta n \end{pmatrix}^T}{1 \times 1} \tag{10}$$

$$\cos\theta = l(l+\delta l) + m(m+\delta m) + n(n+\delta n) \tag{11}$$

$$\cos\theta = l^2 + m^2 + n^2 + l\delta l + m\delta m + n\delta n \tag{12}$$

$$\tag{13}$$

using equation (1) (3) and (9)

$$1 - \frac{\delta\theta^2}{2!} = 1 + \frac{1}{-2}(\delta l)^2 + (\delta m)^2 + (\delta n)^2 \tag{14}$$

Where $\delta\theta$ represents very small θ

$$\delta\theta^2 = (\delta l)^2 + (\delta m)^2 + (\delta n)^2 \tag{15}$$

Hence Proved

Figure

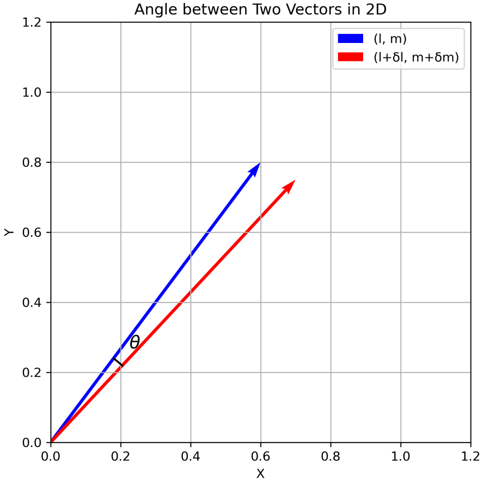

Figure: Caption

C code

```c
#include <stdio.h>
#include <math.h>
void dot_product(double v1[], double v2[], int size, double* result) {
    *result = 0.0;
    for (int i = 0; i < size; i++) {
        *result += v1[i] * v2[i];
    }
    return result;
}
```

Python from Shared Object

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

# Load the shared library
lib = ctypes.CDLL("./libdot.so")

# Define function signature
lib.dot_product.argtypes = [
    np.ctypeslib.ndpointer(dtype=np.double),
    np.ctypeslib.ndpointer(dtype=np.double),
    ctypes.c_int,
    ctypes.POINTER(ctypes.c_double)
]
```

Python from Shared Object

```python
# Vectors
l, m = 0.6, 0.8
dl, dm = 0.1, -0.05

v1 = np.array([l, m], dtype=np.double)
v2 = np.array([l+dl, m+dm], dtype=np.double)

# Call C function for dot product
res = ctypes.c_double()
lib.dot_product(v1, v2, 2, ctypes.byref(res))
dot = res.value

# Normalize
v1u = v1 / np.linalg.norm(v1)
v2u = v2 / np.linalg.norm(v2)
```

```python
# Angle between vectors
theta = np.arccos(np.clip(dot / (np.linalg.norm(v1) * np.linalg.
    norm(v2)), -1.0, 1.0))

# Angles relative to x-axis
a1 = np.arctan2(v1u[1], v1u[0])
a2 = np.arctan2(v2u[1], v2u[0])
start, end = sorted([a1, a2])

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

ax.quiver(0, 0, v1[0], v1[1], angles='xy', scale_units='xy',
    scale=1, color='b', label='(l,m)')
ax.quiver(0, 0, v2[0], v2[1], angles='xy', scale_units='xy',
    scale=1, color='r', label='(l+dl,m+dm)')
```

Python from Shared Object

```
# Arc for angle
r = 0.3
arc_angles = np.linspace(start, end, 100)
arc_x = r * np.cos(arc_angles)
arc_y = r * np.sin(arc_angles)
ax.plot(arc_x, arc_y, 'k-')

# Label theta
mid = (start + end) / 2
ax.text(0.35*np.cos(mid), 0.35*np.sin(mid), r'$\theta$', fontsize
    =14)
```

Python from Shared Object

```python
# Formatting
ax.set_xlim(0, 1.2)
ax.set_ylim(0, 1.2)
ax.set_aspect('equal')
ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_title("Angle between two vectors (C + Python)")
ax.legend()
ax.grid(True)

plt.savefig("vectors.png", dpi=300)
plt.show()
```

Direct Python

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

# Example direction cosines (2D projection)
l, m = 0.6, 0.8
dl, dm = 0.1, -0.05

# Vectors
v1 = np.array([l, m])
v2 = np.array([l+dl, m+dm])

# Normalize
v1_u = v1 / np.linalg.norm(v1)
v2_u = v2 / np.linalg.norm(v2)
```

Direct Python

```
# Compute their angles w.r.t x-axis
angle1 = np.arctan2(v1_u[1], v1_u[0])
angle2 = np.arctan2(v2_u[1], v2_u[0])

# Ensure correct order (draw smaller arc between them)
start_angle = min(angle1, angle2)
end_angle = max(angle1, angle2)

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

Direct Python

```python
# Draw vectors
ax.quiver(0, 0, v1[0], v1[1], angles='xy', scale_units='xy',
    scale=1, color='b', label='(l, m)')
ax.quiver(0, 0, v2[0], v2[1], angles='xy', scale_units='xy',
    scale=1, color='r', label='(l+l, m+m)')

# Draw arc for angle
arc_radius = 0.3
arc_angles = np.linspace(start_angle, end_angle, 100)
arc_x = arc_radius * np.cos(arc_angles)
arc_y = arc_radius * np.sin(arc_angles)
ax.plot(arc_x, arc_y, 'k-')
```

Direct Python

```
# Label  at midpoint of arc
mid_angle = (start_angle + end_angle) / 2
ax.text(0.35*np.cos(mid_angle), 0.35*np.sin(mid_angle), r'$\
    theta$', fontsize=14)

# Formatting
ax.set_xlim(0, 1.2)
ax.set_ylim(0, 1.2)
ax.set_aspect('equal')
ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_title("Angle between Two Vectors in 2D")
ax.legend()
ax.grid(True)

# Save figure
plt.savefig("vectors.png", dpi=300)
plt.show()
```