2.3.4

EE25BTECH11001 - Aarush Dilawri

Sep 1, 2025

Question

Problem:

a and **b** are two unit vectors such that

$$|2\mathbf{a} + 3\mathbf{b}| = |3\mathbf{a} - 2\mathbf{b}| \tag{1}$$

Find the angle between **a** and **b**.

Setup

Let $\mathbf{a}, \mathbf{b}$ be unit vectors with angle θ between them.

$$\mathbf{a} \cdot \mathbf{a} = 1, \quad \mathbf{b} \cdot \mathbf{b} = 1, \quad \mathbf{a} \cdot \mathbf{b} = \cos\theta$$

The Gram matrix is

$$\mathbf{G} = \begin{pmatrix} 1 & \cos\theta \\ \cos\theta & 1 \end{pmatrix}.$$

Norm Calculation (1)

$$\left\|2\mathbf{a} + 3\mathbf{b}\right\|^2 = \begin{pmatrix} 2 & 3 \end{pmatrix} \mathbf{G} \begin{pmatrix} 2 \\ 3 \end{pmatrix} \tag{2}$$

$$= \begin{pmatrix} 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & \cos\theta \\ \cos\theta & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} \tag{3}$$

$$= 13 + 12\cos\theta \tag{4}$$

Norm Calculation (2)

$$\left\|3\mathbf{a} - 2\mathbf{b}\right\|^2 = \begin{pmatrix} 3 & -2 \end{pmatrix} \mathbf{G} \begin{pmatrix} 3 \\ -2 \end{pmatrix} \tag{5}$$

$$= \begin{pmatrix} 3 & -2 \end{pmatrix} \begin{pmatrix} 1 & \cos\theta \\ \cos\theta & 1 \end{pmatrix} \begin{pmatrix} 3 \\ -2 \end{pmatrix} \tag{6}$$

$$= 13 - 12\cos\theta \tag{7}$$

Equating Norms

$$13 + 12\cos\theta = 13 - 12\cos\theta \tag{8}$$

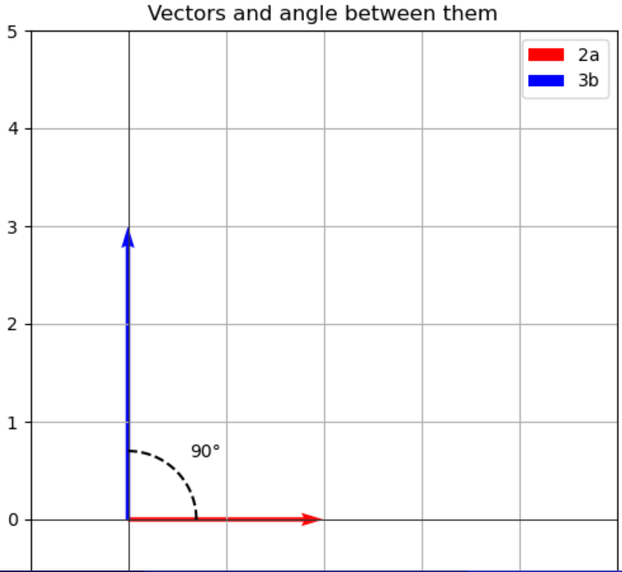
$$24\cos\theta = 0 \tag{9}$$

$$\cos\theta = 0 \tag{10}$$

Therefore,

$$\theta = \frac{\pi}{2} \quad (90^\circ).$$

Plot



C Code (code.c)

```c
#include<math.h>

double angle_between(double p, double q, double r, double s) {
    double numerator = - (p*p + q*q - r*r - s*s);
    double denominator = 2.0 * (p*q - r*s);
    double cos_theta = numerator / denominator;
    return acos(cos_theta);
}
```

Python Code (code.py)

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

if __name__ == ”__main__”:
    # Define two unit vectors directly
    a = np.array([1, 0]) # along x−axis
    b = np.array([0, 1]) # along y−axis (90 from a)

    # Scale them like in the specific case
    a_scaled = 2 * a
    b_scaled = 3 * b
# Compute angle
    dot_product = np.dot(a, b)
```

Python Code (code.py)

```
theta = math.acos(dot_product / (np.linalg.norm(a) ∗ np.linalg.norm(
    b)))
print(”Angle␣(radians):”, theta)
print(”Angle␣(degrees):”, math.degrees(theta))
plt.figure(figsize=(6,6))
ax = plt.gca()
ax.set_aspect(’equal’)
# Plot vectors
plt.quiver(0, 0, a_scaled[0], a_scaled[1], angles=’xy’, scale_units=’xy’,
    scale=1,
           color=’r’, label=’2a’)
plt.quiver(0, 0, b_scaled[0], b_scaled[1], angles=’xy’, scale_units=’xy’,
    scale=1,
           color=’b’, label=’3b’)
```

Python Code (code.py)

```
    angle_arc = np.linspace(0, theta, 100)
    plt.plot(0.7*np.cos(angle_arc), 0.7*np.sin(angle_arc), 'k--')
    plt.text(0.9*math.cos(theta/2), 0.9*math.sin(theta/2), f"{math.
        degrees(theta):.0f}")
    plt.axhline(0, color='black', linewidth=0.5)
    plt.axvline(0, color='black', linewidth=0.5)
    plt.xlim(-1, 5)
    plt.ylim(-1, 5)
    plt.legend()
    plt.title("Vectors-and-angle-between-them-(Python-only)")
    plt.grid(True)
    plt.show()
```

Python Code (nativecode.py)

```python
from ctypes import CDLL, c_double
import os, math
import numpy as np
import matplotlib.pyplot as plt

# Load the shared library (libcode.so must be in the same folder)
_lib_path = os.path.join(os.path.dirname(__file__), ”code.so”)
_lib = CDLL(_lib_path)

# Define arg and return types
_lib.angle_between.argtypes = [c_double, c_double, c_double, c_double]
_lib.angle_between.restype = c_double
```

Python Code (nativecode.py)

```
def angle_between(p, q, r, s):
    """"Call the C function and return angle in radians""""
    return _lib.angle_between(p, q, r, s)

if __name__ == "__main__":
    # Specific case: p=2, q=3, r=3, s=-2
    p, q, r, s = 2, 3, 3, -2
    theta = angle_between(p, q, r, s)
    print("Angle␣(radians):", theta)
    print("Angle␣(degrees):", math.degrees(theta))

    # Pick two unit vectors with the computed angle
    # Take a = (1,0), then b = (cos, sin)
```

Python Code (nativecode.py)

```
a = np.array([1, 0])
b = np.array([math.cos(theta), math.sin(theta)])

# Scale them for visualization
a_scaled = p * a
b_scaled = q * b

plt.figure(figsize=(6,6))
ax = plt.gca()
ax.set_aspect('equal')
```

Python Code (nativecode.py)

```
# Plot vectors
    plt.quiver(0, 0, a_scaled[0], a_scaled[1], angles=’xy’, scale_units=’xy’,
        scale=1,
               color=’r’, label=’2a’)
    plt.quiver(0, 0, b_scaled[0], b_scaled[1], angles=’xy’, scale_units=’xy’,
        scale=1,
               color=’b’, label=’3b’)

    # Draw angle arc
    angle_arc = np.linspace(0, theta, 100)
    plt.plot(0.7∗np.cos(angle_arc), 0.7∗np.sin(angle_arc), ’k——’)
    plt.text(0.9∗math.cos(theta/2), 0.9∗math.sin(theta/2), f”{math.
        degrees(theta):.0f}”)
```

Python Code (nativecode.py)

```
# Axes
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.xlim(−1, max(p,q)+2)
plt.ylim(−1, max(p,q)+2)
plt.legend()
plt.title("Vectors␣and␣angle␣between␣them")
plt.grid(True)
plt.show()
```