Angle Between Vectors Using Gram Matrix

EE25BTECH11008 - Anirudh M Abhilash

September 14, 2025

Problem Statement

Find the acute angle between the planes

$$x - 2y - 2z = 5$$
$$3x - 6y + 2z = 7$$

Solution

The angle between two planes is the angle between their normals. Let

$$\mathbf{n}_1 = \begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix}, \quad \mathbf{n}_2 = \begin{pmatrix} 3 \\ -6 \\ 2 \end{pmatrix}.$$

Form the matrix

$$A = \begin{pmatrix} 1 & 3 \\ -2 & -6 \\ -2 & 2 \end{pmatrix}, \tag{1}$$

whose columns are the normals. The Gram matrix is

$$G = A^\top A = \begin{pmatrix} \mathbf{n}_1^\top \mathbf{n}_1 & \mathbf{n}_1^\top \mathbf{n}_2 \\ \mathbf{n}_2^\top \mathbf{n}_1 & \mathbf{n}_2^\top \mathbf{n}_2 \end{pmatrix}. \tag{2}$$

Solution (cont..)

Now,

$$\mathbf{n}_1^\top \mathbf{n}_1 = 9, \quad \mathbf{n}_2^\top \mathbf{n}_2 = 49, \quad \mathbf{n}_1^\top \mathbf{n}_2 = 11.$$

Thus,

$$G = \begin{pmatrix} 9 & 11 \\ 11 & 49 \end{pmatrix}. \tag{3}$$

Let

$$D = \begin{pmatrix} 9 & 0 \\ 0 & 49 \end{pmatrix}, \quad D^{-1/2} = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{7} \end{pmatrix}. \tag{4}$$

Solution (cont..)

The normalized Gram matrix is

$$C = D^{-1/2} G D^{-1/2} = \begin{pmatrix} 1 & \frac{11}{21} \\ \frac{11}{21} & 1 \end{pmatrix}. \tag{5}$$

The off-diagonal entry gives

$$\cos\theta = \frac{11}{21}. \tag{6}$$

Hence, the acute angle between the planes is

$$\boxed{\theta = \arccos\left(\tfrac{11}{21}\right) \approx 58.41^\circ}$$

Python Code (Plotting Normals)

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

u = np.array([1, −2, −2])
v = np.array([3, −6, 2])
origin = np.zeros(3)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.quiver(*origin, *u, color='r', arrow_length_ratio=0.1)
ax.text(u[0]*1.1, u[1]*1.1, u[2]*1.1, "u", color='r')
```

Python Code (cont..)

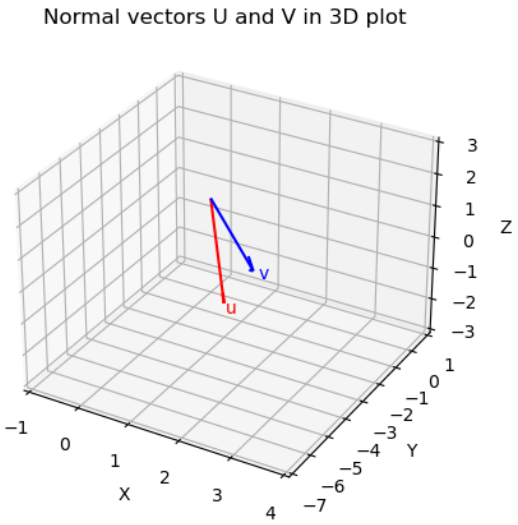
```
ax.quiver(*origin, *v, color='b', arrow_length_ratio=0.1)
ax.text(v[0]*1.1, v[1]*1.1, v[2]*1.1, "v", color='b')

all_points = np.vstack([origin, u, v])
ax.set_xlim([all_points[:,0].min()-1, all_points[:,0].max()+1])
ax.set_ylim([all_points[:,1].min()-1, all_points[:,1].max()+1])
ax.set_zlim([all_points[:,2].min()-1, all_points[:,2].max()+1])

ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Normal vectors U and V in 3D plot")

plt.show()
```

Plot (Python)



Normal vectors U and V in 3D plot
v
u
X
Y
Z
−1
0
1
2
3
4
−7
−6
−5
−4
−3
−2
−1
0
1
−3
−2
−1
0
1
2
3

C Code (Matrix multiplication)

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

void matmul(double A[2][2], double B[2][2], double C[2][2]) {
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            C[i][j] = 0.0;
            for (int k = 0; k < 2; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }
}
```

C Code (Finding G)

```c
void gram_matrix(double *u, double *v, int n, double G[2][2]) {
    double g11 = 0.0, g22 = 0.0, g12 = 0.0;
    for (int i = 0; i < n; i++) {
        g11 += u[i] * u[i];
        g22 += v[i] * v[i];
        g12 += u[i] * v[i];
    }
    G[0][0] = g11;
    G[0][1] = g12;
    G[1][0] = g12;
    G[1][1] = g22;
}
```

C Code (Normalising G)

```c
void normalize_gram(double G[2][2], double G_norm[2][2]) {
    double d11 = 1.0 / sqrt(G[0][0]);
    double d22 = 1.0 / sqrt(G[1][1]);

    double Dinv[2][2] = {{d11, 0.0}, {0.0, d22}};

    double temp[2][2];
    matmul(Dinv, G, temp);
    matmul(temp, Dinv, G_norm);
}
```

C Code (Finding Angle)

```
double angle_from_normalized(double G_norm[2][2]) {
    double cos_theta = G_norm[0][1];
    if (cos_theta > 1.0) {
        cos_theta = 1.0;
    }
    if (cos_theta < -1.0) {
        cos_theta = -1.0;
    }
    return acos(cos_theta);
}
```

Python Code (Calling C)

```
import ctypes
import numpy

lib = ctypes.CDLL("./computations.so")

lib.gram_matrix.argtypes = [
    ctypes.POINTER(ctypes.c_double),
    ctypes.POINTER(ctypes.c_double),
    ctypes.c_int,
    ctypes.POINTER((ctypes.c_double * 2) * 2)
]
```

Python Code (cont..)

```
lib.normalize_gram.argtypes = [
    ctypes.POINTER((ctypes.c_double * 2) * 2),
    ctypes.POINTER((ctypes.c_double * 2) * 2)
]

lib.angle_from_normalized.argtypes = [
    ctypes.POINTER((ctypes.c_double * 2) * 2)
]
lib.angle_from_normalized.restype = ctypes.c_double
```

Python Code (cont..)

```
def compute_angle(u, v):
    n = len(u)
    u_arr = (ctypes.c_double * n)(*u)
    v_arr = (ctypes.c_double * n)(*v)

    G = ((ctypes.c_double * 2) * 2)()
    G_norm = ((ctypes.c_double * 2) * 2)()

    lib.gram_matrix(u_arr, v_arr, n, G)
    lib.normalize_gram(G, G_norm)
    theta = lib.angle_from_normalized(G_norm)
    return theta
```

Python Code (cont..)

```
u = [1, -2, -2]
v = [3, -6, 2]
theta = compute_angle(u, v)

print(f"Angle(radians):{theta}")
print(f"Angle(degrees):{theta*180/numpy.pi}")
```