2.3.13

Vector Geometry

EE25BTECH11010 - Arsh Dhoke

Question

Find the angle which the line $\frac{x}{1} = \frac{y}{-1} = \frac{z}{2}$ makes with the positive direction of the Y axis.

Angle Between Line and Y-axis

The line can be represented as $k \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$

Hence its direction vector is

$$\mathbf{v} = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix} \tag{1}$$

$$\mathbf{e_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{2}$$

$$\mathbf{v}^T\mathbf{e_2} = \begin{pmatrix} 1 & -1 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = -1 \tag{3}$$

Calculation

$$\|\mathbf{v}\| = \sqrt{\mathbf{v}^T\mathbf{v}} = \sqrt{\begin{pmatrix}1 & -1 & 2\end{pmatrix}\begin{pmatrix}1\\-1\\2\end{pmatrix}} = \sqrt{6} \tag{4}$$

$$\|\mathbf{e_2}\| = 1 \tag{5}$$

$$\cos\theta = \frac{\mathbf{v}^T\mathbf{e_2}}{\|\mathbf{v}\|\|\mathbf{e_2}\|} = \frac{-1}{\sqrt{6}} \tag{6}$$

$$\theta = \cos^{-1}\left(-\frac{1}{\sqrt{6}}\right) \tag{7}$$

$$\boxed{\theta = \cos^{-1}\left(-\frac{1}{\sqrt{6}}\right) \approx 114.09^\circ}$$

Plot

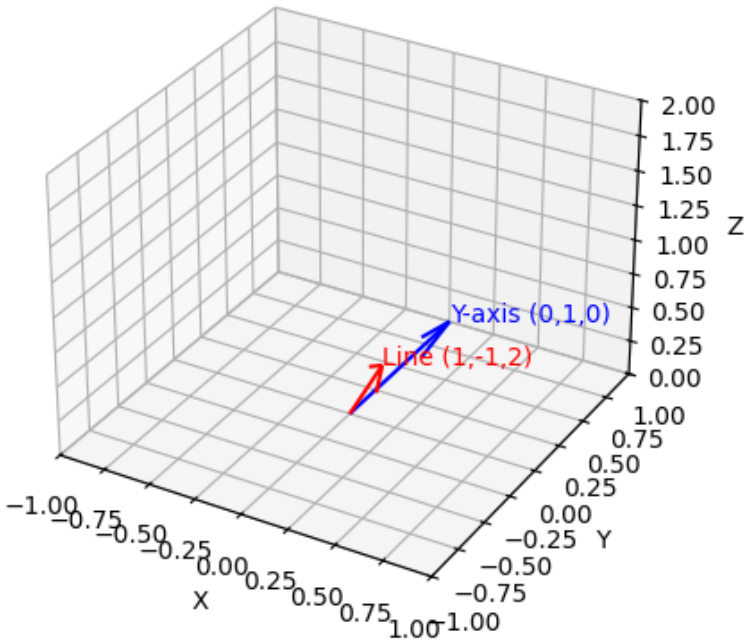

Figure: Graph

C Code

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

double angleWithYAxis(double vx, double vy, double vz) {
    double e2y = 1.0;
    double dot = vy * e2y;
    double magv = sqrt(vx*vx + vy*vy + vz*vz);
    double mage2 = 1.0;
    double cosTheta = dot / (magv * mage2);
    double thetaRad = acos(cosTheta);
    double thetaDeg = thetaRad * 180.0 / M_PI;
    return thetaDeg;
}
int main() {
    double vx = 1, vy = -1, vz = 2;
    double theta = angleWithYAxis(vx, vy, vz);
    printf("Angle with positive Y-axis = %.2f degrees\n", theta);
    return 0;
}
```

Python Code

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

# Vectors
v = np.array([1, -1, 2])
e2 = np.array([0, 1, 0])

# Normalize for plotting
v_unit = v / np.linalg.norm(v)
e2_unit = e2 / np.linalg.norm(e2)

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

origin = np.array([0, 0, 0])

# Plot vectors
```

Python Code

```
ax.quiver(*origin, *v_unit, color='r')
ax.quiver(*origin, *e2_unit, color='b')

# Add labels next to the tips
ax.text(*v_unit, "Line (1,-1,2)", color='r', fontsize=10)
ax.text(*e2_unit, "Y-axis (0,1,0)", color='b', fontsize=10)

# Axes labels and limits
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_xlim([-1, 1])
ax.set_ylim([-1, 1])
ax.set_zlim([0, 2])
ax.grid(True)

plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /2.3.13/figs/q3.png")
plt.show()
```

Python+ C Code

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

# Load the shared library
lib = CDLL("./code.so") # use your code.so file

# Define argument and return types
lib.angleWithYAxis.argtypes = [c_double, c_double, c_double]
lib.angleWithYAxis.restype = c_double

# Vector
vx, vy, vz = 1.0, -1.0, 2.0

# Call C function
theta_deg = lib.angleWithYAxis(vx, vy, vz)
print(f"Angle with positive Y-axis = {theta_deg:.2f} degrees")
```

Python+ C Code

```
# 3D plotting
v = np.array([vx, vy, vz])
e2 = np.array([0, 1, 0]) # Y-axis unit vector

# Normalize for plotting
v_unit = v / np.linalg.norm(v)
e2_unit = e2 / np.linalg.norm(e2)

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

# Plot vectors
ax.quiver(*origin, *v_unit, color='r', length=1)
ax.quiver(*origin, *e2_unit, color='b', length=1)

# Labels
```

Python+ C Code

```
ax.text(*(v_unit + 0.1), f"Line {tuple(v)}", color='r', fontsize
    =10)
ax.text(*(e2_unit + 0.1), f"Y-axis {tuple(e2)}", color='b',
    fontsize=10)
ax.text(0.2, 0.2, 0.2, f"Angle with Y-axis: {theta_deg:.2f}",
    color='k', fontsize=12)

# Axes
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_xlim([-1, 1])
ax.set_ylim([-1, 1])
ax.set_zlim([0, 2])
ax.grid(True)

plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /2.3.13/figs/q3.png")
plt.show()
```