# MatGeo Presentation - Problem 1.10.29

EE25BTECH11061 - V.Sainadh

# Question

A vector $\mathbf{r}$ is inclined at equal angles to the three axes. If the magnitude of $\mathbf{r}$ is $2\sqrt{3}$ units, find $\mathbf{r}$.

# Solution

$\rightarrow$ A vector equally inclined to all three coordinate axes has equal components. Let the common scale be $c$. Then,

$$\mathbf{r} = c \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{0.1}$$

$$\|\mathbf{r}\| = |c|\sqrt{1^2 + 1^2 + 1^2} = |c|\sqrt{3}. \tag{0.2}$$

Given $\|\mathbf{r}\| = 2\sqrt{3}$,

$$2\sqrt{3} = |c|\sqrt{3} \tag{0.3}$$

$$\implies |c| = 2. \tag{0.4}$$

Hence,

$$\mathbf{r} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \quad \text{or} \quad \mathbf{r} = \begin{pmatrix} -2 \\ -2 \\ -2 \end{pmatrix}. \tag{0.5}$$

# Plot

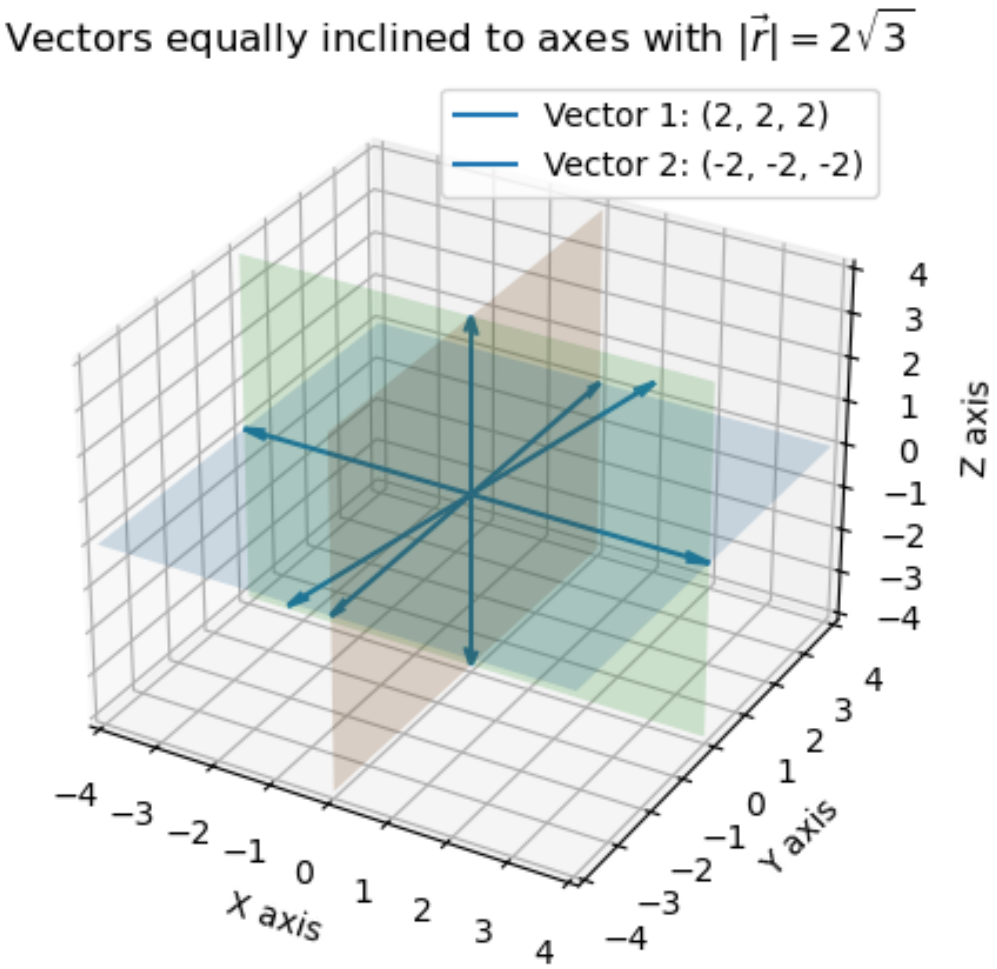


Figure: Plot of the vector **r**

# File: points.c

```
#include <stdio.h>

int main() {
  FILE *fp;




  fp = fopen("points.dat", "w");
  fprintf(fp, "%d,%d,%d\n", 2, 2, 2); // 1
  fprintf(fp, "%d,%d,%d\n", -2, -2, -2); // 2
  fclose(fp);
  return 0;
  }
```

# File: call_c.py

```
import subprocess

# Compile the C program
subprocess.run(["gcc", "points.c", "-o", "points"])

# Run the compiled C program
result = subprocess.run(["./points"], capture_output=True, text=True)

# Print the output from the C program
print(result.stdout)
```

# File: plot.py

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D # noqa: F401

# Points for |r| = 2*sqrt(3) equally inclined -> components equal (2, 2, 2)
point1 = np.array([2, 2, 2], dtype=float)
point2 = np.array([-2, -2, -2], dtype=float)

# Figure and 3D axis
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Vectors from origin
vector1 = point1
vector2 = point2

# Plot the two vectors
ax.quiver(0, 0, 0, vector1[0], vector1[1], vector1[2],
          label='Vector 1: (2, 2, 2)', arrow_length_ratio=0.1)
ax.quiver(0, 0, 0, vector2[0], vector2[1], vector2[2],
          label='Vector 2: (-2, -2, -2)', arrow_length_ratio=0.1)
```

# File: plot.py

```
# Draw coordinate axes (both positive and negative)
scale = 4
ax.quiver(0, 0, 0, scale, 0, 0, arrow_length_ratio=0.1)
ax.quiver(0, 0, 0, -scale, 0, 0, arrow_length_ratio=0.1)
ax.quiver(0, 0, 0, 0, scale, 0, arrow_length_ratio=0.1)
ax.quiver(0, 0, 0, 0, -scale, 0, arrow_length_ratio=0.1)
ax.quiver(0, 0, 0, 0, 0, scale, arrow_length_ratio=0.1)
ax.quiver(0, 0, 0, 0, 0, -scale, arrow_length_ratio=0.1)

# Plot the coordinate planes (transparent)
xx, yy = np.meshgrid(np.linspace(-scale, scale, 10),
                     np.linspace(-scale, scale, 10))
zz = np.zeros_like(xx)
ax.plot_surface(xx, yy, zz, alpha=0.2, rstride=100, cstride=100) # XY-plane

yy, zz = np.meshgrid(np.linspace(-scale, scale, 10),
                     np.linspace(-scale, scale, 10))
xx = np.zeros_like(yy)
ax.plot_surface(xx, yy, zz, alpha=0.2, rstride=100, cstride=100) # YZ-plane

xx, zz = np.meshgrid(np.linspace(-scale, scale, 10),
                     np.linspace(-scale, scale, 10))
yy = np.zeros_like(xx)
ax.plot_surface(xx, yy, zz, alpha=0.2, rstride=100, cstride=100) # ZX-plane
```

# File: plot.py

```
# Limits and labels
ax.set_xlim([-scale, scale])
ax.set_ylim([-scale, scale])
ax.set_zlim([-scale, scale])
ax.set_xlabel('X␣axis')
ax.set_ylabel('Y␣axis')
ax.set_zlabel('Z␣axis')

ax.set_title(r"Vectors␣equally␣inclined␣to␣axes␣with␣$|\vec␣r|=2\sqrt{3}$")
ax.legend()
plt.show()
```