Matgeo Presentation - Problem 2.9.14

ee25btech11063 - Vejith

August 30, 2025

Question

The two adjacent sides of a parallelogram are represented by $2\hat{i} + 4\hat{j} + -5\hat{k}$ and $\hat{i} + 2\hat{j} + 3\hat{k}$.find the unit vectors parallel to its diagonals.using diagonal vectors find the area of parallelogram

Description

Solution:

vector	**Name**
$\begin{pmatrix} 2 \\ 4 \\ -5 \end{pmatrix}$	Vector $\mathbf{a}$
$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$	Vector $\mathbf{b}$

Table: Variables Used

Solution

The diagonals of the parallelogram are given by

$$\mathbf{a} + \mathbf{b} = \begin{pmatrix} 3 \\ 6 \\ -2 \end{pmatrix} \quad \text{and} \quad \mathbf{a} - \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ -8 \end{pmatrix} \tag{0.1}$$

The corresponding unit vectors parallel to diagonals are

$$\frac{\mathbf{a} + \mathbf{b}}{\|\mathbf{a} + \mathbf{b}\|} = \begin{pmatrix} \frac{3}{7} \\ \frac{6}{7} \\ \frac{-2}{7} \end{pmatrix} \quad \text{and} \quad \frac{\mathbf{a} - \mathbf{b}}{\|\mathbf{a} - \mathbf{b}\|} = \begin{pmatrix} \frac{1}{\sqrt{69}} \\ \frac{2}{\sqrt{69}} \\ \frac{-8}{\sqrt{69}} \end{pmatrix} \tag{0.2}$$

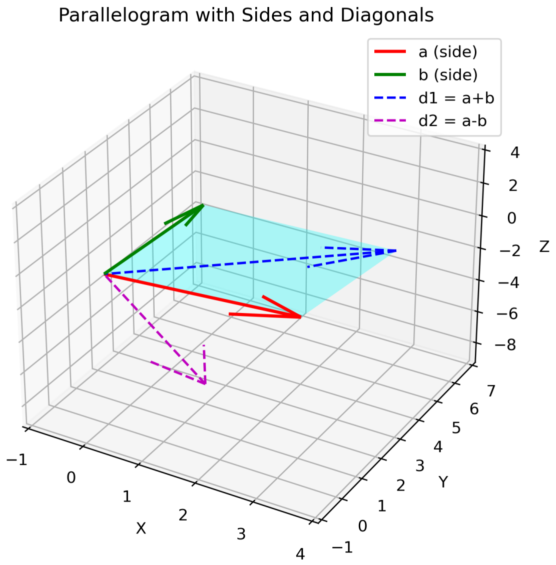
If $\mathbf{d1}$ and $\mathbf{d}2$ are the diagonals of a parallelogram then area of parallelogram is $=\frac{1}{2}\|\mathbf{d1} \times \mathbf{d2}\|$

Conclusion

$$\rightarrow \text{area of parallelogram} = \frac{1}{2}\|(\mathbf{a}+\mathbf{b}) \times (\mathbf{a}-\mathbf{b})\| \implies \text{area} = \frac{1}{2}\|\begin{pmatrix} -44 \\ 22 \\ 0 \end{pmatrix}\| \tag{0.3}$$

$$= \|\begin{pmatrix} -22 \\ 11 \\ 0 \end{pmatrix}\| = \sqrt{605} = 24.59 \tag{0.4}$$

Plot



Parallelogram with Sides and Diagonals
a (side)
b (side)
d1 = a+b
d2 = a-b
X
Y
Z
−1
0
1
2
3
4
−1
0
1
2
3
4
5
6
7
4
2
0
−2
−4
−6
−8

C Code: code.c

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

/* magnitude of a 3D vector */
double magnitude(const double v[3]) {
    return sqrt(v[0]*v[0] + v[1]*v[1] + v[2]*v[2]);}
/* normalize a 3D vector into unit; if zero vector, sets unit to 0,0,0 */
void normalize(const double v[3], double unit[3]) {
    double mag = magnitude(v);
    if (mag == 0.0) {
        unit[0] = unit[1] = unit[2] = 0.0;
        return;}
    unit[0] = v[0] / mag;
    unit[1] = v[1] / mag;
    unit[2] = v[2] / mag;}
int main(void) {
    FILE *fp = fopen("plgm.dat", "w");
    if (fp == NULL) {
        perror("fopen");
        return 1;
    }
/* Given adjacent sides */
    double a[3] = {2.0, 4.0, -5.0};
    double b[3] = {1.0, 2.0, 3.0};

    /* Diagonals */
    double d1[3], d2[3];
```

C Code: code.c

```
    for (int i = 0; i < 3; ++i) {
        d1[i] = a[i] + b[i]; /* first diagonal */
        d2[i] = a[i] - b[i]; /* second diagonal */
    }

    /* Unit vectors parallel to diagonals */
    double u1[3], u2[3];
    normalize(d1, u1);
    normalize(d2, u2);
  /* Cross product of diagonals and area = 0.5 * |d1 x d2| */
    double cross[3];
    cross[0] = d1[1]*d2[2] - d1[2]*d2[1];
    cross[1] = d1[2]*d2[0] - d1[0]*d2[2];
    cross[2] = d1[0]*d2[1] - d1[1]*d2[0];
    double area = 0.5 * magnitude(cross);

    /* Write results to plgm.dat */
    fprintf(fp, "Adjacent␣sides:\n");
    fprintf(fp, "a␣=␣(%.2f,␣%.2f,␣%.2f)\n", a[0], a[1], a[2]);
    fprintf(fp, "b␣=␣(%.2f,␣%.2f,␣%.2f)\n\n", b[0], b[1], b[2]);

    fprintf(fp, "Diagonals:\n");
    fprintf(fp, "d1␣=␣(%.2f,␣%.2f,␣%.2f)\n", d1[0], d1[1], d1[2]);
    fprintf(fp, "d2␣=␣(%.2f,␣%.2f,␣%.2f)\n\n", d2[0], d2[1], d2[2]);

    fprintf(fp, "Unit␣vectors␣parallel␣to␣diagonals:\n");
    fprintf(fp, "u1␣=␣(%.6f,␣%.6f,␣%.6f)\n", u1[0], u1[1], u1[2]);
    fprintf(fp, "u2␣=␣(%.6f,␣%.6f,␣%.6f)\n\n", u2[0], u2[1], u2[2]);
    fprintf(fp, "Area␣of␣parallelogram␣=␣%.6f\n", area);
    fclose(fp);
    printf("plgm.dat␣written␣successfully.\n");
    return 0;
}
```

Python: plot.py

```
   import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.art3d import Poly3DCollection

# Given adjacent sides
a = np.array([2, 4, -5])
b = np.array([1, 2, 3])

# Diagonals
d1 = a + b
d2 = a - b

# Define parallelogram vertices
O = np.array([0, 0, 0]) # Origin
A = a
B = b
C = a + b # Opposite vertex

# Setup 3D plot
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection="3d")

# Draw the parallelogram surface
verts = [[O, A, C, B]]
ax.add_collection3d(Poly3DCollection(verts, alpha=0.3, facecolor="cyan"))

# Plot vectors for sides
ax.quiver(0, 0, 0, a[0], a[1], a[2], color="r", label="a␣(side)", linewidth=2)
ax.quiver(0, 0, 0, b[0], b[1], b[2], color="g", label="b␣(side)", linewidth=2)

# Plot diagonals
ax.quiver(0, 0, 0, d1[0], d1[1], d1[2], color="b", linestyle="dashed", label="d1␣=␣a+b")
ax.quiver(0, 0, 0, d2[0], d2[1], d2[2], color="m", linestyle="dashed", label="d2␣=␣a-b")
```

Python: plot.py

```
# Set labels
ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Parallelogram with Sides and Diagonals")

# Auto scale
max_range = np.array([a, b, d1, d2]).max() - np.array([a, b, d1, d2]).min()
Xb = np.array([O[0], A[0], B[0], C[0], d1[0], d2[0]])
Yb = np.array([O[1], A[1], B[1], C[1], d1[1], d2[1]])
Zb = np.array([O[2], A[2], B[2], C[2], d1[2], d2[2]])

ax.set_xlim([Xb.min()-1, Xb.max()+1])
ax.set_ylim([Yb.min()-1, Yb.max()+1])
ax.set_zlim([Zb.min()-1, Zb.max()+1])

ax.legend()

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