2.6.39

Manohar-AI25BTECH11028

August 31,2025

Question

The area of the quadrilateral ABCD, where A(0,4,1), B(2,3,-1), C(4,5,0) and D(2,6,2), is equal to

Equation

The area of a quadrilateral is given by half the magnitude of the cross product of its diagonals.

First, we find the vectors for the diagonals

$$\mathbf{P} = \mathbf{C} - \mathbf{A} = \begin{pmatrix} 4 \\ 5 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ -1 \end{pmatrix}$$

$$\mathbf{Q} = \mathbf{D} - \mathbf{B} = \begin{pmatrix} 2 \\ 6 \\ 2 \end{pmatrix} - \begin{pmatrix} 2 \\ 3 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix}$$

Solution

Now, we compute the cross product $\mathbf{P} \times \mathbf{Q}$ using the determinant expansion:

$$\mathbf{P} \times \mathbf{Q} = \begin{pmatrix} \left|\begin{pmatrix} 1 & 3 \\ -1 & 3 \end{pmatrix}\right| \\ -\left|\begin{pmatrix} 4 & 3 \\ -1 & 3 \end{pmatrix}\right| \\ \left|\begin{pmatrix} 4 & 1 \\ 0 & 3 \end{pmatrix}\right| \end{pmatrix}$$

$$= \begin{pmatrix} (1)(3) - (3)(-1) \\ -((4)(3) - (3)(-1)) \\ (4)(3) - (1)(0) \end{pmatrix}$$

$$= \begin{pmatrix} 6 \\ -12 \\ 12 \end{pmatrix}$$

Solution

The area is half the magnitude of this vector:

$$
\begin{aligned}
\text{Area} &= \tfrac{1}{2}\|\mathbf{P} \times \mathbf{Q}\| \\
&= \frac{1}{2}\sqrt{6^2 + (-12)^2 + 12^2} \\
&= \frac{1}{2}\sqrt{36 + 144 + 144} \\
&= \frac{1}{2}\sqrt{324} \\
&= \frac{1}{2}(18) \\
&= 9
\end{aligned}
$$

Thus, the area of the quadrilateral is 9 square units.

C Code

```c
// quad_area.c
#include <math.h>

double quad_area(double Px, double Py, double Pz, double Qx,
    double Qy, double Qz) {
    // cross product components
    double cx = Py*Qz - Pz*Qy;
    double cy = Pz*Qx - Px*Qz;
    double cz = Px*Qy - Py*Qx;

    // magnitude of cross product
    double magnitude = sqrt(cx*cx + cy*cy + cz*cz);

    // area = half the magnitude
    return 0.5 * magnitude;
}
```

Python+C Code

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

# Load shared library
lib = ctypes.CDLL("./quad_area.so") # use quad_area.dll on
    Windows

# Function signature
lib.quad_area.restype = ctypes.c_double
lib.quad_area.argtypes = [ctypes.c_double, ctypes.c_double,
    ctypes.c_double,
                        ctypes.c_double, ctypes.c_double, ctypes.
                            c_double]
```

Python+C Code

```python

# Points A, B, C, D
A = np.array([0, 4, 1])
B = np.array([2, 3, -1])
C = np.array([4, 5, 0])
D = np.array([2, 6, 2])

# Diagonals
P = C - A
Q = D - B

# Call C function
area = lib.quad_area(P[0], P[1], P[2], Q[0], Q[1], Q[2])
print("Area (from C via ctypes) =", area)
```

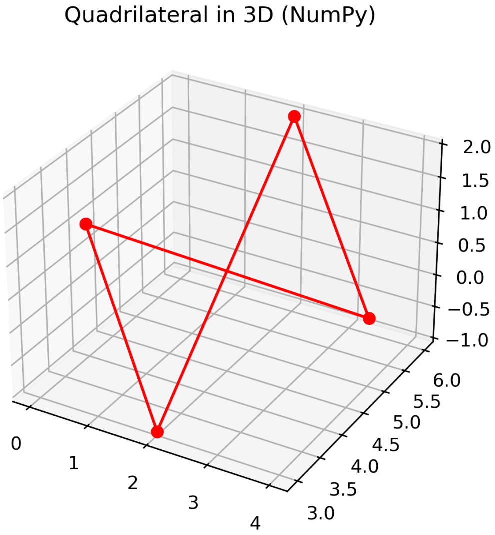
Python+C Code

```python
# --- Plot quadrilateral ---
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")

# Connect vertices in order A-B-D-C-A
X = [A[0], B[0], D[0], C[0], A[0]]
Y = [A[1], B[1], D[1], C[1], A[1]]
Z = [A[2], B[2], D[2], C[2], A[2]]

ax.plot(X, Y, Z, 'b-', marker='o')
ax.set_title("Quadrilateral in 3D (ctypes + C)")
```

Python+C Code

```python


# Save before show
plt.savefig("/storage/emulated/0/matrix/Matgeo/1.11.14/figs/Figure_1.png", dpi=300, bbox_inches='tight')
plt.show()
```

Plot


Quadrilateral in 3D (NumPy)
0
1
2
3
4
3.0
3.5
4.0
4.5
5.0
5.5
6.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Python Code

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

# Points A, B, C, D
A = np.array([0, 4, 1])
B = np.array([2, 3, -1])
C = np.array([4, 5, 0])
D = np.array([2, 6, 2])

# Diagonals
P = C - A
Q = D - B

# Cross product & area
cross = np.cross(P, Q)
area = 0.5 * np.linalg.norm(cross)
print("Area (NumPy) =", area)
```

Python Code

```
# --- Plot quadrilateral ---
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")

X = [A[0], B[0], D[0], C[0], A[0]]
Y = [A[1], B[1], D[1], C[1], A[1]]
Z = [A[2], B[2], D[2], C[2], A[2]]

ax.plot(X, Y, Z, 'r-', marker='o')
ax.set_title("Quadrilateral in 3D (NumPy)")
```

Python Code

```
    # Save before show
plt.savefig("/storage/emulated/0/matrix/Matgeo/1.11.14/figs/Figure_1.png", dpi=300, bbox_inches='tight')
plt.show()
```