Matgeo Presentation - Problem 4.7.45

Revanth Siva Kumar.D – EE25BTECH11048

October 1, 2025

QUESTION

For any three vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$, prove or disprove

$$(\mathbf{a} - \mathbf{b}) \cdot \big((\mathbf{b} - \mathbf{c}) \times (\mathbf{c} - \mathbf{a})\big) = 2\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}).$$

Solution:

We write the scalar triple product in determinant form:

$$(\mathbf{a}-\mathbf{b})\cdot\big((\mathbf{b}-\mathbf{c})\times(\mathbf{c}-\mathbf{a})\big) = \det\begin{pmatrix}(\mathbf{a}-\mathbf{b})^T\\(\mathbf{b}-\mathbf{c})^T\\(\mathbf{c}-\mathbf{a})^T\end{pmatrix}. \tag{0.1}$$

Now observe that

$$(\mathbf{a}-\mathbf{b})+(\mathbf{b}-\mathbf{c})+(\mathbf{c}-\mathbf{a}) = \mathbf{0}. \tag{0.2}$$

Thus the three rows of the determinant are linearly dependent. From matrix theory, the determinant of a matrix with linearly dependent rows is zero. Hence

$$(\mathbf{a}-\mathbf{b})\cdot\big((\mathbf{b}-\mathbf{c})\times(\mathbf{c}-\mathbf{a})\big) = 0. \tag{0.3}$$

Solution:

On the other hand, the right-hand side is

$$2\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = 2 \det \begin{pmatrix} \mathbf{a}^T \\ \mathbf{b}^T \\ \mathbf{c}^T \end{pmatrix}, \tag{0.4}$$

which is not identically zero for arbitrary $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

Conclusion

The given statement is **false**.
The left-hand side $(\mathbf{a}-\mathbf{b})\cdot\big((\mathbf{b}-\mathbf{c})\times(\mathbf{c}-\mathbf{a})\big)$is always zero, while the right-hand side $2\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})$ can be nonzero.

C Code

```
#include <stdio.h>

// Cross product u  v
void cross(double u[3], double v[3], double result[3]) {
    result[0] = u[1]*v[2] - u[2]*v[1];
    result[1] = u[2]*v[0] - u[0]*v[2];
    result[2] = u[0]*v[1] - u[1]*v[0];
}

// Dot product u  v
double dot(double u[3], double v[3]) {
    return u[0]*v[0] + u[1]*v[1] + u[2]*v[2];
}
```

C Code

```
// Compute lhs (always 0) and rhs
void compute(double a[3], double b[3], double c[3], double* lhs, double* rhs) {
    double cross_b_c[3];

    // b  c
    cross(b, c, cross_b_c);

    // LHS always 0 due to linear dependence
    *lhs = 0.0;

    // RHS = 2a  (b  c)
    *rhs = 2 * dot(a, cross_b_c);
}
```

callc.py Python code

```
import ctypes

# Load shared library
lib = ctypes.CDLL("./points.so")

# Argument and return types
lib.compute.argtypes = [
    ctypes.POINTER(ctypes.c_double), # a
    ctypes.POINTER(ctypes.c_double), # b
    ctypes.POINTER(ctypes.c_double), # c
    ctypes.POINTER(ctypes.c_double), # lhs
    ctypes.POINTER(ctypes.c_double) # rhs
]

def compute(a, b, c):
    a_arr = (ctypes.c_double*3)(*a)
    b_arr = (ctypes.c_double*3)(*b)
    c_arr = (ctypes.c_double*3)(*c)
    lhs = ctypes.c_double()
    rhs = ctypes.c_double()
    lib.compute(a_arr, b_arr, c_arr, ctypes.byref(lhs), ctypes.byref(rhs))
    return lhs.value, rhs.value
```

callc.py Python code

```
if __name__ == "__main__":
    # Input vectors
    a = list(map(float, input("Enter vector a (3 values): ").split()))
    b = list(map(float, input("Enter vector b (3 values): ").split()))
    c = list(map(float, input("Enter vector c (3 values): ").split()))

    lhs, rhs = compute(a, b, c)

    print(f"LHS = {lhs:.4f}")
    print(f"RHS = {rhs:.4f}")

    if abs(lhs - rhs) < 1e-6:
        print(" The equality holds (both sides are zero).")
    else:
        print(" The equality does NOT hold.")
```