Matgeo Presentation - Problem 2.7.33

ee25btech11021 - Dhanush sagar

September 6, 2025

Problem Statement

Find the value of p if

$$(2\hat{i} + 6\hat{j} + 27\hat{k}) \times (\hat{i} + 3\hat{j} + p\hat{k}) = 0.$$

solution

Solution:

The given vectors are

$$\mathbf{A} = \begin{pmatrix} 2 \\ 6 \\ 27 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 1 \\ 3 \\ p \end{pmatrix}. \tag{0.1}$$

Construct the matrix

$$M = \begin{pmatrix} 2 & 6 & 27 \\ 1 & 3 & p \end{pmatrix}. \tag{0.2}$$

If $\mathbf{A} \times \mathbf{B} = 0$, then $\mathbf{A}$ and $\mathbf{B}$ are linearly dependent. Thus,

$$\text{rank}(M) < 2. \tag{0.3}$$

For a 2×3 matrix, this happens exactly when all 2×2 minors vanish.

First minor:

$$\det \begin{pmatrix} 2 & 6 \\ 1 & 3 \end{pmatrix} = 6 - 6 = 0. \tag{0.4}$$

solution

Second minor:

$$\det \begin{pmatrix} 2 & 27 \\ 1 & p \end{pmatrix} = 2p - 27. \tag{0.5}$$

Third minor:

$$\det \begin{pmatrix} 6 & 27 \\ 3 & p \end{pmatrix} = 6p - 81. \tag{0.6}$$

For $\text{rank}(M) < 2$, all three determinants must vanish. The first is already zero. From the second,

$$2p - 27 = 0 \Rightarrow p = \tfrac{27}{2}. \tag{0.7}$$

From the third,

$$6p - 81 = 0 \Rightarrow p = \tfrac{27}{2}. \tag{0.8}$$

Thus, the required value is

$$\boxed{p = \tfrac{27}{2}} \tag{0.9}$$

C Source Code: cross.c

```
#include <stdio.h>
void cross_product(double a[3], double b[3], double result[3])
    result[0] = a[1]*b[2] - a[2]*b[1];
    result[1] = a[2]*b[0] - a[0]*b[2];
    result[2] = a[0]*b[1] - a[1]*b[0];
}
double find_p() {
    double a[3] = {2, 6, 27};
    double p = 27.0 / 2.0;
    double b[3] = {1, 3, p};
    double res[3];
    cross_product(a, b, res);
   if(res[0] == 0 && res[1] == 0 && res[2] == 0) {
        return p;
    }
    return -1;
}
```

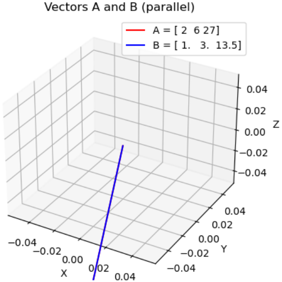
Python Script: vector solve.py

```
import ctypes
import numpy as np
# Load shared library
lib = ctypes.CDLL("./cross.so")
lib.find_p.restype = ctypes.c_double
# Call C function
p = lib.find_p()
print("Computed value of p:", p)
# Verify using numpy
A = np.array([2, 6, 27])
B = np.array([1, 3, p])
cross_prod = np.cross(A, B)
print("Cross product A × B =", cross_prod)
if np.allclose(cross_prod, [0, 0, 0]):
    print("✓ A and B are parallel. Solution verified.")
else:
    print("× Something went wrong.")
```

Python Script: plot vector.py

```
import numpy as np
import matplotlib.pyplot as plt
# Vectors
A = np.array([2, 6, 27])
p = 27/2
B = np.array([1, 3, p])
# Plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.quiver(0, 0, 0, A[0], A[1], A[2], color='r', label=f'A = {
ax.quiver(0, 0, 0, B[0], B[1], B[2], color='b', label=f'B = {
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.legend()
plt.title("Vectors A and B (parallel)")
plt.savefig("vectors.png")                    plt.show()
```

Result Plot



Vectors A and B (parallel)
A = [2 6 27]
B = [1.   3. 13.5]
0.04
0.02
0.00
−0.02
−0.04
Z
0.04
0.02
0.00
−0.02
−0.04
Y
−0.04
−0.02
0.00
0.02
0.04
X