Matgeo Presentation - Problem 1.6.19

ai25btech11004 - Jaswanth

September 2, 2025

Question

The vectors $\lambda\hat{i}+\lambda\hat{j}+2\hat{k}$,$1\hat{i}+\lambda\hat{j}-1\hat{k}$ and $2\hat{i}-1\hat{j}+\lambda\hat{k}$ are coplanar if $\lambda =$

Description

Name	**vector**
vector**A**	$\begin{pmatrix} \lambda \\ \lambda \\ 2 \end{pmatrix}$
vector**B**	$\begin{pmatrix} 1 \\ \lambda \\ -1 \end{pmatrix}$
vector**C**	$\begin{pmatrix} 2 \\ -1 \\ \lambda \end{pmatrix}$

Table: variables used

Solution

Form the 3×3 matrix whose columns are the given vectors:

$$A = \begin{pmatrix} \lambda & 1 & 2 \\ \lambda & \lambda & -1 \\ 2 & -1 & \lambda \end{pmatrix}.$$

The three vectors are coplanar if the columns are linearly dependent, i.e. if there exists a nonzero vector $u = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ with $Au = 0$. Writing $Au = 0$ gives the system

$$\begin{cases} \lambda x + y + 2z = 0, \\ \lambda x + \lambda y - z = 0, \\ 2x - y + \lambda z = 0. \end{cases}$$

Subtract the first equation from the second to eliminate x:

$$(\lambda x + \lambda y - z) - (\lambda x + y + 2z) = 0 \implies (\lambda - 1)y - 3z = 0 \implies z = \frac{\lambda - 1}{3}$$

Solution

Substitute this z into the first equation to express x in terms of y:

$$\lambda x + y + 2\left(\frac{\lambda - 1}{3} y\right) = 0 \implies \lambda x + \frac{2\lambda + 1}{3}\, y = 0 \implies x = -\frac{2\lambda + 1}{3\lambda}\, y,$$

(valid when $\lambda \neq 0$; the case $\lambda = 0$ is checked separately below).
Now substitute x and z (both expressed in terms of y) into the third equation:

$$2x - y + \lambda z = 0.$$

Using $x = -\dfrac{2\lambda + 1}{3\lambda} y$ and $z = \dfrac{\lambda - 1}{3} y$ we get

$$-\frac{4\lambda + 2}{3\lambda}\, y - y + \frac{\lambda(\lambda - 1)}{3}\, y = 0.$$

Multiply through by 3λ and factor y:

$$y\left(\lambda^3 - \lambda^2 - 7\lambda - 2\right) = 0.$$

A nontrivial solution requires $y \neq 0$, hence

$$\lambda^3 - \lambda^2 - 7\lambda - 2 = 0.$$

Solution

Factor the cubic. One checks $\lambda = -2$ is a root, and polynomial division yields

$$\lambda^3 - \lambda^2 - 7\lambda - 2 = (\lambda + 2)(\lambda^2 - 3\lambda - 1).$$

The quadratic factor has roots

$$\lambda = \frac{3 \pm \sqrt{9+4}}{2} = \frac{3 \pm \sqrt{13}}{2}.$$

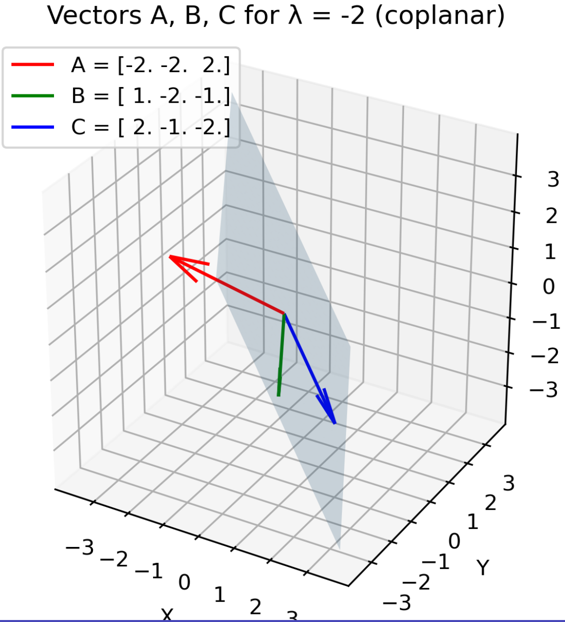
Finally check the special case $\lambda = 0$: the system becomes

$$\begin{cases} y + 2z = 0, \\ -z = 0, \\ 2x - y = 0, \end{cases}$$

which forces $x = y = z = 0$, so $\lambda = 0$ does *not* give a nontrivial solution.

Therefore the vectors are coplanar exactly for $\lambda = -2,\ \dfrac{3+\sqrt{13}}{2},\ \dfrac{3-\sqrt{13}}{2}$.

Plot



C Code: Vector.c

```
#include <stdio.h>

int main() {
    FILE *fp;
    fp = fopen("vector.dat", "w");
    if (fp == NULL) {
        printf("Error␣opening␣file!\n");
        return 1;
    }

    // The determinant expansion:
    // |   2 |
    // | 1  -1 |
    // | 2 -1   |
    //
    // Det = -^3 + ^2 + 5 - 4

    fprintf(fp, "Determinant␣condition␣for␣coplanarity:\n");
    fprintf(fp, "(-^3␣+␣^2␣+␣5␣-␣4)␣=␣0\n\n");

    fprintf(fp, "Checking␣integer␣values␣of␣␣from␣-10␣to␣10:\n");

    for (int lambda = -10; lambda <= 10; lambda++) {
        int val = -lambda*lambda*lambda + lambda*lambda + 5*lambda - 4;
        if (val == 0) {
            fprintf(fp, "␣=␣%d␣is␣a␣solution.\n", lambda);
        }
    }

    fclose(fp);
    printf("Results␣written␣to␣vector.dat\n");
    return 0;
}
```

Python: plot.py

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

# ----- Correct value -----
lam = -2 # correct

# Vectors
A = np.array([lam, lam, 2], dtype=float)
B = np.array([1, lam, -1], dtype=float)
C = np.array([2, -1, lam], dtype=float)

# Verify coplanarity via scalar triple product: A  (B  C) = 0
triple = float(np.dot(A, np.cross(B, C)))
print(f"Scalar triple product at ={lam}: {triple:.6g} (0 => coplanar)")

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

origin = np.zeros(3)

# Plot vectors from origin
ax.quiver(*origin, *A, length=1, normalize=False, label=f"A = {A}", color='r')
ax.quiver(*origin, *B, length=1, normalize=False, label=f"B = {B}", color='g')
ax.quiver(*origin, *C, length=1, normalize=False, label=f"C = {C}", color='b')

# Plot the plane spanned by B and C (shows A lies in this plane)
s = np.linspace(-1.2, 1.2, 20)
t = np.linspace(-1.2, 1.2, 20)
S, T = np.meshgrid(s, t)
plane = np.outer(S.ravel(), B) + np.outer(T.ravel(), C)
X = plane[:, 0].reshape(S.shape)
```

Python: plot.py

```
Y = plane[:, 1].reshape(S.shape)
Z = plane[:, 2].reshape(S.shape)
ax.plot_surface(X, Y, Z, alpha=0.2, edgecolor='none')

# Aesthetic: equal aspect & limits
all_pts = np.vstack([origin, A, B, C, plane])
mins = all_pts.min(axis=0)
maxs = all_pts.max(axis=0)
ranges = maxs - mins
center = (maxs + mins) / 2
max_range = ranges.max() * 0.55 + 1e-9
ax.set_xlim(center[0]-max_range, center[0]+max_range)
ax.set_ylim(center[1]-max_range, center[1]+max_range)
ax.set_zlim(center[2]-max_range, center[2]+max_range)
ax.set_box_aspect([1,1,1])

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title(f"Vectors␣A,␣B,␣C␣for␣␣=␣{lam}␣(coplanar)")

ax.legend(loc='upper␣left')

# ----- Save the figure -----
plt.savefig("vectors.png", dpi=300, bbox_inches='tight')

# Show on screen too (optional)
plt.show()
```