# Matgeo Presentation - Problem 1.6.19

ai25btech11004 - Jaswanth

September 9, 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 $\lambda =$

## Solution

The vectors are coplanar $\iff$ they are linearly dependent. Form the matrix with these vectors as columns:

$$A = \begin{bmatrix} \lambda & 1 & 2 \\ \lambda & \lambda & -1 \\ 2 & -1 & \lambda \end{bmatrix}. \tag{0.1}$$

The three vectors are linearly dependent $\iff$ $\det(A) = 0$. We compute $\det(A)$ using row reduction.

$$A = \begin{bmatrix} \lambda & 1 & 2 \\ \lambda & \lambda & -1 \\ 2 & -1 & \lambda \end{bmatrix} \tag{0.2}$$

$$R_2 \to R_2 - R_1 \quad \Rightarrow \quad \begin{bmatrix} \lambda & 1 & 2 \\ 0 & \lambda - 1 & -3 \\ 2 & -1 & \lambda \end{bmatrix} \tag{0.3}$$

# Solution

$$R_3 \to R_3 - \tfrac{2}{\lambda} R_1 \quad \Rightarrow \quad \begin{bmatrix} \lambda & 1 & 2 \\ 0 & \lambda - 1 & -3 \\ 0 & -1 - \frac{2}{\lambda} & \lambda - \frac{4}{\lambda} \end{bmatrix} \tag{0.4}$$

$$R_3 \to R_3 - \frac{-1 - \frac{2}{\lambda}}{\lambda - 1} R_2 \quad \Rightarrow \quad \begin{bmatrix} \lambda & 1 & 2 \\ 0 & \lambda - 1 & -3 \\ 0 & 0 & \dfrac{\lambda^3 - \lambda^2 - 7\lambda - 2}{\lambda(\lambda - 1)} \end{bmatrix} \tag{0.5}$$

Now the matrix is upper triangular, so

$$\det(A) = \lambda(\lambda - 1) \cdot \frac{\lambda^3 - \lambda^2 - 7\lambda - 2}{\lambda(\lambda - 1)} = \lambda^3 - \lambda^2 - 7\lambda - 2. \tag{0.6}$$

**Step 2: Solve the cubic**

$$\lambda^3 - \lambda^2 - 7\lambda - 2 = 0. \tag{0.7}$$

# Solution

Factor:

$$(\lambda + 2)(\lambda^2 - 3\lambda - 1) = 0. \tag{0.8}$$

So the solutions are

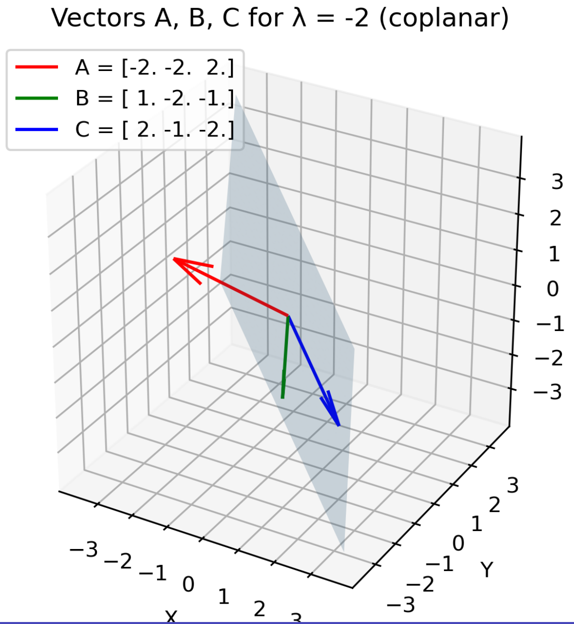
$$\lambda = -2, \qquad \lambda = \frac{3 + \sqrt{13}}{2}, \qquad \lambda = \frac{3 - \sqrt{13}}{2}. \tag{0.9}$$

**Conclusion:**

For these values of $\lambda$, $\det(A) = 0 \implies \operatorname{rank}(A) < 3$, so the vectors are **linearly dependent** (coplanar).

$\boxed{\text{The vectors are coplanar for } \lambda = -2,\ \frac{3+\sqrt{13}}{2},\ \frac{3-\sqrt{13}}{2}.}$

# Plot


Vectors A, B, C for λ = -2 (coplanar)
A = [-2. -2.  2.]
B = [ 1. -2. -1.]
C = [ 2. -1. -2.]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
X
3
2
1
0
−1
−2
−3
Y

# 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()
```