# Problem 5.13.49

ee25btech11023-Venkata Sai

September 16, 2025

1. Problem
2. Solution
   - Matrix Equation
   - Conclusion
   - Plot
3. C Code
4. Python Code

# Problem

If the system of equations

$$x - ky - z = 0 \tag{1.1}$$

$$kx - y - z = 0 \tag{1.2}$$

$$x + y + z = 0 \tag{1.3}$$

has a non-zero solution, then the possible values of $k$ are

# Matrix Equation

For the given homogeneous system

$$\mathbf{Ax} = 0 \tag{2.1}$$

Augmented matrix of $\left(\mathbf{A} \mid 0\right)$ is given by

$$\left(\begin{array}{ccc|c} 1 & -k & -1 & 0 \\ k & -1 & -1 & 0 \\ 1 & 1 & -1 & 0 \end{array}\right) \xrightarrow[R_3 \to R_3 - R_1]{R_2 \to R_2 - kR_1} \left(\begin{array}{ccc|c} 1 & -k & -1 & 0 \\ 0 & k^2-1 & k-1 & 0 \\ 0 & 1+k & 0 & 0 \end{array}\right) \tag{2.2}$$

$$\left(\begin{array}{ccc|c} 1 & -k & -1 & 0 \\ 0 & k^2-1 & k-1 & 0 \\ 0 & 1+k & 0 & 0 \end{array}\right) \xrightarrow{R_2 - (k-1)R_3} \left(\begin{array}{ccc|c} 1 & -k & -1 & 0 \\ 0 & k+1 & 0 & 0 \\ 0 & 0 & k-1 & 0 \end{array}\right) \tag{2.3}$$

# Conclusion

For a non-zero solution,The rank of the matrix must be less than the number of variables
From (5), In order to be Rank$<3$

$$k+1=0 \ \text{(or)} \ k-1=0 \tag{2.4}$$

$$k=-1 \ \text{(or)} \ k=1 \tag{2.5}$$

# Plot

figs/fig1.png

# C Code

```c
void get_system_coeffs(double* out_coeffs) {

    out_coeffs[0] = 1.0;
    out_coeffs[1] = -1.0;

    out_coeffs[2] = -1.0;
    out_coeffs[3] = -1.0;


    out_coeffs[4] = 1.0;
    out_coeffs[5] = 1.0;
    out_coeffs[6] = 1.0;
}
```

# Python Code for Solving

```python
import ctypes
import sympy

lib = ctypes.CDLL('./code.so')

double_array_7 = ctypes.c_double * 7
lib.get_system_coeffs.argtypes = [ctypes.POINTER(ctypes.c_double)]

out_data_c = double_array_7()
lib.get_system_coeffs(out_data_c)


coeffs = list(int(v) for v in out_data_c)

k = sympy.Symbol('k')
```

# Python Code for Solving

```
M = sympy.Matrix([
        [coeffs[0], -k, coeffs[1]],
        [k, coeffs[2], coeffs[3]],
        [coeffs[4], coeffs[5], coeffs[6]]
    ])

    # Calculate the determinant of the matrix in terms of k
det_M = M.det()
print(f"\nDeterminant = {det_M}")


solutions = sympy.solve(det_M, k)

print(f"k can be {solutions}")
```