MatGeo Assignment 5.3.1

AI25BTECH11007

September 27, 2025

Question

For what value of k, the system of linear equations $x + y + z = 2$
$2x + y - z = 3$
$3x + 2y + kz = 4$ has a unique solution?

Solution

$$\text{System: } \begin{cases} x + y + z = 2 \\ 2x + y - z = 3 \\ 3x + 2y + kz = 4 \end{cases}$$

$$\mathbf{A} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 3 \\ 2 \\ k \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}$$

$$\text{Augmented matrix(M)} : \begin{pmatrix} A & B & C & D \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 2 \\ 2 & 1 & -1 & 3 \\ 3 & 2 & k & 4 \end{pmatrix}$$

by row reducing,

$$R_2 \to R_2 - 2R_1, \quad R_3 \to R_3 - 3R_1, \quad R_3 \to R_3 - R_2$$

$$\begin{pmatrix} 1 & 1 & 1 & 2 \\ 0 & -1 & -3 & -1 \\ 0 & 0 & k & -1 \end{pmatrix}$$

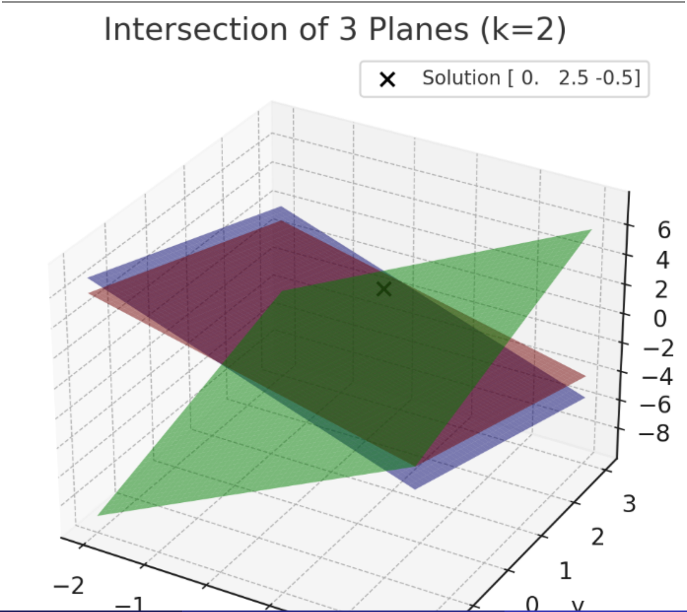
- **If** $k \neq 0$**:** the augmented matrix has three non-zero rows
  , so $\text{rank}(M) = 3$.
  hence, Unique Solution for system
- **If** $k = 0$**:** the row-echelon form becomes

$$\left[\begin{pmatrix} 1 & 1 & 1 \\ 0 & -1 & -3 \\ 0 & 0 & 0 \end{pmatrix} \middle| \begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix} \right].$$

Here $\text{rank}(M) = 2$
so, system is Inconsistent; No Solution.

$$\boxed{\text{Therefore, the system has a unique solution precisely when } k \neq 0.}$$

Plot


Intersection of 3 Planes (k=2)
Solution [0.   2.5 -0.5]
6
4
2
0
−2
−4
−6
−8
3
2
1
0
y
−2
−1

C code

```c
    #include <stdio.h>

int main() {
    double A[3][4]; // augmented matrix [3x4]
    double k;

    // Input value of k
    printf("Enter the value of k: ");
    scanf("%lf", &k);

    // Initialize augmented matrix
    A[0][0] = 1; A[0][1] = 1; A[0][2] = 1; A[0][3] = 2;
    A[1][0] = 2; A[1][1] = 1; A[1][2] = -1; A[1][3] = 3;
    A[2][0] = 3; A[2][1] = 2; A[2][2] = k; A[2][3] = 4;

    // Row reduction: R2 = R2 - 2*R1
    for(int j=0; j<4; j++)
        A[1][j] -= 2*A[0][j];
```

C code

```c
// Row reduction: R3 = R3 - 3*R1
for(int j=0; j<4; j++)
    A[2][j] -= 3*A[0][j];
// Row reduction: R3 = R3 - R2
for(int j=0; j<4; j++)
    A[2][j] -= A[1][j];
// Check k for uniqueness
if (A[2][2] != 0) {
    printf("The system has a unique solution.\n");
    double z = A[2][3] / A[2][2];
    double y = (A[1][3] - A[1][2]*z) / A[1][1];
    double x = A[0][3] - A[0][1]*y - A[0][2]*z;

    printf("Solution: x = %.2lf, y = %.2lf, z = %.2lf\n", x,
        y, z);
} else {
    printf("The system has no solution (inconsistent).\n");
}
return 0; }
```

Python code

```
    import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Define k (nonzero)
k = 2

# Planes equations: coefficients and RHS
planes = [
    (np.array([1, 1, 1]), 2), # x + y + z = 2
    (np.array([2, 1, -1]), 3), # 2x + y - z = 3
    (np.array([3, 2, k]), 4) # 3x + 2y + k z = 4
]

# Solve for intersection point
A = np.array([p[0] for p in planes])
b = np.array([p[1] for p in planes])
solution = np.linalg.solve(A, b)
```

Python code

```
    # Create grid for plotting planes
xx, yy = np.meshgrid(np.linspace(-2, 3, 50), np.linspace(-2, 3,
    50))

# Calculate corresponding z for each plane
zz1 = (2 - xx - yy)
zz2 = (3 - 2*xx - yy) * -1
zz3 = (4 - 3*xx - 2*yy) / k

# Plot planes and intersection point
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(xx, yy, zz1, alpha=0.5, color='red')
ax.plot_surface(xx, yy, zz2, alpha=0.5, color='green')
ax.plot_surface(xx, yy, zz3, alpha=0.5, color='blue')
```

python code

```
        # Plot solution point
ax.scatter(*solution, color='black', s=50, label=f"Solution {
    solution.round(2)}")

ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
ax.legend()ame}
```