5.2.49

RAVULA SHASHANK REDDY - EE25BTECH11047

September 30, 2025

Question

Solve the system of equations using matrices:

$$\begin{aligned} 3x - y - 2z &= 2, \\ 2y - z &= -1, \\ 3x - 5y &= 3. \end{aligned}$$

Solution

Given:

$$\begin{pmatrix} 3 \\ -1 \\ -2 \end{pmatrix}^T \mathbf{x} = 2, \tag{1}$$

$$\begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix}^T \mathbf{x} = -1, \tag{2}$$

$$\begin{pmatrix} 3 \\ -5 \\ 0 \end{pmatrix}^T \mathbf{x} = 3 \tag{3}$$

$$\begin{pmatrix} 3 & -1 & -2 \\ 0 & 2 & -1 \\ 3 & -5 & 0 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix} \tag{4}$$

Solution

$$R_3 \to R_3 - R_1 \Rightarrow \begin{pmatrix} 3 & -1 & -2 & 2 \\ 0 & 2 & -1 & -1 \\ 0 & -4 & 2 & 1 \end{pmatrix} \tag{5}$$

$$R_2 \to \frac{1}{2}R_2 \Rightarrow \begin{pmatrix} 3 & -1 & -2 & 2 \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -4 & 2 & 1 \end{pmatrix} \tag{6}$$

$$R_1 \to R_1 + R_2, \quad R_3 \to R_3 + 4R_2 \Rightarrow \begin{pmatrix} 3 & 0 & -\frac{5}{2} & \frac{3}{2} \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{7}$$

$$\implies 0 = -1 \tag{8}$$

$$\text{System inconsistent} \quad \Rightarrow \quad \boxed{\text{No solution}}$$

C Code

```c
    #include <stdio.h>
#include <math.h>

#define N 3

int main() {
    int i, j, k;
    double A[N][N+1] = {
        {3, -1, -2, 2},
        {0, 2, -1, -1},
        {3, -5, 0, 3}
    };

    // Forward elimination
    for (i = 0; i < N; i++) {
        // Pivot should not be zero (no pivoting added here for
            simplicity)
        if (fabs(A[i][i]) < 1e-9) continue;
```

C Code

```c
        // Normalize row
        double div = A[i][i];
        for (j = i; j <= N; j++) {
            A[i][j] /= div;
        }

        // Eliminate other rows
        for (k = 0; k < N; k++) {
            if (k == i) continue;
            double factor = A[k][i];
            for (j = i; j <= N; j++) {
                A[k][j] -= factor * A[i][j];
            }
        }
    }

    // Check for inconsistency
    int inconsistent = 0;
```

C Code

```c
    for (i = 0; i < N; i++) {
        int allZero = 1;
        for (j = 0; j < N; j++) {
            if (fabs(A[i][j]) > 1e-9) {
                allZero = 0;
                break;
            }
        }
        if (allZero && fabs(A[i][N]) > 1e-9) {
            inconsistent = 1;
            break;
        }
    }
```

C Code

```c
    if (inconsistent) {
        printf(System is inconsistent -> No solution\\n);
    } else {
        printf(Solution:\\n);
        for (i = 0; i < N; i++) {
            printf(x%d = %lf\\n, i+1, A[i][N]);
        }
    }

    return 0;
}
```

Python Direct

```
import numpy as np
import matplotlib.pyplot as plt

# local imports as you asked
import libs.line.funcs as linefuncs
import libs.triangle.funcs as trifuncs

# Coefficients of the system Ax=b
A = np.array([
    [3, -1, -2],
    [0, 2, -1],
    [3, -5, 0]
], dtype=float)

b = np.array([2, -1, 3], dtype=float)

# Check consistency via rank
rankA = np.linalg.matrix_rank(A)
```

Python Direct

```
rankAb = np.linalg.matrix_rank(np.c_[A, b])

print(Rank(A) =, rankA)
print(Rank([A|b]) =, rankAb)

if rankA != rankAb:
    print(System is inconsistent -> No solution)

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

Python Direct

```
# Create a meshgrid for x,y
x_vals = np.linspace(-5, 5, 50)
y_vals = np.linspace(-5, 5, 50)
X, Y = np.meshgrid(x_vals, y_vals)

# Equation 1: 3x - y - 2z = 2 -> z = (3x - y - 2)/2
Z1 = (3*X - Y - 2)/2

# Equation 2: 2y - z = -1 -> z = 2y + 1
Z2 = 2*Y + 1

# Equation 3: 3x - 5y = 3 -> (no z term, it's a vertical plane)
# So plot separately
Z3 = np.linspace(-5, 5, 50)
X3, Z3m = np.meshgrid(x_vals, Z3)
Y3 = (3*X3 - 3)/5
```

Python Direct

```
# Plot the planes
ax.plot_surface(X, Y, Z1, alpha=0.5, color='red', label='Plane 1'
    )
ax.plot_surface(X, Y, Z2, alpha=0.5, color='blue', label='Plane 2
    ')
ax.plot_surface(X3, Y3, Z3m, alpha=0.5, color='green', label='
    Plane 3')

# Labels
ax.set_xlabel(x)
ax.set_ylabel(y)
ax.set_zlabel(z)
ax.set_title(Three Planes (Inconsistent System))

plt.show()
```

Python Shared

```python
import ctypes
import numpy as np

# Load the shared library
lib = ctypes.CDLL(./libgauss.so)

# Function signature: int solve_system(double *A_in, double *
    x_out)
lib.solve_system.argtypes = [ctypes.POINTER(ctypes.c_double),
                             ctypes.POINTER(ctypes.c_double)]
lib.solve_system.restype = ctypes.c_int

# Input augmented matrix [A|b]
A = np.array([
    [3, -1, -2, 2], # 3x - y - 2z = 2
    [0, 2, -1, -1], # 2y - z = -1
    [3, -5, 0, 3] # 3x - 5y = 3
], dtype=np.float64)
```

Python Shared

```python
A_flat = A.flatten()
x_out = np.zeros(3, dtype=np.float64)

# Call the C function
status = lib.solve_system(A_flat.ctypes.data_as(ctypes.POINTER(
    ctypes.c_double)),
                              x_out.ctypes.data_as(ctypes.POINTER(
                                  ctypes.c_double)))

if status == 1:
    print(System is inconsistent -> No solution)
else:
    print(Solution:, x_out)
```

Plot

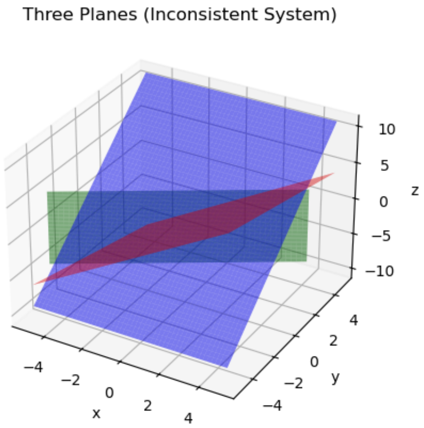

Figure: