5.5.12

Namaswi-EE25BTECH11060

september 14,2025

Question

If $\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 2 \\ 3 & 1 & 1 \end{pmatrix}$ find $\mathbf{A}^{-1}$. Hence solve the following system of equations

$$x + y + z = 6$$
$$x + 2z = 7$$
$$3x + y + z = 12$$

Solution

$$\mathbf{Ax} = \mathbf{I} \tag{1}$$

Forming Argumented Matrix

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 0 & 1 \end{array}\right) \tag{2}$$

Solution

Replace $R2 \to R2 - R1$ and $R3 \to R3 - 3R1$

$$\left(\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 1 & 0 \\ 0 & -2 & -2 & -3 & 0 & 1 \end{array}\right) \tag{3}$$

Replace $R2 \to -R2$ and $R3 \to R3 + 2R2$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 & -1 & 0 \\ 0 & 0 & -4 & -1 & -2 & 1 \end{array}\right) \tag{4}$$

Solution

Repalce $R_3 \leftarrow -\frac{1}{4}R_3$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & -1 & 1 & -1 & 0 \\ 0 & 0 & 1 & \frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{array}\right) \tag{5}$$

Replace $R_1 \leftarrow R_1 - 2R_3,\ R_2 \leftarrow R_2 + R_3$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 & \frac{5}{4} & -\frac{1}{2} & -\frac{1}{4} \\ 0 & 0 & 1 & \frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{array}\right) \tag{6}$$

Solution

Thus

$$A^{-1} = \begin{pmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{5}{4} & -\frac{1}{2} & -\frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{pmatrix}. \tag{7}$$

Solution

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{b} \tag{8}$$

$$\mathbf{X} = \begin{pmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{5}{4} & -\frac{1}{2} & -\frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{pmatrix} \begin{pmatrix} 6 \\ 7 \\ 12 \end{pmatrix} \tag{9}$$

$$\mathbf{X} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix} \tag{10}$$

Python Code

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

# Define the grid
x = np.linspace(-1, 10, 30)
y = np.linspace(-1, 10, 30)
X, Y = np.meshgrid(x, y)
```

Python Code

```
        # Define the planes
Z1 = 6 - X - Y # From x + y + z = 6
Z2 = (7 - X) / 2 # From x + 2z = 7
Z3 = 12 - 3*X - Y # From 3x + y + z = 12

# Plot the surfaces
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
```

Python Code

```
ax.plot_surface(X, Y, Z1, alpha=0.5, color='red', label='x+y+z=6'
    )
ax.plot_surface(X, Y, Z2, alpha=0.5, color='green', label='x+2z=7
    ')
ax.plot_surface(X, Y, Z3, alpha=0.5, color='blue', label='3x+y+z
    =12')

# Labels
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Graph of 3 Planes')

plt.show()
```

C Code

```c
    #include <stdio.h>
#include <math.h>
#define N 3 // Number of variables
int main() {
    int i, j, k;
    float a[N][N+1], ratio;
    float x[N];

    // Augmented matrix input
    // System:
    // x + y + z = 6
    // x + 2z = 7
    // 3x + y + z = 12
    a[0][0] = 1; a[0][1] = 1; a[0][2] = 1; a[0][3] = 6;
    a[1][0] = 1; a[1][1] = 0; a[1][2] = 2; a[1][3] = 7;
    a[2][0] = 3; a[2][1] = 1; a[2][2] = 1; a[2][3] = 12;
```

C Code

```c
 // Forward Elimination
for (i = 0; i < N-1; i++) {
    if (a[i][i] == 0.0) {
        printf("Mathematical Error!\n");
        return 0;
    }
    for (j = i+1; j < N; j++) {
        ratio = a[j][i] / a[i][i];
        for (k = 0; k <= N; k++) {
            a[j][k] -= ratio * a[i][k];
        }
    }
}
```

C Code

```
    // Back Substitution
    x[N-1] = a[N-1][N] / a[N-1][N-1];
    for (i = N-2; i >= 0; i--) {
        x[i] = a[i][N];
        for (j = i+1; j < N; j++) {
            x[i] -= a[i][j] * x[j];
        }
        x[i] /= a[i][i];
    }
    // Display solution
    printf("Solution:\n");
    for (i = 0; i < N; i++) {
        printf("x[%d] = %.2f\n", i+1, x[i]);
    }
    return 0;
}
```

C and Python Code

```
    import ctypes
import numpy as np

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

# Define return type and argument types
lib.gauss_elimination.argtypes = [ctypes.POINTER(ctypes.c_double)
    ]
lib.gauss_elimination.restype = None
```

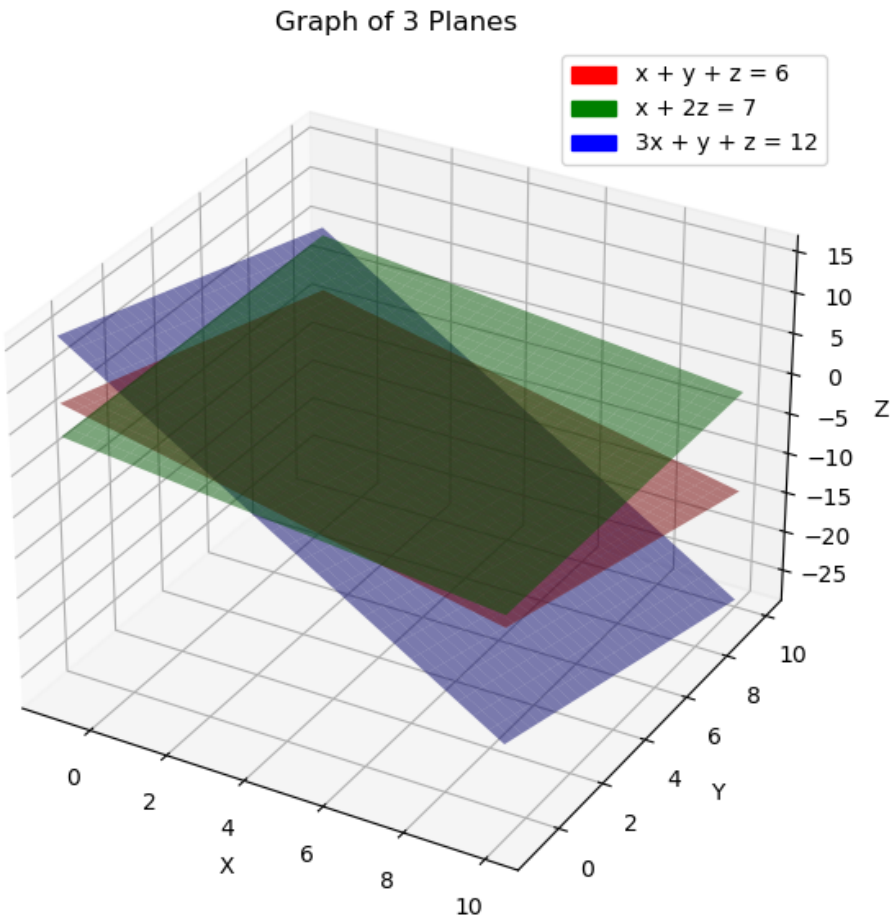
C and Python Code

```
    # Prepare array for results
res = (ctypes.c_double * 3)()

# Call the function
lib.gauss_elimination(res)

# Convert to Python list
solution = [res[i] for i in range(3)]
print("Solution (x, y, z) =", solution)
```

Plot


Graph of 3 Planes
x + y + z = 6
x + 2z = 7
3x + y + z = 12
X
Y
Z
0
2
4
6
8
10
15
10
5
0
−5
−10
−15
−20
−25