5.5.1

Sai Krishna Bakki - EE25BTECH11049

Question

If $\mathbf{A} = \begin{pmatrix} 5 & -1 & 4 \\ 2 & 3 & 5 \\ 5 & -2 & 6 \end{pmatrix}$, find $\mathbf{A}^{-1}$ and use it to solve the following system of equations

$$\begin{aligned} 5x - y + 4z &= 5 \\ 2x + 3y + 5z &= 2 \\ 5x - 2y + 6z &= -1 \end{aligned}$$

Theoretical Solution

$$\left(\begin{array}{ccc|ccc} 5 & -1 & 4 & 1 & 0 & 0 \\ 2 & 3 & 5 & 0 & 1 & 0 \\ 5 & -2 & 6 & 0 & 0 & 1 \end{array}\right) R_3 \leftarrow R_3 - R_1 \left(\begin{array}{ccc|ccc} 5 & -1 & 4 & 1 & 0 & 0 \\ 2 & 3 & 5 & 0 & 1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 1 \end{array}\right) \tag{1}$$

$$[\, R_2 \leftarrow R_2 + 3R_3] \, R_1 \leftarrow R_1 - R_3 \left(\begin{array}{ccc|ccc} 5 & 0 & 2 & 2 & 0 & -1 \\ 2 & 0 & 11 & -3 & 1 & 3 \\ 0 & -1 & 2 & -1 & 0 & 1 \end{array}\right) \tag{2}$$

$$R_3 \leftarrow -R_3 \left(\begin{array}{ccc|ccc} 5 & 0 & 2 & 2 & 0 & -1 \\ 2 & 0 & 11 & -3 & 1 & 3 \\ 0 & 1 & -2 & 1 & 0 & -1 \end{array}\right) \tag{3}$$

Theoretical Solution

$$R_2 \leftrightarrow R_3 \left(\begin{array}{ccc|ccc} 5 & 0 & 2 & 2 & 0 & -1 \\ 0 & 1 & -2 & 1 & 0 & -1 \\ 2 & 0 & 11 & -3 & 1 & 3 \end{array}\right) \tag{4}$$

$$R_1 \leftarrow \frac{1}{5}R_1 \left(\begin{array}{ccc|ccc} 1 & 0 & 2/5 & 2/5 & 0 & -1/5 \\ 0 & 1 & -2 & 1 & 0 & -1 \\ 2 & 0 & 11 & -3 & 1 & 3 \end{array}\right) \tag{5}$$

$$R_3 \leftarrow R_3 - 2R_1 \left(\begin{array}{ccc|ccc} 1 & 0 & 2/5 & 2/5 & 0 & -1/5 \\ 0 & 1 & -2 & 1 & 0 & -1 \\ 0 & 0 & 51/5 & -19/5 & 1 & 17/5 \end{array}\right) \tag{6}$$

Theoretical Solution

$$R_3 \leftarrow \frac{5}{51} R_3 \left(\begin{array}{ccc|ccc} 1 & 0 & 2/5 & 2/5 & 0 & -1/5 \\ 0 & 1 & -2 & 1 & 0 & -1 \\ 0 & 0 & 1 & -19/51 & 5/51 & 17/51 \end{array}\right) \tag{7}$$

$$[\, R_2 \leftarrow R_2 + 2R_3]\; R_1 \leftarrow R_1 - \frac{2}{5} R_3 \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 28/51 & -2/51 & -17/51 \\ 0 & 1 & 0 & 13/51 & 10/51 & -17/51 \\ 0 & 0 & 1 & -19/51 & 5/51 & 17/51 \end{array}\right) \tag{8}$$

$$\therefore \mathbf{A}^{-1} = \begin{pmatrix} 28/51 & -2/51 & -17/51 \\ 13/51 & 10/51 & -17/51 \\ -19/51 & 5/51 & 17/51 \end{pmatrix} \tag{9}$$

Theoretical Solution

Now, Finding system of equations

$$\mathbf{AX} = \mathbf{C} \tag{10}$$

where $\mathbf{C} = \begin{pmatrix} 5 \\ 2 \\ -1 \end{pmatrix}$ and $\mathbf{X} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{C} \tag{11}$$

$$\mathbf{X} = \begin{pmatrix} 28/51 & -2/51 & -17/51 \\ 13/51 & 10/51 & -17/51 \\ -19/51 & 5/51 & 17/51 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \\ -1 \end{pmatrix} \tag{12}$$

$$\therefore \mathbf{X} = \begin{pmatrix} 3 \\ 2 \\ -2 \end{pmatrix} \tag{13}$$

C Code

```
#include <stdio.h>

#define N 3 // matrix size (you can generalize)

void inverse(double A[N][N], double inv[N][N]) {
    // Step 1: Create augmented matrix [A|I]
    double aug[N][2*N];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            aug[i][j] = A[i][j]; // copy A
            aug[i][j+N] = (i == j) ? 1 : 0; // identity
        }
    }

    // Step 2: GaussJordan elimination
    for (int i = 0; i < N; i++) {
        // Make pivot = 1
        double pivot = aug[i][i];
```

C Code

```
        for (int j = 0; j < 2*N; j++) {
            aug[i][j] /= pivot;
        }

        // Eliminate other rows
        for (int k = 0; k < N; k++) {
            if (k != i) {
                double factor = aug[k][i];
                for (int j = 0; j < 2*N; j++) {
                    aug[k][j] -= factor * aug[i][j];
                }
            }
        }}
    // Step 3: Extract inverse from augmented matrix
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            inv[i][j] = aug[i][j+N];
        }
    }}
```

Python Through Shared Output

```
import ctypes
import numpy as np
import sympy as sp

# Load C library
lib = ctypes.CDLL(./matrix.so)

# Define function signature
lib.inverse.argtypes = [ctypes.POINTER((ctypes.c_double * 3) * 3)
    ,
                        ctypes.POINTER((ctypes.c_double * 3) * 3)]

# Input matrix
A = np.array([[5, -1, 4],
              [2, 3, 5],
              [5, -2, 6]], dtype=np.double)

inv = np.zeros((3,3), dtype=np.double)
```

Python Through Shared Output

```
# Call C function
lib.inverse(A.ctypes.data_as(ctypes.POINTER((ctypes.c_double * 3)
     * 3)),
           inv.ctypes.data_as(ctypes.POINTER((ctypes.c_double *
               3) * 3)))

inverse=sp.Matrix(inv)
sp.pprint(inverse)
```

Python Code

```
import sympy as sp

A = sp.Matrix([[5, -1, 4], [2, 3, 5],[5, -2, 6]])
A_inv = A.inv()
sp.pprint(A_inv)
```