Matgeo Presentation - 5.5.15

ee25btech11063 - Vejith

September 13, 2025

Question

Using elementary row transformations,find the inverse of the matrix $\mathbf{A}=$
$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ -2 & -4 & -5 \end{pmatrix}$$

Solution

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ -2 & -4 & -5 \end{pmatrix} \tag{0.1}$$

The Augmented matrix is

$$\left(\mathbf{A} \mid \mathbf{I} \right) \implies \left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 7 & 0 & 1 & 0 \\ -2 & -4 & -5 & 0 & 0 & 1 \end{array} \right) \tag{0.2}$$

$$\xleftrightarrow{R_2 \to R_2 - 2R_1} \left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ -2 & -4 & -5 & 0 & 0 & 1 \end{array} \right) \tag{0.3}$$

$$\xrightarrow{R_3 \to R_3 + 2R_1} \left(\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 0 & 1 \end{array} \right) \tag{0.4}$$

Solution

$$\xrightarrow{R_1 \to R_1 - 3R_3} \left(\begin{array}{ccc|ccc} 1 & 2 & 0 & -5 & 0 & 1 \\ 0 & 1 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 0 & 1 \end{array}\right) \tag{0.5}$$

$$\xrightarrow{R_2 \to R_2 - R_3} \left(\begin{array}{ccc|ccc} 1 & 2 & 0 & -5 & 0 & -3 \\ 0 & 1 & 0 & -4 & 1 & -1 \\ 0 & 0 & 1 & 2 & 0 & 1 \end{array}\right) \tag{0.6}$$

$$\xrightarrow{R_1 \to R_1 - 2R_2} \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 3 & -2 & -1 \\ 0 & 1 & 0 & -4 & 1 & -1 \\ 0 & 0 & 1 & 2 & 0 & 1 \end{array}\right) \tag{0.7}$$

As the left block of the Augmented matrix is $\mathbf{I}$ the right block is $\mathbf{A}^{-1}$.

$$\implies \mathbf{A}^{-1} = \begin{pmatrix} 3 & -2 & -1 \\ -4 & 1 & -1 \\ 2 & 0 & 1 \end{pmatrix} \tag{0.8}$$

C Code: inverse.c

```
#include <stdio.h>

#define SIZE 3

int main() {
    FILE *fp;
    double A[SIZE][SIZE] = {
        {1, 2, 3},
        {2, 5, 7},
        {-2, -4, -5}
    };

    double I[SIZE][SIZE] = { {1,0,0}, {0,1,0}, {0,0,1} };

    int i, j, k;
    double factor;

    // Convert A to identity, apply same operations to I
    for (i = 0; i < SIZE; i++) {
        // Make sure pivot is not zero
        if (A[i][i] == 0.0) {
            printf("Pivot is zero, cannot continue.\n");
            return 1;
        }

        // Scale pivot row to make pivot = 1
        factor = A[i][i];
        for (j = 0; j < SIZE; j++) {
            A[i][j] /= factor;
            I[i][j] /= factor;
        }

        // Eliminate other rows
```

C Code: inverse.c

```
        for (k = 0; k < SIZE; k++) {
            if (k != i) {
                factor = A[k][i];
                for (j = 0; j < SIZE; j++) {
                    A[k][j] -= factor * A[i][j];
                    I[k][j] -= factor * I[i][j];
                }
            }
        }
    }

    // Write result to file
    fp = fopen("inverse.dat", "w");
    if (fp == NULL) {
        printf("Error␣opening␣file.\n");
        return 1;
    }

    fprintf(fp, "Inverse␣of␣the␣matrix␣is:\n");
    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE; j++) {
            fprintf(fp, "%8.3f␣", I[i][j]);
        }
        fprintf(fp, "\n");
    }

    fclose(fp);
    printf("Inverse␣written␣to␣inverse.dat\n");

    return 0;
}
```

Python: inverse.py

```
import numpy as np

# Define the matrix
A = np.array([[1, 2, 3],
             [2, 5, 7],
             [-2, -4, -5]])

# Compute inverse
A_inv = np.linalg.inv(A)

print("Inverse of the matrix is:")
print(A_inv)

# Verify by multiplying A and A_inv
identity_check = np.dot(A, A_inv)

print("\nVerification (A * A_inv):")
print(identity_check)
```