5.4.27

Kartik Lahoti - EE25BTECH11032

September 19, 2025

Question

Using elementary transformations, find the inverse of the following matrix.

$$\begin{pmatrix} 2 & 0 & -1 \\ 5 & 1 & 0 \\ 0 & 1 & 3 \end{pmatrix}$$

Theoretical Solution

Given the matrix,

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

Let $\mathbf{A}^{-1}$ be the inverse of the matrix $\mathbf{A}$
We know that,

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \tag{2}$$

Theoretical Solution

The augmented matrix of $\left(\mathbf{A} \mid \mathbf{I}\right)$ is given by ,

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

Theoretical Solution

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

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

Theoretical Solution

$$\left(\begin{array}{ccc|ccc} 1 & 0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{5}{2} & \frac{-5}{2} & 1 & 0 \\ 0 & 1 & 3 & 0 & 0 & 1 \end{array}\right) \xleftrightarrow{R_3 \to R_3 - R_2} \left(\begin{array}{ccc|ccc} 1 & 0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{5}{2} & \frac{-5}{2} & 1 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{5}{2} & -1 & 1 \end{array}\right) \tag{6}$$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{5}{2} & \frac{-5}{2} & 1 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{5}{2} & -1 & 1 \end{array}\right) \xleftrightarrow{R_3 \to 2R_3} \left(\begin{array}{ccc|ccc} 1 & 0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{5}{2} & \frac{-5}{2} & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & 2 \end{array}\right) \tag{7}$$

Theoretical Solution

$$\left(\begin{array}{ccc|ccc} 1 & 0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{5}{2} & \frac{-5}{2} & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & 2 \end{array}\right) \xleftrightarrow{R_2 \to R_2 - \frac{5}{2}R_3} \left(\begin{array}{ccc|ccc} 1 & 0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & -15 & 6 & -5 \\ 0 & 0 & 1 & 5 & -2 & 2 \end{array}\right) \tag{8}$$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & \frac{-1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & -15 & 6 & -5 \\ 0 & 0 & 1 & 5 & -2 & 2 \end{array}\right) \xleftrightarrow{R_1 \to R_1 + \frac{1}{2}R_3} \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 3 & -1 & 1 \\ 0 & 1 & 0 & -15 & 6 & -5 \\ 0 & 0 & 1 & 5 & -2 & 2 \end{array}\right) \tag{9}$$

Theoretical Solution

Hence ,

$$\mathbf{A}^{-1} = \begin{pmatrix} 3 & -1 & 1 \\ -15 & 6 & -5 \\ 5 & -2 & 2 \end{pmatrix} \tag{10}$$

C Code - To find inverse of a Matrix

```c
#include <stdio.h>
void row_mal(double A[3][6] , int n , int m, double k ){
        for(int i = 0 ; i < 6 ;i++)
        {
                A[m][i] -= A[n][i]*k;
        }
}
void row_div(double A[3][6] , int n , int m){
        double k = A[n][m];
        for(int i = 0 ; i <6 ; i++)
        {
                A[n][i] /= k ;
        }
}
```

C Code

```
void inv( double *A , double *B , double *C ){
        double K[3][6];
        for(int i = 0 ; i < 3 ; i++) {
                K[i][0] = A[i];
                K[i][1] = B[i];
                K[i][2] = C[i];
        }
        for(int i = 0 ; i < 3 ; i++) {
           // K[i][i] = 1 ;
                for(int j = 3 ; j < 6 ; j++){
                        if( j-3 == i )
                                K[i][j] = 1 ;
                        else
                                K[i][j] = 0 ;
        }
}
```

C Code

```
        //print
        for(int i = 0 ; i < 3 ; i++)
        {

                for(int j = 0 ; j < 6; j++)
                {
                    if( j < 3){
                        printf("%.1f ",K[i][j]);}
                }
                printf("\n");
        }
```

C Code

```
if(K[0][0] != 0 )
{
    row_div(K , 0 , 0 );
    row_mal(K , 0 , 1 , K[1][0]);
    row_mal(K , 0 , 2 , K[2][0]);
}
else
{
    if(K[1][0] != 0)
            row_mal(K,0,1,-1);
    else if(K[2][0] != 0)
            row_mal(K,0,2,-1);
      row_div(K , 0 , 0 );
      row_mal(K , 0 , 1 , K[1][0]);
      row_mal(K , 0 , 2 , K[2][0]);
}
```

C Code

```
if ( K[1][1] != 0 )
{
    row_div(K , 1, 1);
    row_mal(K, 1, 0 , K[0][1]);
    row_mal(K , 1, 2 , K[2][1]);
}
else
{
    if(K[0][1] != 0 )
            row_mal(K, 1 , 0 , -1);
    else if(K[2][1] != 0 )
            row_mal(K, 1 , 2 ,-1);
    row_div(K , 1, 1);
    row_mal(K, 1, 0 , K[0][1]);
    row_mal(K , 1, 2 , K[2][1]);
}
```

C Code

```
if (K[2][2] != 0 )
{
    row_div(K , 2, 2);
    row_mal(K, 2, 0 , K[0][2]);
    row_mal(K , 2, 1 , K[1][2]);
}
else
{
    if(K[0][2] != 0 )
            row_mal(K,2 , 0 , -1);
    else if(K[1][2] != 0 )
            row_mal(K,2,1,-1);
    row_div(K , 2, 2);
    row_mal(K, 2, 0 , K[0][2]);
    row_mal(K , 2, 1 , K[1][2]);
}
```

C Code

```
    printf("_____________________\n");
        for(int i = 0 ; i < 3 ; i++)
        {

                for(int j = 0 ; j < 6; j++)
                {
                    if ( j >= 3){
                        printf("%.3f ",K[i][j]);}
                }
                printf("\n");
        }
}
```

Python Code

```
import ctypes as ct
import numpy as np

handc1 = ct.CDLL("./func.so")

handc1.inv.argtypes = [
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double),
    ct.POINTER(ct.c_double)
]
```

Python Code

```
A = np.array([2 , 5 , 0 ], dtype =np.float64).reshape(-1,1)
B = np.array([0 , 1 , 1 ], dtype =np.float64).reshape(-1,1)
C = np.array([-1 , 0 , 3 ], dtype =np.float64).reshape(-1,1)

handc1.inv.restype = None

handc1.inv(
    A.ctypes.data_as(ct.POINTER(ct.c_double)),
    B.ctypes.data_as(ct.POINTER(ct.c_double)),
    C.ctypes.data_as(ct.POINTER(ct.c_double))
)
```