

# Linear Algebra Fall 2022– Course Project “Cramer’s Rule”

---

By Omar Elewa

Link to the project:

<https://drive.google.com/drive/u/0/folders/1jwJcedRgYuUoGD7r8Ikt3V22UmUMJi7L>

## Instructions

Write a programming code using MATLAB, Python, C++ or any programming language to solve a system of  $n$  linear equations in  $n$  variables using Cramer’s Rule.

- Your code should allow the user to choose the size  $n \times n$  of the system.
- Make sure that your program first checks that the coefficient matrix is invertible.
- The output of your program must be the unique solution of the system.
- Use your code to solve exercises 25, 26 of Section 3.4, page 142.
- Save your code together with the inputs and outputs of exercise 25 or 26 of section 3.4 as a second PDF file, then submit it to its assigned slot on Gradescope.

## Exercise 25

Solve the following system of linear equation using Cramer’s Rule:

$$3x_1 - 2x_2 + 9x_3 + 4x_4 = 35$$

$$-x_1 - 9x_3 - 6x_4 = -17$$

$$3x_3 + x_4 = 5$$

$$2x_1 + 2x_2 + 8x_4 = -4$$

## Solution

```
Enter the number of rows and columns: 4
Enter the elements of matrix:
Enter the elements of row 1
3
-2
9
4
Enter the elements of row 2
-1
0
-9
-6
Enter the elements of row 3
0
```

```
0
3
1
Enter the elements of row 4
2
2
0
8
The determinant of the coefficient matrix is: 36
The coefficient matrix is non-singular.
Enter the elements of solution matrix:
35
-17
5
-4
The solution matrix is: [35, -17, 5, -4]
The value of x 1 is: 5.0
The value of x 2 is: -3.0
The value of x 3 is: 2.0
The value of x 4 is: -1.0

Process finished with exit code 0
```

## Exercise 26

Solve the following system of linear equation using Cramer's Rule:

```
-x_1 - x_2 + x_4 = -8
3x_1 + 5x_2 + 5x_3 = 24
2x_3 + x_4 = -6
-2x_1 - 3x_2 - 3x_3 = -15
```

### Solution

```
Enter the number of rows and columns: 4
Enter the elements of matrix:
Enter the elements of row 1
-1
-1
0
1
Enter the elements of row 2
3
5
5
0
Enter the elements of row 3
0
0
0
```

```
2
1
Enter the elements of row 4
-2
-3
-3
0
The determinant of the coefficient matrix is: 1
The coefficient matrix is non-singular.
Enter the elements of solution matrix:
-8
24
-6
-15
The solution matrix is: [-8, 24, -6, -15]
The value of x 1 is: 3.0
The value of x 2 is: 7.0
The value of x 3 is: -4.0
The value of x 4 is: 2.0

Process finished with exit code 0
```

## Python Code

```
# determinant of matrix.
# defining a function to get the minor matrix after excluding i-th row and
j-th column.

def get_cofactor(m, i, j):
    return [row[: j] + row[j + 1:] for row in (m[: i] + m[i + 1:])]

# defining the function to calculate determinant value of given matrix a.
def determinant_of_matrix(mat):
    # if given matrix is of order 2*2 then simply return det value by
    cross multiplying elements of matrix.
    if len(mat) == 2:
        value = mat[0][0] * mat[1][1] - mat[1][0] * mat[0][1]
        return value

    # initialize Sum to zero
    Sum = 0

    # loop to traverse each column of matrix a.
    for current_column in range(len(mat)):
        # calculating the sign corresponding to co-factor of that sub
        matrix.
        sign = (-1) ** current_column

        # calling the function recursively to get determinant value of sub
        matrix obtained.
```

```
        sub_det = determinant_of_matrix(get_cofactor(mat, 0,
current_column))

        # adding the calculated determinant value of particular column
matrix to total Sum.
        Sum += (sign * mat[0][current_column] * sub_det)

# returning the final Sum
return Sum

# Driver code
if __name__ == '__main__':

    # input the number of rows
    rows = int(input("Enter the number of rows and columns: "))

    # initialize matrix
    mat = [[0] * rows for i in range(rows)]

    # input the elements of matrix
    print("Enter the elements of matrix: ")
    for i in range(rows):
        print("Enter the elements of row", i + 1)
        for j in range(rows):
            mat[i][j] = int(input())

    # calling the function to get determinant value of given matrix.
    det_coefficient = determinant_of_matrix(mat)

    # printing the determinant value of given matrix.
    print("The determinant of the coefficient matrix is: ",
det_coefficient)

    # Exit the program if determinant is zero.
    if det_coefficient == 0:
        print("The coefficient matrix is singular.")
        exit()
    else:
        print("The coefficient matrix is non-singular.")

    # Enter the solution matrix.
    print("Enter the elements of solution matrix: ")
    solution_matrix = []
    for i in range(rows):
        solution_matrix.append(int(input()))

    # print the solution matrix.
    print("The solution matrix is: ", solution_matrix)

    # Use cramer's rule to solve the system of linear equations.
    for i in range(rows):
        # initialize the matrix to get the minor matrix.
        minor_matrix = [[0] * rows for i in range(rows)]
```

```
# get the minor matrix.
for j in range(rows):
    for k in range(rows):
        if j == i:
            minor_matrix[k][j] = solution_matrix[k]
        else:
            minor_matrix[k][j] = mat[k][j]

# calculate the determinant of minor matrix.
det_minor_matrix = determinant_of_matrix(minor_matrix)

# Exit the program if determinant is zero.
if det_minor_matrix == 0:
    print("The system has infinitely many solutions.")
    exit()

# calculate the value of x.
x = det_minor_matrix / det_coefficient

# print the value of x.
print("The value of x", i + 1, "is: ", x)
```