SPL-1 Project Report , [2022]

Implementation of Matrix Calculator and Linear System Solver

SE 305 : Software Project Lab 1

Submitted by
Faria Akter
BSSE Roll No:1305
BSSE Session:2020-2021

Supervised by
Mohd. Zulfiquar Hafiz
Designation: Professor
Institute of Information Technology

Institute of Information Technology
University of Dhaka

[21-05-2023]

Table of Contents

1. Introduction

A matrix calculator typically provides a user-friendly interface that allows users to input matrices and perform operations such as addition, subtraction, multiplication, transpose, determinant calculation, inverse calculation, and solving linear systems.

A linear system solver is a computational tool or algorithm designed to find the solution to a system of linear equations. A linear system consists of multiple equations, each involving variables raised to the power of one (i.e., linear terms). The goal is to determine the values of the variables that satisfy all the equations simultaneously. Linear system solvers utilize various mathematical techniques to solve these systems efficiently.The common approaches to solving linear equations are Cramer's rule,Inverse method,Gaussian elimination,Gauss Jordan method,Gauss Jacobi method ,Gauss seidel method etc.

1.1 Background study

As this project is based on matrix I have to study all matrix definitions and functionalities.During implementation of those functionalities I choose all efficient algorithms that I thought during my previous semester.I have learnt how to find matrix multiplication efficiently in fast time implementing Strassen's matrix multiplication and learnt Moore-Penrose pseudo inverse for finding not invertible matrix inverse.
I have to know different methods to solve linear systems like Gauss elimination,Gauss jordan,Gauss jacobi,Gauss seidel etc.

1.2 Challenges

There are a number of challenges I had to face while implementing the software. A lot of terms and the majority of the tasks were completely new to me that led me towards much confusion and complexity in implementation. Some of the challenges I faced during this implementation is enlisted below-

- Working with header files for the first time.
- Working with multiple source files.
- Working with big data and handling it.
- Making a user friendly interface.
- Passing a 2D matrix in a function and getting results from it.
- Implementing mathematical notations in a code.

I overcame those problems with the help of my supervisor and getting information from the internet.

2. Project Overview

There are basically two parts in my project:

- ❖ Matrix Calculator
- ❖ Linear System Solving

2.1. Matrix Calculator

In this part, the main goal is to implement all the functionality used in matrix(matrix addition,subtraction,multiplication,transpose,determinant,inverse)

Matrix Addition: To add Two matrices, I used element wise addition process. Here I took input of rows and columns and then took input of two matrices and then added them element wise.

Matrix Subtraction: To subtract two matrices, I used element wise Subtraction process. Here I took input of rows and columns and then took input of two matrices and then subtracted them element wise.

Matrix Multiplication: To multiply two matrices, I used two method

1.Dot multiplication process
2.Strassen's matrix multiplication

In the dot multiplication process at first, I took input of rows and columns of the first matrix and then second matrix and then took input of two matrices and then multiplied them. Every row of the first is multiplied with every column of the second matrix.

Strassen's multiplication is an algorithm used to multiply two matrices more efficiently than the traditional matrix multiplication method. It was developed by Volker Strassen in 1969.
The main idea behind Strassen's multiplication is to reduce the number of scalar multiplications required to compute the product of two matrices. The algorithm takes advantage of the divide-and-conquer approach and recursively divides the matrices into smaller submatrices.

Transpose Matrix: Here I take an input matrix and pass the value of column to a value of row. For this, I took a matrix which stores the values of columns into the values of rows of the inputted matrix and vice-versa.

Determinant: Here I calculated the determinant. I used recursion to find it. At first, the inputted matrix is sent to a function which reduces the matrix by the row and column of the present cell and again finds the determinant until a 2x2 matrix is obtained.

Inverse of a Matrix: For finding Inverse of a Matrix, I used two method

1.Moore-Penrose pseudo inverse

2.Gauss jordan method

Pseudo Inverse: Here I at first transpose the inputted matrix and then I multiplied it with the main matrix. After that, I did the inverse using normal inverse operation where I found the determinant at first and checked if the determinant was 0 or not. If not, then I did the inverse finding the adjoint and then dividing it elementwise by its determinant. I used Pseudo Inverse because it can find the inverse of any matrix (square or non-square).

Gauss jordan method: Here I just adjoin the identity matrix to the right side of the inputted matrix and apply row operation using Gauss jordan method . After this the left part of augmented matrix will become identity matrix and the right part will be the inverse of the inputted matrix

2.2. Linear System Solving

This part represents the solution of any kind of Linear System given by the user. First it takes the Linear equation, then asks the user in which method it will be solved? User option will be like,

Cramer’s Rules.
Inverse Matrix.
Gaussian Elimination.
Gauss Jordan Method.
Gauss Jacobi Iteration Method.
Gauss Seidel Iteration Method.

Cramer’s Rules: Cramer’s Rule is a method used to solve a system of linear equations by finding the values of the variables.It relies on determinants to determine the solution.

a system of linear equations in the form of:

$a_{11}x + a_{12}y + a_{13}z = b_1$

$a_{21}x + a_{22}y + a_{23}z = b_2$

$a_{31}x + a_{32}y + a_{33}z = b_3$

1. Calculate the determinant of the coefficient matrix, denoted as D. The coefficient matrix consists of the coefficients of the variables (a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}).
2. Calculate the determinant of the matrix obtained by replacing the first column of the coefficient matrix with the constants matrix, denoted as Dx. The constants matrix consists of the constants on the right-hand side of the equations (b_1, b_2, b_3).
3. Calculate the determinant of the matrix obtained by replacing the second column of the coefficient matrix with the constants matrix, denoted as Dy.
4. Calculate the determinant of the matrix obtained by replacing the third column of the coefficient matrix with the constants matrix, denoted as Dz.

Finally, the values of the variables can be found using the formulas:

$x = Dx / D$

$y = Dy / D$

$z = Dz / D$

Inverse Matrix method: The inverse matrix method is another approach to solve a system of linear equations by using the concept of matrix inverse.

Given a system of linear equations in matrix form: Ax = b, where A is the coefficient matrix, x is the column vector of variables, and b is the column vector of constants. Calculate the inverse matrix of A, denoted as A^(-1). Multiply both sides of the equation by the inverse matrix: A^(-1)(Ax) = A^(-1)b. Since the product of a matrix and its inverse is the identity matrix, the equation becomes: Ix = A^(-1)b, where I is the identity matrix. Calculate the product of the inverse matrix A^(-1) and the column vector b to obtain the solution vector x, which contains the values of the variables.

Gaussian Elimination Method: Gaussian elimination is a widely used method for solving systems of linear equations. It involves a sequence of elementary row operations to transform the system into an equivalent triangular form, making it easier to solve.
First convert the system to an augmented matrix.
Convert to upper triangular matrix by applying row transformation.
Apply back substitution to find the value of the system

$a_{11}x + a_{12}y + a_{13}z = b_1$

$a_{21}x + a_{22}y + a_{23}z = b_2$

$a_{31}x + a_{32}y + a_{33}z = b_3$

$$\begin{array}{ccc|c} a11 & a12 & a13 & b1 \\ a21 & a22 & a23 & b2 \\ a31 & a32 & a33 & b3 \end{array} \xrightarrow{\text{row operations}} \begin{array}{ccc|c} a11' & a12' & a13' & b1' \\ 0 & a22' & a23' & b2' \\ 0 & 0 & a33' & b3' \end{array}$$

After converting the upper triangular matrix by applying back substitution, find the solution of the system.

Gauss Jordan Method: Gaussian-Jordan elimination, also known as the Gauss-Jordan method, is an extension of the Gaussian elimination method for solving systems of linear equations. It aims to transform the augmented matrix into reduced row-echelon form, leading to a unique and simplified solution.

$a_{11}x + a_{12}y + a_{13}z = b_1$

$a_{21}x + a_{22}y + a_{23}z = b_2$

$a_{31}x + a_{32}y + a_{33}z = b_3$

$$\begin{array}{ccc|c} a11 & a12 & a13 & b1 \\ a21 & a22 & a23 & b2 \\ a31 & a32 & a33 & b3 \end{array} \xrightarrow{\text{row operations}} \begin{array}{ccc|c} a11' & 0 & 0 & b1' \\ 0 & a22' & 0 & b2' \\ 0 & 0 & a33' & b3' \end{array}$$

In this method we don’t need to perform back substitution . we get the result after perform row operations
Here
$x=b1'/a11'$
$y=b2'/a22'$
$z=b3'/a33'$

Gauss Jacobi Iteration Method:Gaussian-Jacobi is an iterative method used to solve systems of linear equations. It is based on the idea of repeatedly refining an initial guess for the solution until it converges to the actual solution
In the Jacobi method, the solution vector is updated using all the previous values from the previous iteration. In other words, all the components of the solution vector are updated simultaneously using the previous iteration values of all the components.

$a_{11}x + a_{12}y + a_{13}z = b_1$

$a_{21}x + a_{22}y + a_{23}z = b_2$

$a_{31}x + a_{32}y + a_{33}z = b_3$

x[i+1]=(b1-a12*y[i]-a13*z[i]) /a11

y[i+1]=(b2-a21*x[i]-a23*z[i]) /a22

z[i+1]=(b3-a31*x[i]-a32*y[i]) /a33

Gauss Seidel Iteration Method: Gaussian-Seidel is an iterative method used to solve systems of linear equations. It is similar to the Gaussian-Jacobi method but includes updated values of the variables within each iteration.

$a_{11}x + a_{12}y + a_{13}z = b_1$

$a_{21}x + a_{22}y + a_{23}z = b_2$

$a_{31}x + a_{32}y + a_{33}z = b_3$

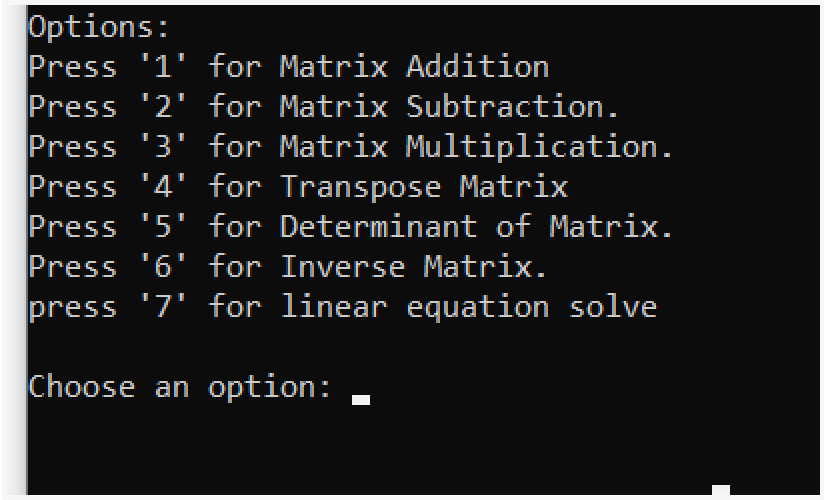
x[i+1]=(b1-a12*y[i]-a13*z[i]) /a11

y[i+1]=(b2-a21*x[i+1]-a23*z[i]) /a22

z[i+1]=(b3-a31*x[i+1]-a32*y[i+1]) /a33

3.User Manual:

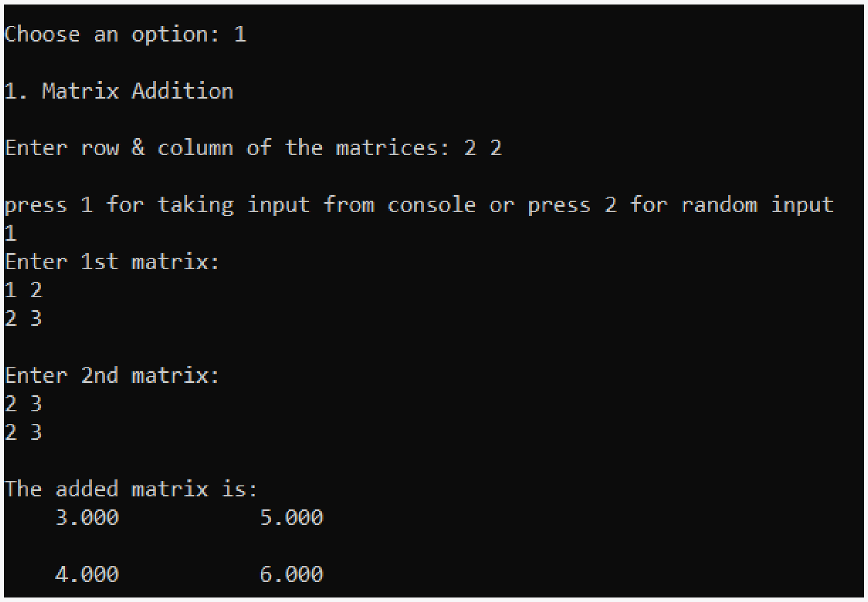
After Opening the console window , we get to choose options that which operation of matrix we want to do



```
Options:
Press '1' for Matrix Addition
Press '2' for Matrix Subtraction.
Press '3' for Matrix Multiplication.
Press '4' for Transpose Matrix
Press '5' for Determinant of Matrix.
Press '6' for Inverse Matrix.
press '7' for linear equation solve

Choose an option: _
```

Figure1:Main function.

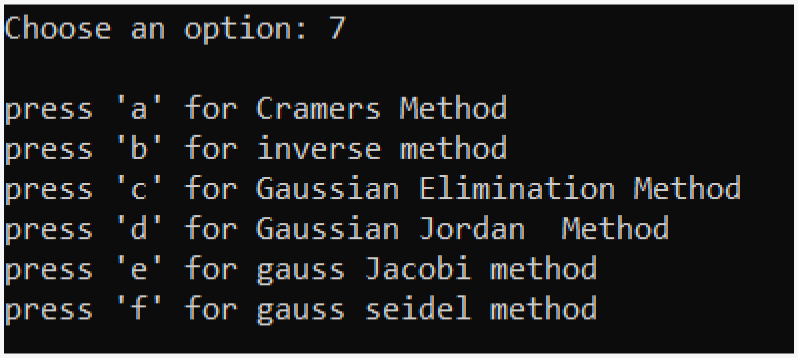
Now, if we select option ‘1’, it will take us to the matrix addition part & will take the dimension of the matrix as input. Then, we get two options. ‘1’ for user input & ‘2’ for random input. If we go to user input it will take the matrices and input and will show us the results



```
Choose an option: 1

1. Matrix Addition

Enter row & column of the matrices: 2 2

press 1 for taking input from console or press 2 for random input
1
Enter 1st matrix:
1 2
2 3

Enter 2nd matrix:
2 3
2 3

The added matrix is:
    3.000          5.000

    4.000          6.000
```

Figure 2 : Matrix Addition.

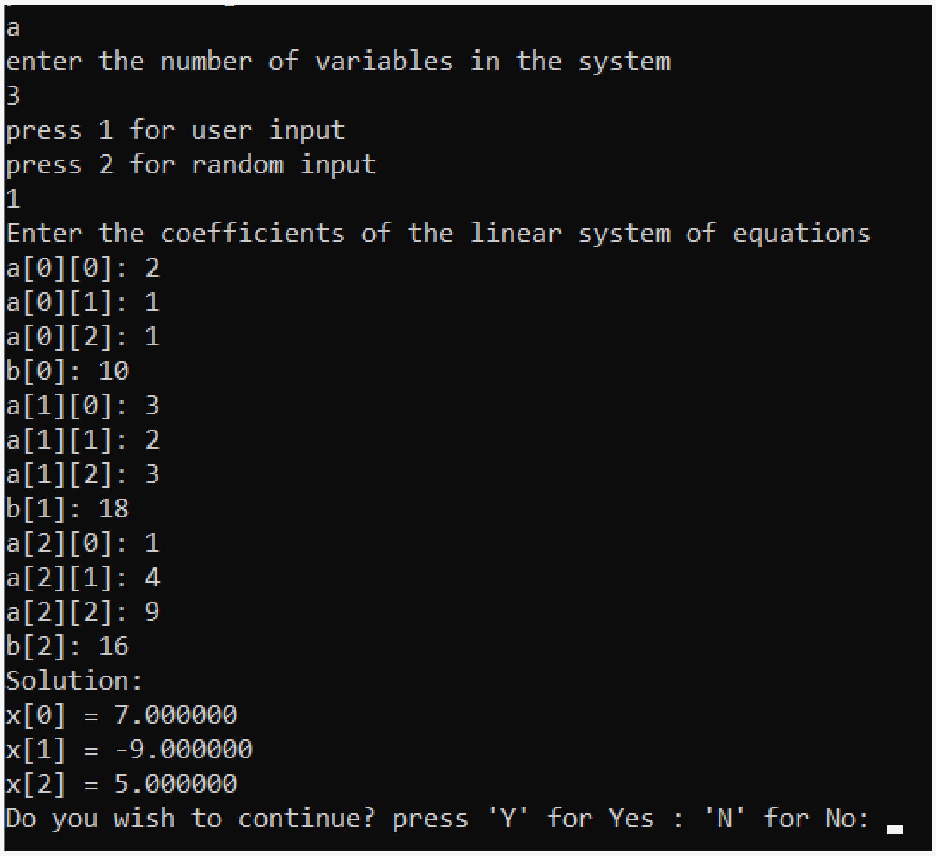
If we choose option 7 we will get option to solve the linear system in 6 ways



```
Choose an option: 7

press 'a' for Cramers Method
press 'b' for inverse method
press 'c' for Gaussian Elimination Method
press 'd' for Gaussian Jordan  Method
press 'e' for gauss Jacobi method
press 'f' for gauss seidel method
```

Figure 3: Linear System Solution Methods.

If we press a it will take number of unknown variables in the system and take the coefficients of the linear system of equations



```
a
enter the number of variables in the system
3
press 1 for user input
press 2 for random input
1
Enter the coefficients of the linear system of equations
a[0][0]: 2
a[0][1]: 1
a[0][2]: 1
b[0]: 10
a[1][0]: 3
a[1][1]: 2
a[1][2]: 3
b[1]: 18
a[2][0]: 1
a[2][1]: 4
a[2][2]: 9
b[2]: 16
Solution:
x[0] = 7.000000
x[1] = -9.000000
x[2] = 5.000000
Do you wish to continue? press 'Y' for Yes : 'N' for No: _
```

Figure 4:Linear System Solve Cramer’s Method.

4.Conclusion :

From the project, I have learnt the operations of matrix. I have the algorithms, which can help me in future studies. I have learnt handling big data and big code & multiple CPP files. I have also learnt how to make a user-friendly program.
My future plan is to extend the project. I want to make a mobile app that can solve all linear equations through the function implemented in this project.I want to make a graphical interface for this.

References

https://www.studypug.com/algebra-help/solving-a-linear-system-with-matrices-using-gaussian-elimination

Last accessed on 14-02-2023

https://atozmath.com/example/CONM/GaussEli.aspx?q=GE2&q1=E1

Last accessed on 21-03-2023

Book : Numerical Methods
Rao V. Dukkipati