

**POLTAVA UNIVERSITY OF ECONOMICS AND TRADE  
EDUCATIONAL AND SCIENTIFIC INSTITUTE OF INTERNATIONAL  
EDUCATION  
FORM OF DAY EDUCATION  
DEPARTMENT OF COMPUTER SCIENCES AND INFORMATION  
TECHNOLOGY**

**Allowed for protection**

Head of the department \_\_\_\_\_ Olena OLKHOVSKA  
(signature)

" \_\_\_\_\_ " \_\_\_\_\_ 2022

**EXPLANATORY NOTE  
FOR THE GRADUATE THESIS**

**DEVELOPMENT OF SIMULATOR SOFTWARE ON THE TOPIC «SYSTEM OF  
LINEAR EQUATIONS» OF THE DISTANCE LEARNING COURSE «HIGHER  
AND APPLIED MATHEMATICS»**

**from specialty 122 «Computer science»  
educational program «Computer science»  
bachelor's degree**

**The executor of the work is Safouane RIFAAT**

\_\_\_\_\_ " \_\_\_\_ " \_\_\_\_\_ 2023.  
(signature)

**Scientific supervisor, Ph.D. k.ped.s. Oksana KOSHOVA**

\_\_\_\_\_ " \_\_\_\_ " \_\_\_\_\_ 2023.  
(signature)

**POLTAVA – 2023**

**POLTAVA UNIVERSITY OF ECONOMICS AND TRADE**

**I APPROVE**

**Head of the department \_\_Olkhovska O.V.  
" \_\_ » \_\_\_\_\_ 2023**

**TASKS AND CALENDAR SCHEDULE  
OF EXECUTION OF DIPLOMA THESIS**

**Graduate of higher education in specialty 122 "Computer science"**

**Educational program "Computer Science"**

**Surname, first name, patronymic \_ Safouane Rifaat**

**1. The topic «Development of simulator software on the topic "System of Linear Equations" of the distance learning course "Higher and applied mathematics"» approved by the rector's order No. \_\_ from \_\_\_\_\_. 2022**

The deadline for the student to submit a thesis « \_\_ » \_\_\_\_\_ 2023

2. Source data for the master's thesis: publications on the topic of educational simulators in distance courses in computer science.

3. Content of the explanatory note (list of issues to be developed)

**LIST OF SYMBOLS, UNITS, ABBREVIATIONS, TERMS**

**INTRODUCTION:**

**CHAPTER 1. PROBLEM STATEMENT**

1.1 Problem Statement:

1.2 Information Review:

**CHAPTER 2. THEORETICAL PART**

2.1 System of linear equations and methods of solving:

2.2 Gaussian elimination:

2.3 Cramer's Rule:

2.4 Matrix Inversion:

2.5 Jacobi Iteration Method:

2.6 Justification of the choice of software:

**CHAPTER 3. PRACTICAL PART 32**

3.1 Development of a block diagram to be programmed:

3.2 Description of software implementation:

**CONCLUSIONS**

**LIST OF REFERENCES**

APPENDIX (A) C# code for the Quiz Form (QuizForm.cs)

APPENDIX (B) C# code for the Solver Form (SolverForm.cs)

4. List of graphic material : 8 sheets of block diagrams, other necessary illustrations.

5. Consultants of sections of the master's thesis

Section	Surname, initials,	Signature, date
---------	--------------------	-----------------

	position of consultant	issued the task	accepted the task
1. Statement of the problem	Oksana Koshova		
2. Information review	Oksana Koshova		
3. Theoretical part	Oksana Koshova		
4. Practical implementation	Oksana Koshova		

6. Calendar schedule of master's work

The content of the work	Deadline	Actual performance
1. Introduction		
2. Study of methodological recommendations and standards and report to the manager		
3. Formulation of the problem		
4. Information review of library and Internet sources		
5. Theoretical part		
6. The practical part		
7. Completion of registration		
8. Student report at the department		
9. Editing (if necessary), reviewing		

Issue date of task " \_\_\_\_ " \_\_\_\_\_ 2023.

Safouane Rifaat, a higher education graduate

Scientific supervisor, Ph.D., k.ped.n., Oksana Koshova.

***The results of the thesis defense***

The thesis was evaluated at \_\_\_\_\_

(points, assessment according to the national scale,

assessment according to ECTS)

Minutes of the meeting of the EC No. \_\_\_\_ of " \_\_\_\_ " \_\_\_\_\_ 2023.

Secretary of the EC \_\_\_\_\_

(signature)

(initials and surname)

**I approve**

Chief department \_\_\_\_\_

Ph.D., k.ph.-m.s. O. Olkhovska

" \_\_\_\_ " \_\_\_\_\_ 2023

**Agreed**

Supervisor \_\_\_\_\_

associate professor, Ph.D. O. Koshova

" \_\_\_\_ " \_\_\_\_\_ 2023

**Plan****Thesis of a higher education graduate with a master's degree****Majors 122 Computer science****Educational program 122 Computer science****Surname, first name, patronymic Safouane Rifaat****On the topic «Development of simulator software on the topic "System of Linear Equations" of the distance learning course "Higher and applied mathematics"».**

LIST OF SYMBOLS, UNITS, ABBREVIATIONS, TERMS

INTRODUCTION:

CHAPTER 1. PROBLEM STATEMENT

1.1 Problem Statement:

1.2 Information Review:

CHAPTER 2. THEORETICAL PART

2.1 System of linear equations and methods of solving:

2.2 Gaussian elimination:

2.3 Cramer's Rule:

2.4 Matrix Inversion:

2.5 Jacobi Iteration Method:

2.6 Justification of the choice of software:

CHAPTER 3. PRACTICAL PART 32

3.1 Development of a block diagram to be programmed:

3.2 Description of software implementation:

CONCLUSIONS

LIST OF REFERENCES

APPENDIX (A) C# code for the Quiz Form (QuizForm.cs)

APPENDIX (B) C# code for the Solver Form (SolverForm.cs)

Graduate of higher education \_\_ Safouane Rifaat \_\_\_\_ " \_\_\_\_ " \_\_\_\_\_ 2023

## ABSTRACT

**Note:** 60 total pages, 9 fig., 2 appendices (on 11 pages), 4 references.

**The purpose of the Grade work.** The purpose of the grade work is development of simulator software on the topic “System of linear equations” of the distance learning course “Higher and applied mathematics”.

**The object of the Grade work** is process of professional education of students majoring in Computer Sciences.

**The subject of the Grade work** is to build an interactive software simulator in the topic System of linear equations for the distance learning of Higher and Applied mathematics.

**Keywords:** Simulator, Distant learning, System of linear equations, Visual Studio 22, C# programming language, Methods of solving linear equations.

## CONTENTS

LIST OF SYMBOLS, UNITS, ABBREVIATIONS, TERMS .....	7
1INTRODUCTION:.....	8
CHAPTER 1. PROBLEM STATEMENT .....	10
1.1 Problem Statement:.....	10
1.2 Information Review: .....	11
CHAPTER 2. THEORETICAL PART .....	12
2.1 System of Linear Equations and methods of solving:.....	12
2.2 Gaussian elimination: .....	13
2.3 Cramer’s Rule: .....	18
2.4 Matrix Inversion: .....	20
2.5 Jacobi Iteration Method:.....	22
2.6 Justification of the choice of software: .....	23
CHAPTER 3. PRACTICAL PART .....	24
3.1 Development of a block diagram to be programmed:.....	24
3.2 Description of software implementation: .....	31
CONCLUSIONS .....	41
LIST OF REFERENCES .....	42
APPENDIX (A) C# code for the Quiz Form (QuizForm.cs).....	44
APPENDIX (B) C# code for the Solver Form (SolverForm.cs).....	56

## LIST OF SYMBOLS, UNITS, ABBREVIATIONS, TERMS

Symbols, units, abbreviations, terms	Explanation of symbols, units, abbreviations, terms
Coefficient Matrix [A]	It represents the matrix that contains the coefficients of the variables in a system of linear equations.
Constant matrix [B]	It represents the matrix of constants on the right-hand side of a system of linear equations. It is a column matrix that contains the constants associated with each equation in the system.
<u><math>Det(A)</math></u>	Determinant is a scalar value associated with a square matrix. It is a mathematical operation performed on a square matrix to extract useful information about the matrix and its properties.
Inverse Matrix $A^{-1}$	It is a matrix denoted as $A^{-1}$ that, when multiplied with the original matrix A, produces the identity matrix. It allows for the recovery of the original values or solving equations involving A.

## 1 INTRODUCTION

Linear algebra is an important part of mathematics that has real-world applications in fields such as engineering, physics, computer science, and economics. Solving systems of linear equations is an important skill that involves solving sets of equations with multiple variables at once. It's a necessary skill for students as it forms the basis for understanding more complex mathematical concepts.

Distance learning has become more popular, giving students with greater flexibility and access to educational resource. "Higher and Applied Mathematics" distance learning course goal is to give students knowledge, skills needed to comprehend how it works, including understanding of linear algebra and its applications.

This grade work focuses on developing simulator software on topic "System of Linear Equations" for distance learning course "Higher and Applied Mathematics." The aim of grade work is to make an interactive, user-friendly software tool that make learning process easier, helping students, understand concepts behind solving systems of linear equations. Software will have a few methods of solving a system of linear equations.

The grade work is divided in four main parts: Introduction, Theoretical Part, Practical Part, and Conclusions. Introduction sets grade work's context, objectives. Theoretical Part looks at problem statement, method chosen for solving problem, brief overview of algorithm used. Practical Part focus on developing software simulator, include flowchart algorithm, implement method, detailed description of program. Finally, Conclusions part offers impartial evaluation of grade work's results, suggestions for implementation, discussion on work's significance.

By making this simulator software, we hope to better learning experience for students in distance learning course System of Linear equations of Higher and Applied Mathematics", to let them understand key concepts of linear algebra, apply them effectively in real-world situations.

**The purpose of the Grade work.** The purpose of the grade work is development of simulator software on the topic "System of linear equations" of the distance learning course "Higher and applied mathematics".



**The object of the Grade work** is process of professional education of students majoring in Computer Sciences.

**The subject of the Grade work** is to build an interactive software simulator in the topic System of linear equations for the distance learning of Higher and Applied mathematics

This simulator software will give students an interactive, user-friendly tool to solve systems of linear equations using different algorithms, visualize solutions, and get better understanding of mathematics concepts. By utilizing this software, students can practice and improve problem-solving skills, help in their overall academic progress.

## CHAPTER 1. PROBLEM STATEMENT

### 1.1 Problem Statement:

The objective of this grade work is to develop a simulator software for the distance learning course "Higher and Applied Mathematics," on the topic "System of Linear Equations." The software should provide an interactive learning experience for students by incorporating the following elements:

- A clear problem statement outlining the goals and objectives of the simulator.
- A comprehensive review of distance learning and the importance of using simulators in education.
- A thorough presentation of the fundamental concepts and methods of solving systems of linear equations, including Gaussian elimination, Cramer's rule, matrix inversion, and the Jacobi method.
- Development of algorithms for each method and creation of corresponding block diagrams.
- Justification for the choice of programming language (C#) and software development platform (Visual Studio 22) used in creating the simulator.
- Detailed description of the software implementation process, including the main stages of development and user instructions.
- An interactive quiz containing 10 questions on the topic.
- A linear equation solver that accepts input coefficients and vectors and outputs the solution.

The initial window of the simulator should display the following information:

- The topic of the simulator
- Theoretical material.
- System Linear Solver.
- Interactive Quiz.

By providing a comprehensive and interactive learning experience, the simulator aims to improve students' understanding of systems of linear equations and their methods of solving.

## 1.2 Information Review:

Distance learning is when you can learn from anywhere without going to a traditional classroom. This way of teaching uses things like online platforms, video lessons, and digital materials. It's the best way for people far away or can't attend traditional classrooms. Also many students prefer distance learning because it's easy to fit into their schedule, accessible, and doesn't cost much. It's a great way to learn new skills efficiently in a short period of time.

Simulators are important for distance learning because they help students learn and practice new things in an interactive and fun way. Simulators let students:

- Practice what they learn in a way that's easy to use.
- Get feedback quickly on how they're doing.
- Learn how to solve problems and think critically.
- Be involved in their learning, which helps them understand better.

When teachers use simulators in distance learning classes, they can help students learn better and prepare them for the real world.

The project features a software with 3 features:

- Theory of System of linear equations.
- System of Linear Equations Quiz
- System of Linear Equations Solver.

## CHAPTER 2. THEORETICAL PART

### 2.1 System of Linear Equations and methods of solving:

System of linear equations is a group of equations with multiple variables. We have to find the values of the variables which satisfy all the equations. There are different ways to solve them.

One way to solve a system of linear equations is by graphing. To do this, we graph each equation on the same coordinate plane and find the point where they intersect. This point represents the solution to the system of equations. But this method can be time-consuming and less accurate, especially when dealing with large systems or numbers.

Another method to solve a system of linear equations is substitution. To use this method, we solve one equation for one variable and substitute that expression into the other equation. This allows us to solve for the remaining variable. We can then substitute this value back into one of the original equations to solve for the first variable.

A third method to solve a system of linear equations is elimination. In this method, we subtract or add multiples of the equations to eliminate one of the variables. Then we solve the remaining variable and substitute this value back into one of the original equations to solve for the other variable.

Also it is important to know that not all systems of linear equations have a solution. If the equations are inconsistent, which means they have no solution, their graphs will be parallel lines that never cross. If the equations are dependent, meaning they have an infinite number of solutions, their graphs will be coincident lines that overlap.

In conclusion, there are many methods to solve systems of linear equations, like graphing, substitution, and elimination. Each methods have their advantages and disadvantages, so it's important to choose the most appropriate method based on the problem we have.

## 2.2 Gaussian elimination:

Gaussian elimination is a method for solving system of linear equations, especially when dealing with large systems of equations, however it may encounter difficulties in certain cases, such as when a row is all zeros which leads to an infinite amount of solutions or when the system is ill-conditioned which is basically leading to numerical instability.

Gaussian elimination involves basic steps, transforming the system of equations into an equivalent system with simpler equations until a solution is obtained. It makes the matrix into a triangular form. Then it solves the system easily

- Create the matrix of the system.
- Use row operations to make zeros below the diagonal.
- Solve the triangular system

The Gaussian elimination method has two main steps: forward elimination and back substitution. Here is a brief description of the algorithm:

1. Forward Elimination:
  - a. Iterate through each row in the coefficient matrix.
  - b. For each row, find the pivot element (the element with the largest absolute value) in the current row or below it.
  - c. Swap the current row with the row containing the pivot element.
  - d. For all the rows below the current row, eliminate the current row's leading coefficient by subtracting an appropriate multiple of the current row from each of the rows below it.
2. Back Substitution:
  - a. Start with the last row and solve for the corresponding variable.
  - b. Move up to the previous row, substitute the previously found variable value(s) into the equation, and solve for the remaining variable.

Continue moving up and solving for variables until the first row is reached.

And these are steps to follow:

- 1) Step 1: Write the system of equations
  - a) Provide the system of linear equations in standard form.
  - b) Arrange the equations in a matrix format (augmented matrix)
- 2) Step 2: Row operations
  - a) Perform row operations on the augmented matrix to simplify the equations.
  - b) The three elementary row operations are:
    - i) Scaling: Multiply a row by a nonzero constant.
    - ii) Row switching: Interchange two rows.
    - iii) Row addition: Add a multiple of one row to another row.
- 3) Step 3: Eliminate variables
  - a) Starting from the first row, use row operations to eliminate the variables below the pivot element (the leftmost nonzero element) in each row.
    - b) Repeat this process for each row until the augmented matrix is in row echelon form or reduced row echelon form.
- 4) Step 4: Back substitution
  - a) Once the augmented matrix is in row echelon form or reduced row echelon form, solve for the variables starting from the bottom row.
    - b) Substitute the known values of variables into the equations above and solve for the remaining variables.
- 5) Step 5: Interpret the solution
  - a) Write the solution as an ordered set of values for each variable.
  - b) If there is no solution (inconsistent system), identify any inconsistencies or contradictions in the system of equations.

**Here is an example:**

Consider the following equations:

$$2x + 3y - z = 7$$

$$x - y + 2z = -1$$

$$3x + 2y - 4z = 4$$

Step 1: Write the system of equations in augmented matrix form:

$$\begin{bmatrix} 2 & 3 & -1 \\ 1 & -1 & 2 \\ 3 & 2 & -4 \end{bmatrix} \begin{bmatrix} 7 \\ -1 \\ 4 \end{bmatrix}$$

Step 2: Perform row operations to simplify the equations.

Let's multiply each element of Row 1 by 1/2

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} \\ 1 & -1 & 2 \\ 3 & 2 & -4 \end{bmatrix} \begin{bmatrix} \frac{7}{2} \\ -1 \\ 4 \end{bmatrix}$$

Now let's perform the row operation:  $R2 = R2 - R1$

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} \\ 1-1 & -1-\frac{3}{2} & 2-(-\frac{1}{2}) \\ 3 & 2 & -4 \end{bmatrix} \begin{bmatrix} \frac{7}{2} \\ -1-\frac{7}{2} \\ 4 \end{bmatrix}$$

We then simplify Row 2:

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} \\ 0 & -\frac{5}{2} & \frac{5}{2} \\ 3 & 2 & -4 \end{bmatrix} \begin{bmatrix} \frac{7}{2} \\ -\frac{9}{2} \\ 4 \end{bmatrix}$$

Now let's perform the row operation:  $R3 = R3 - 3 * R1$

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} \\ 0 & -\frac{5}{2} & \frac{5}{2} \\ 3-3*1 & 2-3*\frac{3}{2} & -4-3*(-\frac{1}{2}) \end{bmatrix} \begin{bmatrix} \frac{7}{2} \\ -\frac{9}{2} \\ 4-3(\frac{7}{2}) \end{bmatrix}$$

We then simplify Row 3:

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} \\ 0 & -\frac{5}{2} & \frac{5}{2} \\ 0 & -\frac{5}{2} & -\frac{5}{2} \end{bmatrix} \begin{bmatrix} \frac{7}{2} \\ -\frac{9}{2} \\ -\frac{13}{2} \end{bmatrix}$$

Now let's multiply each element of Row 2 by  $-\frac{2}{5}$

$$\left[ \begin{array}{ccc|c} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{7}{2} \\ 0 * -\frac{2}{5} & -\frac{5}{2} * (-\frac{2}{5}) & \frac{5}{2} * (-\frac{2}{5}) & -\frac{9}{2} * (-\frac{2}{5}) \\ 0 & -\frac{5}{2} & -\frac{5}{2} & -\frac{13}{2} \end{array} \right]$$

We then simplify Row 2:

$$\left[ \begin{array}{ccc|c} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{7}{2} \\ 0 & 1 & -1 & \frac{9}{5} \\ 0 & -\frac{5}{2} & -\frac{5}{2} & -\frac{13}{2} \end{array} \right]$$

Now let's perform the row operation  $R3 = R3 + \frac{5}{2} * R2$

$$\left[ \begin{array}{ccc|c} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{7}{2} \\ 0 & 1 & -1 & \frac{9}{5} \\ 0 + \frac{5}{2} * 0 & -\frac{5}{2} + \frac{5}{2} * 1 & -\frac{5}{2} + \frac{5}{2} * (-1) & -\frac{13}{2} + \frac{5}{2} * \frac{9}{2} \end{array} \right]$$

We then simplify Row 3:

$$\left[ \begin{array}{ccc|c} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{7}{2} \\ 0 & 1 & -1 & \frac{9}{5} \\ 0 & 0 & -5 & -2 \end{array} \right]$$

Now let's multiply each element of Row 3 by  $-\frac{1}{5}$

$$\left[ \begin{array}{ccc|c} 1 & \frac{3}{2} & -\frac{1}{2} & \frac{7}{2} \\ 0 & 1 & -1 & \frac{9}{5} \\ 0 * (-\frac{1}{5}) & 0 * (-\frac{1}{5}) & -5 * (-\frac{1}{5}) & -2 * (-\frac{1}{5}) \end{array} \right]$$

Now let's simplify Row 3:



$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{2} \\ 9 \\ \frac{2}{5} \end{bmatrix}$$

Then we perform the row operation  $R2 = R2 + R3$

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} \\ 0+0 & 1+0 & -1+1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{2} \\ 9 + \frac{2}{5} \\ \frac{2}{5} \end{bmatrix}$$

Let's simplify Row 2:

$$\begin{bmatrix} 1 & \frac{3}{2} & -\frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{2} \\ \frac{11}{5} \\ \frac{2}{5} \end{bmatrix}$$

Then we perform the row operation  $R1 = R1 + \frac{1}{2} * R3$

$$\begin{bmatrix} 1 + \frac{1}{2} * 0 & \frac{3}{2} + \frac{1}{2} * 0 & -\frac{1}{2} + \frac{1}{2} * 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{2} + \frac{1}{2} * \frac{2}{5} \\ \frac{11}{5} \\ \frac{2}{5} \end{bmatrix}$$

Let's simplify Row 1:

$$\begin{bmatrix} 1 & \frac{3}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{37}{10} \\ \frac{11}{5} \\ \frac{2}{5} \end{bmatrix}$$

Now we perform our last row operation  $R1 = R1 - \frac{3}{2} * R2$

$$\begin{bmatrix} 1 - \frac{3}{2} * 0 & \frac{3}{2} - \frac{3}{2} * 1 & 0 - \frac{3}{2} * 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{37}{10} - \frac{3}{2} * \frac{11}{5} \\ \frac{11}{5} \\ \frac{2}{5} \end{bmatrix}$$

Let's simplify Row 1:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{5} \\ \frac{11}{5} \\ \frac{2}{5} \end{bmatrix}$$

The solution of the matrix is:  $x = \frac{2}{5}$ ;  $y = \frac{11}{5}$ ;  $z = \frac{2}{5}$ .

### 2.3 Cramer's Rule:

Cramer's rule is a method for solving system of linear equations. It works for square matrices and only when the determinant of the coefficient matrix is non-zero. Here is a brief description of the algorithm:

- 1. Check the coefficient matrix:
  - Ensure the matrix is square and calculate its determinant.
  - If the determinant is zero, Cramer's Rule cannot be used.
- 2. Calculate the determinants:
  - For each variable, replace the corresponding column in the coefficient matrix with the constant matrix.
    - Calculate the determinant for each modified matrix.
- 3. Solve for the variables:
  - Divide the determinant of each modified matrix by the determinant of the coefficient matrix.
    - The resulting quotients are the solutions for the corresponding variables.

#### Example:

Suppose we have a system of linear equations:

$$x + 2y = 8$$

$$3x - 4y = 6$$

First, we represent the system in matrix form:

Coefficient matrix (A):

$$\begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix}$$

Constant matrix (B):

$$\begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

1- Check the coefficient matrix:

Calculate the determinant of matrix A:

$$\text{Det}(A) = 1 * (-4) - 3 * 2 = -10$$

Since the determinant is non-zero, we can use Cramer's Rule.

2- Calculate the determinants:

For each variable, replace the corresponding column in the coefficient matrix with the constant matrix and calculate the determinant:

Matrix  $A_1$ :

$$\begin{bmatrix} 8 & 2 \\ 6 & 4 \end{bmatrix}$$

$$\text{Det}(A_1) = 8 * (-4) - 6 * 2 = -44$$

Matrix  $A_2$ :

$$\begin{bmatrix} 1 & 8 \\ 3 & 6 \end{bmatrix}$$

$$\text{Det}(A_2) = 1 * 6 - 3 * 8 = -18$$

3- Solve for the variables:

Divide the determinant of each modified matrix by the determinant of the coefficient matrix:

$$x = \text{Det}(A_1) / \text{Det}(A) = -44 / -10 = 4.4$$

$$y = \text{Det}(A_2) / \text{Det}(A) = -18 / -10 = 1.8$$

Solution:

$$x = 4.4$$

$$y = 1.8$$

Thus, the solution to the system of linear equations using Cramer's Rule is  $x = 4.4$  and  $y = 1.8$ .

## 2.4 Matrix Inversion:

Matrix inversion is a method that uses inverse of a matrix to solve the system. First, we need to find the inverse of the matrix, then multiply it with the constants to find the solution.

1. Check the coefficient matrix:

- a. Ensure the matrix is square and calculate its determinant.
- b. If the determinant is zero, the matrix is not invertible and the method cannot be used.

2. Calculate the inverse:

- a. Find the inverse of the coefficient matrix using any suitable method (e.g., Gaussian elimination).

3. Multiply the inverse:

- a. Multiply the inverse of the coefficient matrix by the constant matrix.
- b. The resulting matrix will have the solutions for the variables.

### Example:

Suppose we have a system of linear equations:

$$3x + 8y = 5$$

$$4x + 11y = 7$$

First, we represent the system in matrix form:

Coefficient matrix (A):

$$\begin{bmatrix} 3 & 8 \\ 4 & 11 \end{bmatrix}$$

Constant matrix (B):

$$\begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

Constant matrix (X):

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

Then

$$\begin{bmatrix} 3 & 8 \\ 4 & 11 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

1. Check the coefficient matrix:

Calculate the determinant of matrix A:

$$\text{Det}(A) = 3 * 11 - 8 * 4 = 1$$

Since the determinant is non-zero, the matrix is invertible, and we can use the Matrix Inversion method.

2. Calculate the inverse:

We can use any method to find the inverse of matrix A.

Let's use the formula to calculate the inverse of a 2 by 2 matrix:

$$A^{-1} = \frac{1}{ad - bc} * \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$A^{-1} = \frac{1}{3 * 11 - 8 * 4} * \begin{bmatrix} 11 & -8 \\ -4 & 3 \end{bmatrix}$$

$$A^{-1} = \frac{1}{1} * \begin{bmatrix} 11 & -8 \\ -4 & 3 \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} 11 & -8 \\ -4 & 3 \end{bmatrix}$$

3. Multiply the inverse:

Now let's multiply the sides of the equations by the inverse matrix.

$$(A^{-1})AX = (A^{-1})B$$

$$\begin{bmatrix} 11 & -8 \\ -4 & 3 \end{bmatrix} * \begin{bmatrix} 3 & 8 \\ 4 & 11 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 11 & -8 \\ -4 & 3 \end{bmatrix} * \begin{bmatrix} 5 \\ 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} 11 * 5 & -8 * 7 \\ -4 * 5 & 3 * 7 \end{bmatrix}$$

$$\begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

The solution to this system linear equation is  $x = -1$  and  $y = 1$

## 2.5 Jacobi Iteration Method:

The Jacobi Iteration method is an iterative algorithm for solving system of linear equations. It works by updating the variables' estimates in parallel. Here is a brief description of the algorithm:

1- Initialize the variables:

a. Provide an initial guess for each variable (e.g., zeros).

2- Iterate until convergence:

a. For each variable, compute the new estimate using the current values of the other variables.

b. Update all variable estimates in parallel.

c. Check for convergence (e.g., based on the difference between consecutive estimates or the residual).

d. If the convergence criterion is not met, repeat the process.

3- Obtain the solution:

a. When the convergence criterion is met, the final estimates represent the solution to the system.

### Example:

Suppose that we have 3 equations that we want to solve with three unknowns  $x$ ,  $y$  and  $z$ .

$$\text{Equation 1: } 3x_1 + x_2 + 4x_3 = 1$$

$$\text{Equation 2: } x_1 + 5x_2 + 9x_3 = 1$$

$$\text{Equation 3: } 2x_1 + 6x_2 + 5x_3 = 1$$

Let's suppose that our initial approximation of the solution is  $x^{(0)} = \begin{bmatrix} 0.22 \\ 0.03 \\ 0.06 \end{bmatrix}$

We isolate the variable  $x_i$  from equation  $E(i)$  for  $i = 1, 2, 3$  to get:

$$x_1 = \frac{1 - x_2 - 4x_3}{3}$$

$$x_2 = \frac{1 - x_1 - 9x_3}{5}$$

$$x_3 = \frac{1 - 2x_1 - 6x_2}{5}$$

Based on our initial approximation  $x^{(1)}$  using the Jacobi Iteration method, we have:

$$x_1^{(1)} = \frac{1 - (0.03) - 4 * (0.06)}{3} = 0.243$$

$$x_2^{(1)} = \frac{1 - (0.22) - 9 * (0.06)}{5} = 0.048$$

$$x_3^{(1)} = \frac{1 - 2 * (0.22) - 4 * (0.03)}{5} = 0.184$$

We can then continue the Jacobi Iteration to have further approximations/

The solution to this system is  $x = \begin{bmatrix} 0.23 \\ 0.03 \\ 0.06 \end{bmatrix}$ .

## 2.6 Justification of the choice of software:

A number of variables influenced the software used to create the simulator, including:

- Visual Studio 2022: One of the most complete and commonly used integrated development environments (IDEs) is Visual Studio. It supports various languages, including C#, PHP, Python, JavaScript etc... and it contains a set of developer tools that facilitate its usage. Visual Studio 2022 was chosen its easy use, great performance, and great support for Windows application development.
- Language C#: C# is a well-known programming language, it is simple to learn and to work with. It's especially good for Windows development, and it's also integrated with the .NET framework, which has a large library of pre-coded answers to common programming problems, speeding up the development process even further.
- Windows Forms: Windows Forms is a graphical user interface (GUI) class library that is part of Microsoft .NET Framework. It is simple to use and allows the building of dynamic and user friendly interfaces. It is especially useful in our project since it provides for easy organization of the simulator's numerous parts.

## CHAPTER 3. PRACTICAL PART

### 3.1 Development of a block diagram to be programmed:

Figure 1 displays the block diagram of the algorithm for the main simulator initializer.

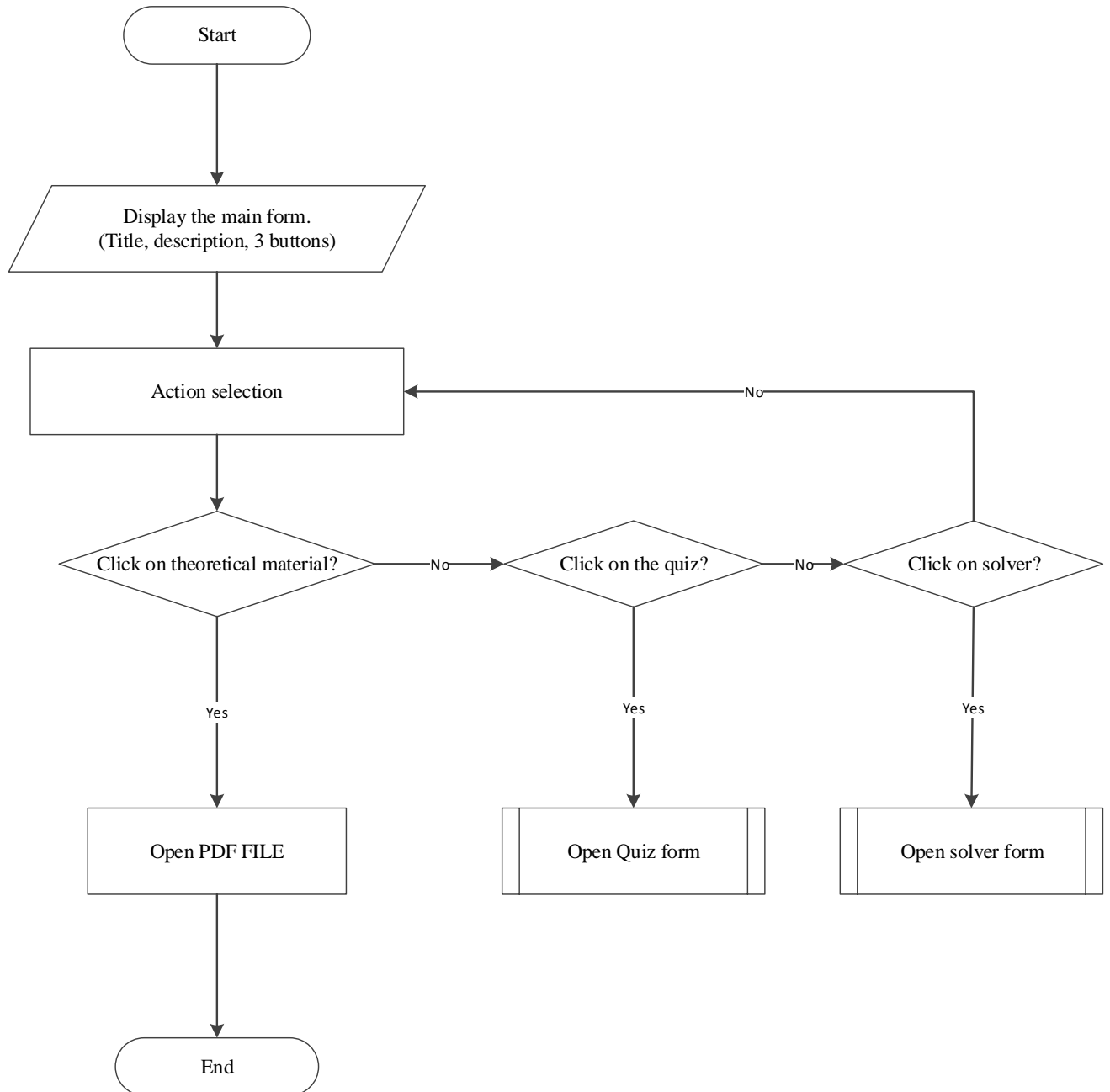


Figure 1. Block Diagram of the Main Form of the Simulator.



Figure 2.1 – 2.5 displays the block diagram of the algorithm for the quiz form.

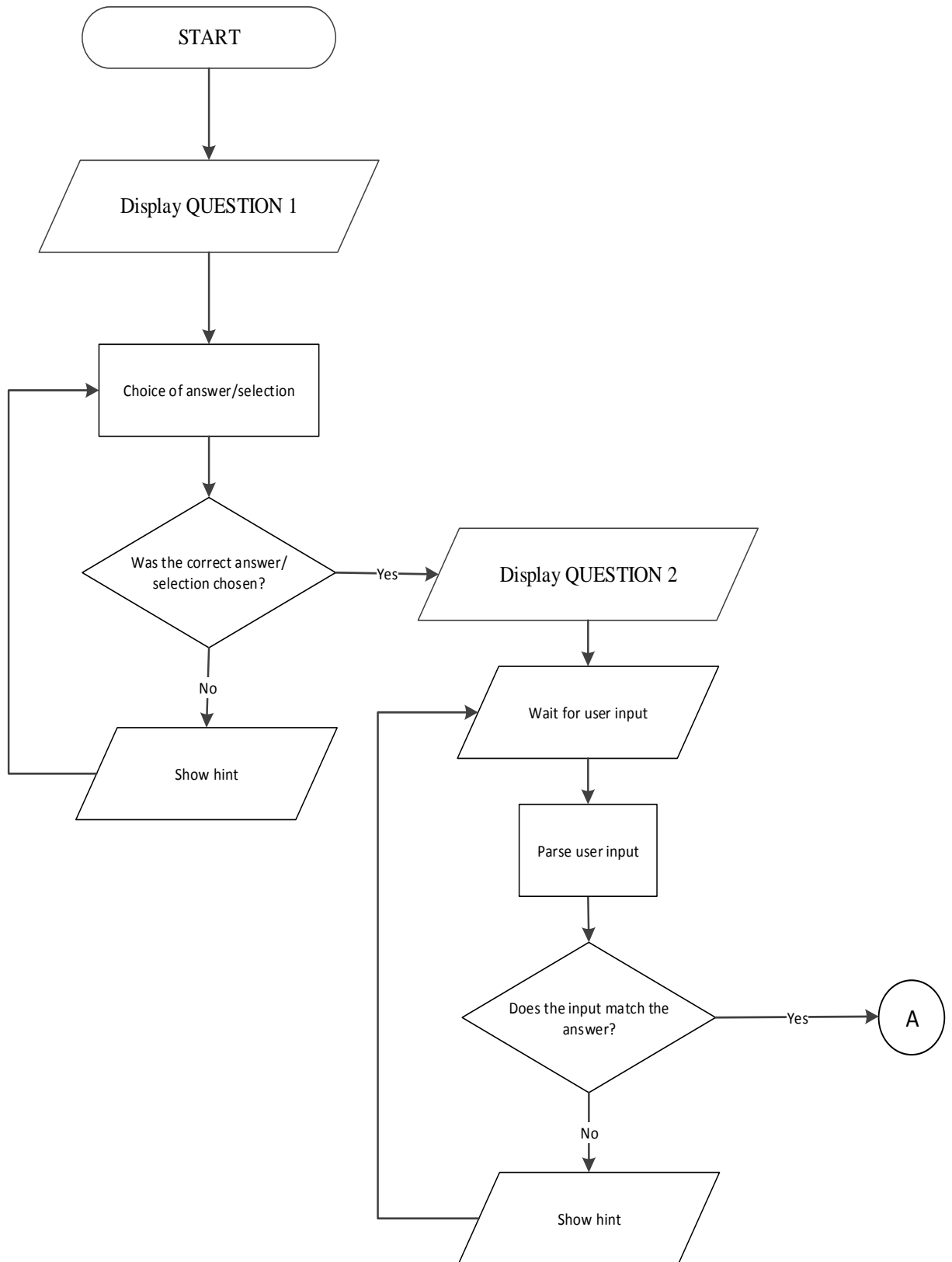


Figure 2.1 Block diagram of the algorithm for the Quiz Form.

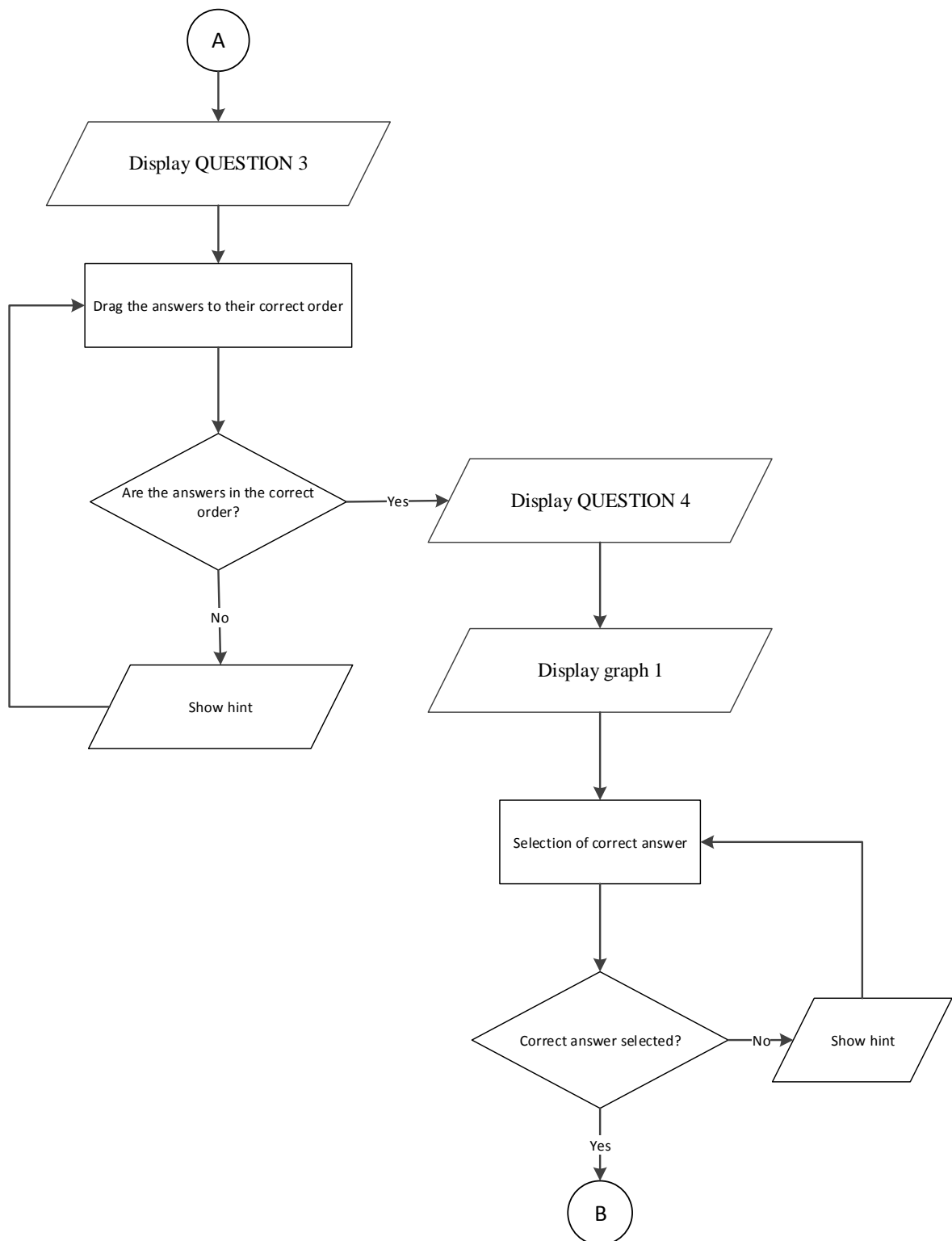


Figure 2.2 Continuation of the Block diagram of the algorithm for the Quiz Form.

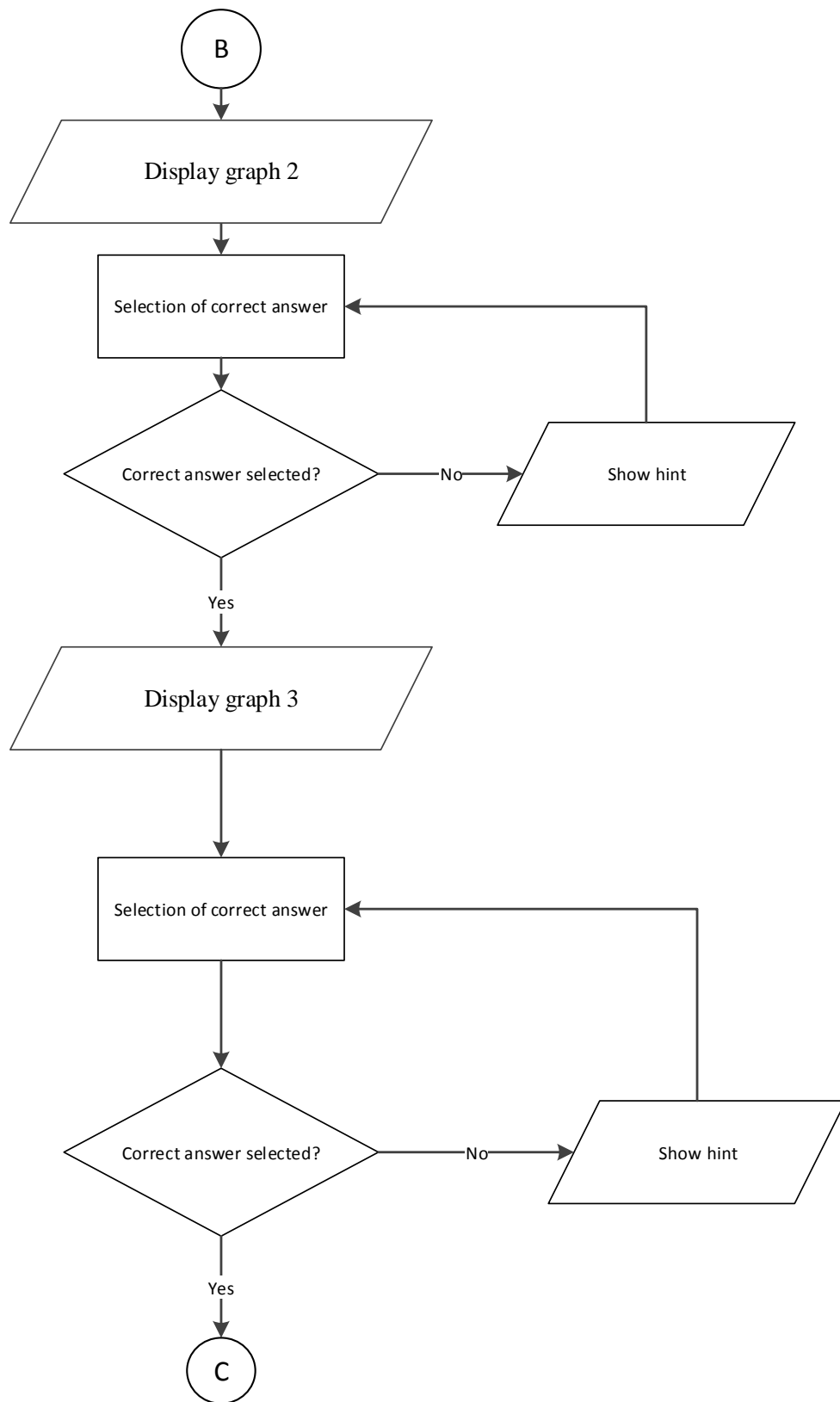


Figure 2.3 Continuation of the Block diagram of the algorithm for the Quiz Form.

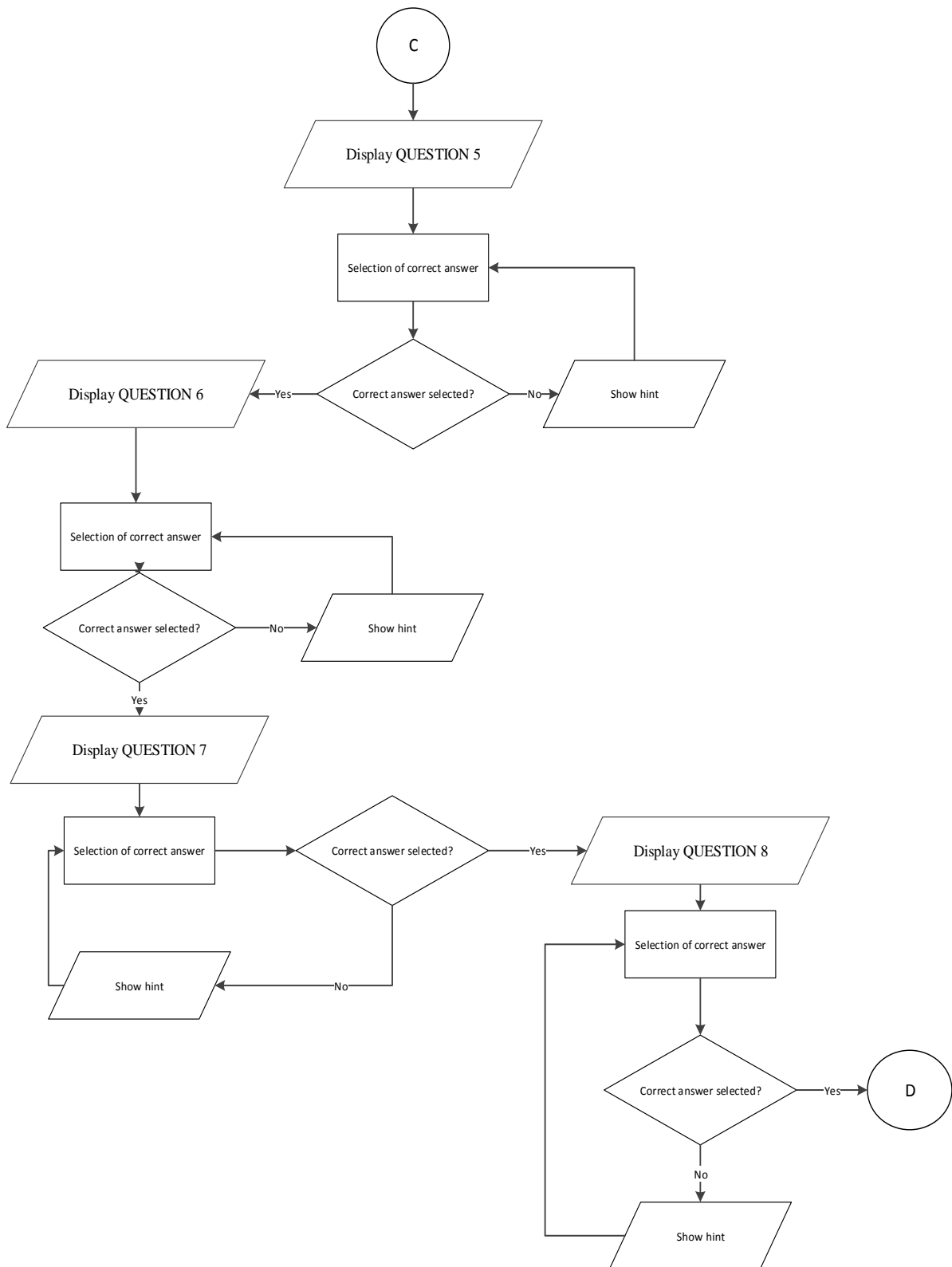
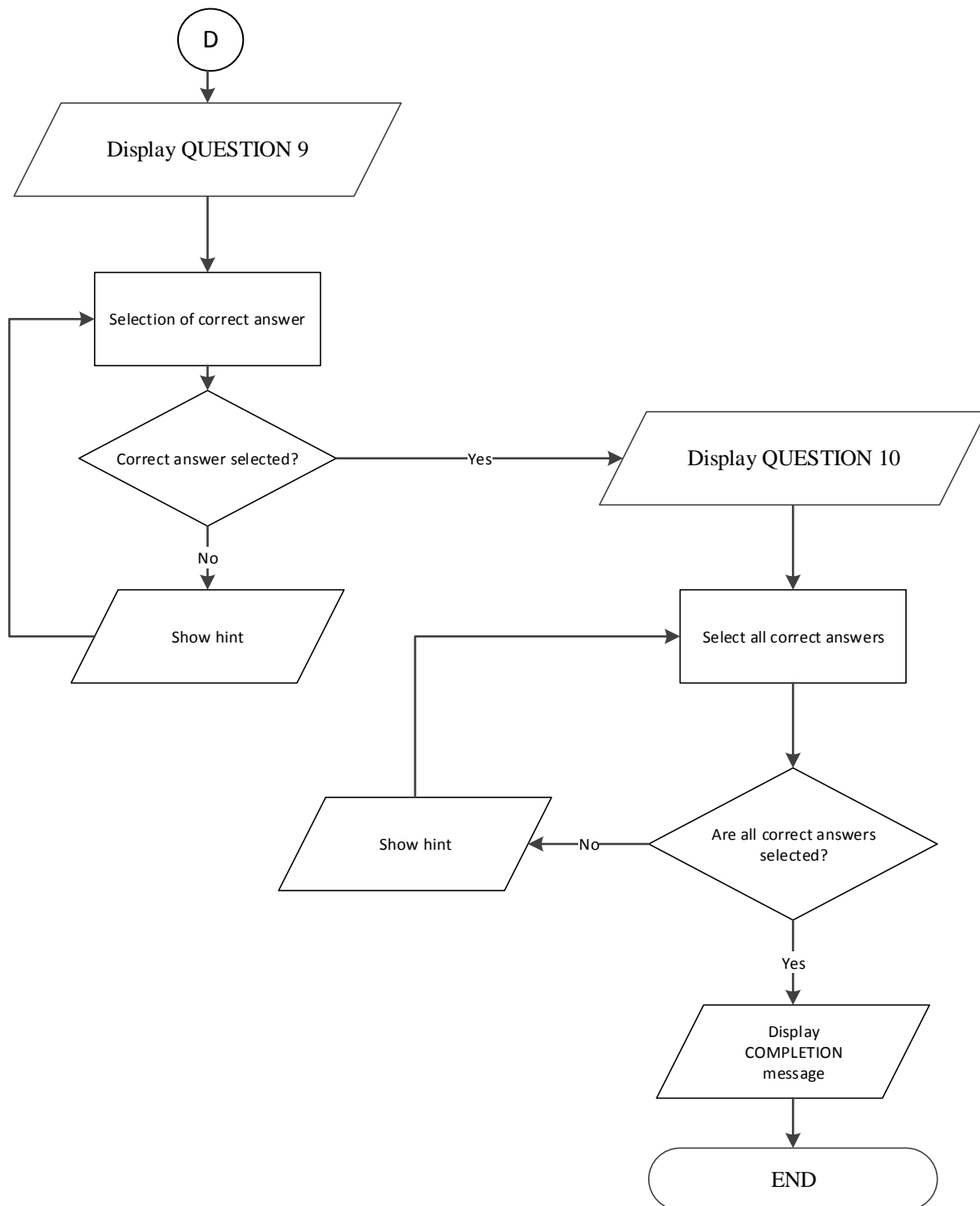


Figure 2.4 Continuation of the Block diagram of the algorithm for the Quiz Form.



Figure

2.5 Continuation of the Block diagram of the algorithm for the Quiz Form.

Figure 3 displaying the Block Diagram of algorithm for System Linear Solver

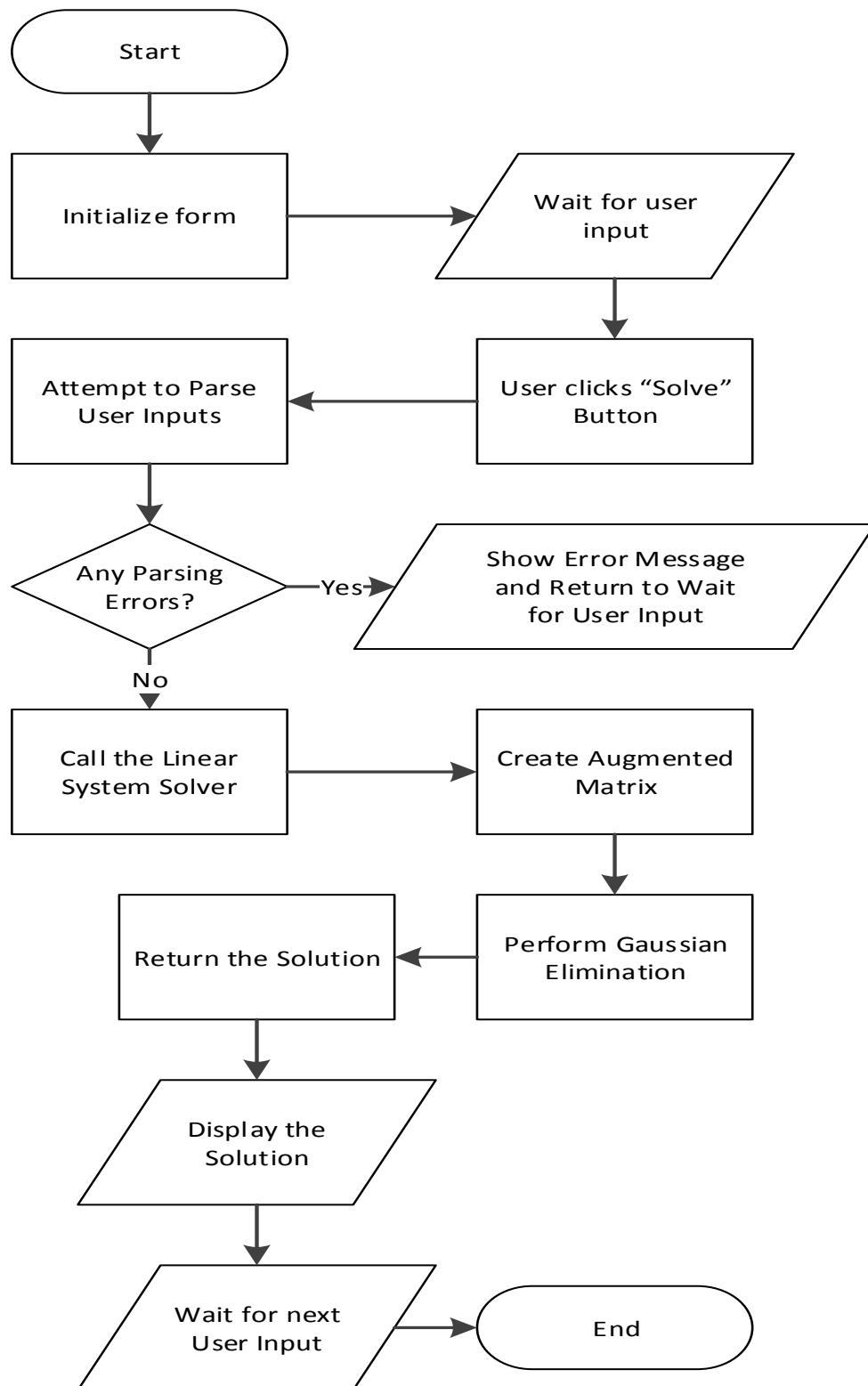


Figure 3. Block Diagram of the algorithm for System of linear equations solver

### 3.2 Description of software implementation:

The system of linear equations software simulator was created using Visual Studio 2022, and the language used for coding was C#.

The software simulator starts with an initial window that displays the title of the software, a description of the simulator and three buttons the user can interact with.

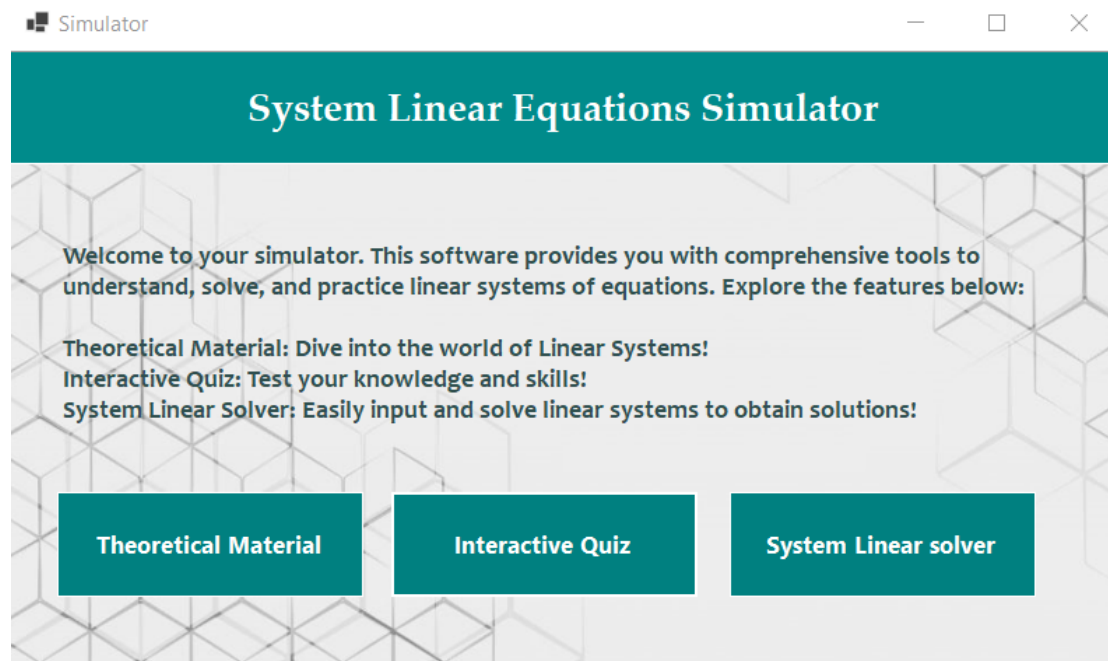


Figure 4.1 Initial main form of the software simulator

The user can choose to interact with the Theoretical Material which will lead the user to open a PDF file in any of his available PDF readers, or else their internet browser.

The interactive Quiz button opens a pop up form which initializes the Quiz questions and check for the user's answers.

The System Linear Solver button opens a pop up form which initializes the Solver form and takes input from the user then solving the linear equation by parsing the input.

- **The Theoretical Material button is executed by the following code:**

```
private void openPdfButton_Click(object sender, EventArgs e)
{
    string pdfFileName = "Theoretical material.pdf";
    string pdfPath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, pdfFileName);
    try
    {
        OpenPdfWithShellExecute(pdfPath);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: Unable to open the PDF file.\n" + ex.Message, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void OpenPdfWithShellExecute(string pdfPath)
{
    SHELLEXECUTEINFO execInfo = new SHELLEXECUTEINFO
    {
        cbSize = Marshal.SizeOf(typeof(SHELLEXECUTEINFO)),
        lpVerb = "open",
        lpFile = pdfPath,
        nShow = (int)SW_NORMAL,
        fMask = SEE_MASK_INVOKEIDLIST
    };

    if (!ShellExecuteEx(ref execInfo))
    {
        throw new Win32Exception(Marshal.GetLastWin32Error());
    }
}
```



Here we have two methods used, `openPdfButton_Click` and `OpenPdfWithShellExecute`.

The first method declares a string variable with the name of the PDF file to open, then constructs the path to the file, then it tries to execute the second method if the previous failed.

The second method creates an instance of “SHELLEXECUTEINFO”, a structure used to pass parameters to and receive results from the `ShellExecuteEX` function, it then sets different properties of the structure, then tries to open the file using “`ShellExecuteEx`” which opens the file in the device’s default internet browser if no available pdf reader is available.

- **The Interactive Quiz button is executed by the following code:**

```
private void startQuizButton_Click(object sender, EventArgs e)
{
    QuizForm quizForm = new QuizForm();
    quizForm.Show();
}
```

This C# code represents an event handler for a button click event in Windows Forms application.

“`QuizForm.Show();`” is responsible for popping up the Quiz Form.

- **The System Linear Solver button is executed by the following code:**

```
private void solver_Click(object sender, EventArgs e)
{
    SolverForm solverForm = new SolverForm();
    solverForm.ShowDialog();
}
}
```

This C# code represents an event handler for a button click event in Windows Forms application.

“solverform.ShowDialog();” is responsible for popping up the System linear of equations Form.

For our necessary files, in figure 4.2 is our Solution Explorer which displays all the forms and files that we need for the project. (PDF and background picture)

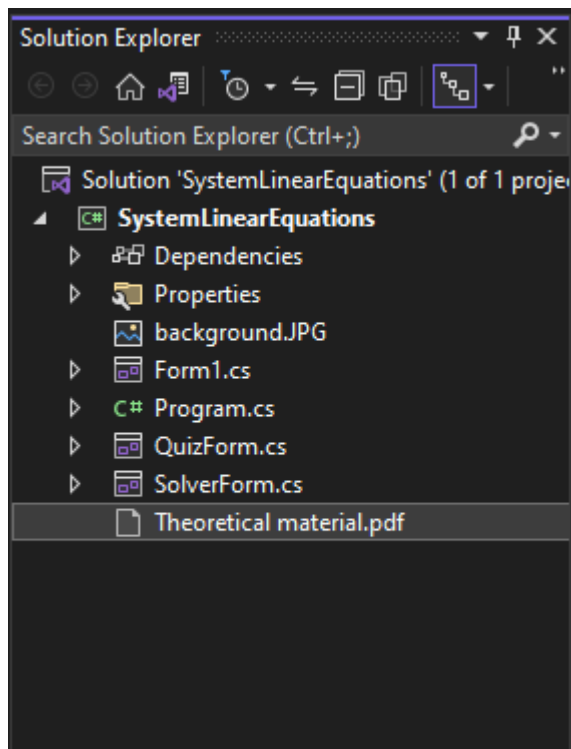


Figure 4.2 Solution Explorer window in VS 2022

For the QuizForm.cs, we will briefly detail how it initializes when the user clicks on the Quiz button.

The pop up window shows the 1st question then when the user clicks on the Submit button, this code is executed:

```

private void submitButton_Click(object sender, EventArgs e)
{
    if (optionC.Checked)
    {
        MessageBox.Show("Correct! The system with no solution is  $x + 2y = 4$ ,  $2x - y = 1$ .", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

        // Hide the first question components
        questionLabel.Visible = false;
        optionA.Visible = false;
        optionB.Visible = false;
        optionC.Visible = false;
        optionD.Visible = false;
        submitButton.Visible = false;

        //Showing and positioning Question 2 components
        question2Label.Location = questionLabel.Location;
        question2Label.Visible = true;
        answerTextBox.Visible = true;
        submit2Button.Visible = true;
    }
    else
    {
        MessageBox.Show("Incorrect. The correct answer is  $x + 2y = 4$ ,  $2x - y = 1$ .", "Result", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    }
}

```

The quiz consists of several interactive methods, like parsing input from the user:

```

private void submit2Button_Click(object sender, EventArgs e)
{
    string userInput = answerTextBox.Text;
    if (userInput.Trim() == "2")
    {
        // Code executed when the user input is correct
        MessageBox.Show("Correct! Gaussian Elimination Method is based on finding a triangular matrix by
        performing elementary row operations.", "Result", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

```

// Hide buttons, textbox and labels
    question2Label.Visible = false;
    answerTextBox.Visible = false;
    submit2Button.Visible = false;

// Adjust the position of question3Label and make it visible
    question3Label.Location = question2Label.Location;
    question3Label.Visible = true;
    step1Label.Visible = true;
    step2Label.Visible = true;
    step3Label.Visible = true;
    step4Label.Visible = true;
    drop1Label.Visible = true;
    drop2Label.Visible = true;
    drop3Label.Visible = true;
    drop4Label.Visible = true;

// Make the submit button for question 3 visible
    submit3Button.Visible = true;

    }
    else
    {
        // Code executed when the condition is false
        MessageBox.Show("Incorrect. The correct answer is Gaussian Elimination Method.", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

See APPENDIX (A) for the full quiz.cs code.

Another interactive method was used which involves dropping the correct answer in a respective order:

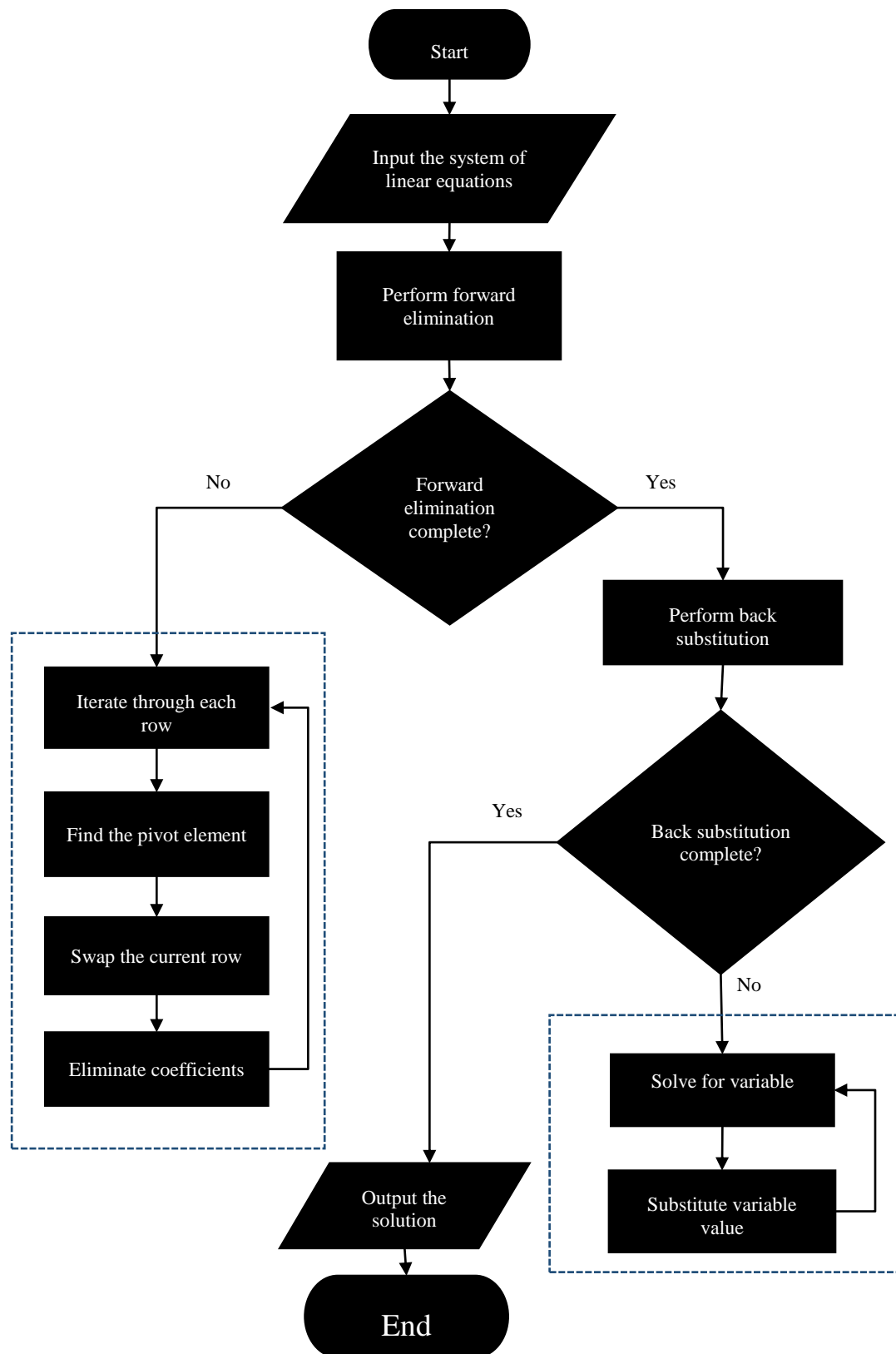
```
private void submit3Button_Click(object sender, EventArgs e)
{
    if (drop1Label.Text == "Form a matrix from the system of linear equations." &&
        drop2Label.Text == "Use row operations to transform the matrix into row echelon form." &&
        drop3Label.Text == "Solve the linear system using back-substitution." &&
        drop4Label.Text == "Interpret the solution.")
    {
        // Code executed when the condition is true
        MessageBox.Show("Correct! Well done!");

        // Hide buttons and labels
        question3Label.Visible = false;
        step1Label.Visible = false;
        step2Label.Visible = false;
        step3Label.Visible = false;
        step4Label.Visible = false;
        drop1Label.Visible = false;
        drop2Label.Visible = false;
        drop3Label.Visible = false;
        drop4Label.Visible = false;
        submit3Button.Visible = false;

        // Adjust the position of question4Label and make it visible
        question4Label.Location = question3Label.Location;
        question4Label.Visible = true;
        option4A.Visible = true;
        option4B.Visible = true;
        option4C.Visible = true;
        systemImage1.Visible = true;
        submit4Button.Visible = true;

    }
    else
    {
        // Code executed when the condition is false
        MessageBox.Show("Incorrect. Please try again.");
    }
}
```

For the system linear solver form, the Gaussian Elimination method was implemented using C#, this is the block diagram of how the solver operates:



See APPENDIX B for the C# code used to implement the solver for system of linear equations, here is the algorithmic process of the code:

1. “*SolveLinearSystem*” Method:

- This method takes a matrix ‘*A*’ and a vector ‘*b*’ as input and returns an array ‘*x*’ containing the solution to the linear system.
- The method performs Gaussian elimination to solve the system.
- It starts by creating an augmented matrix ‘*Ab*’ using the ‘*AugmentedMatrix*’ method.
- Then it iterates through each row (‘*k*’) to perform Gaussian elimination steps:
- It finds the pivot row (‘*maxRow*’) by locating the row with the maximum absolute value in the current column.
- If necessary, it swaps the current row with the pivot row.
- It eliminates the coefficients below the pivot by subtracting multiples of the pivot row from the subsequent rows.
- After Gaussian elimination, it checks if the system has no solution or infinite solutions by examining the values in the augmented matrix ‘*Ab*’.
- Finally, it performs back-substitution to find the solution by solving for each variable starting from the last row and working upward.

- The method returns the solution array ‘*x*’

2. ‘*AugmentedMatrix*’ Method:

- This method takes a matrix ‘*A*’ and a vector ‘*b*’ and returns an augmented matrix ‘*Ab*’.

- It creates a new matrix ‘*Ab*’ of size 3x4.

3. *SolveButton\_Click*’ Event Handler:

- This method is triggered when the user clicks the “*Solve*” button.
- It retrieves the coefficient values from text boxes and populates the matrix ‘*A*’.
- It retrieves the constant values from text boxes and populates the vector ‘*b*’.

- It handles any format exception that may occur during parsing of input values.
- It calls the *'SolveLinearSystem'* method with matrix *'A'* and vector *'b'* as arguments.
- If the solution is not null, it displays the solution values in the *'solutionLabel'*.
- If the solution is null, it indicates that there is no solution or infinite solutions in the *'solutionLabel'*.



## CONCLUSIONS

System of linear equations are very important in mathematics which have many real world applications in several fields such as engineering, physics, computer science and many more. And having the ability to solve it is an important skill.

We have focused on this project to develop a software simulator on the topic of “System of Linear Equations” for distance learning course “Higher and Applied Mathematics” which is more of an interactive simulator with a user friendly interface.

The simulator consists of 3 main parts, the first part gives the user the necessary knowledge to learn about the theory of System of Linear Equations and their solving methods, Gaussian Method, Matrix Inversion, Cramer’s rule and Jacobi Method. The second part is a 10 questions quiz which consists of several interactive methods to interact with. The third part of the simulator is a systems linear equations solver, which is capable of accepting input in the form of coefficients (represented by  $a_{ij}$ ) and constants (represented by  $b_i$ ) and provide a solution.

The environment used to build the simulator is Visual Studio 2022 considering its easy deployment of objects and designing the simulator. And the code used was C# due to its simplicity in making algorithms into code.

## REFERENCES

1. "Numerical Linear Algebra" by Lloyd N. Trefethen and David Bau III ISBN: 978-0898719574
2. "Numerical Methods for Engineers" by Steven C. Chapra & Raymond P. Canale ISBN: 978-0073397924
3. "Numerical Recipes: The Art of Scientific Computing" by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery ISBN: 978-1548919078
4. "C# 9.0 in a Nutshell: The Definitive Reference" by Joseph Albahari and Ben Albahari ISBN: 978-1098100919
5. Albahari, J., Albahari, B. (2020). C# 9.0 in a Nutshell: The Definitive Reference. O'Reilly Media. ISBN-13: 978-1098101044.
6. Schildt, H. (2021). C# 10.0: The Complete Reference. McGraw-Hill Education. ISBN-13: 978-1260466802.
7. Liberty, J. (2021). Learning C# by Developing Games with Unity. O'Reilly Media. ISBN-13: 978-1492072299.
8. Deitel, P., Deitel, H. (2021). C# 10 and .NET 6: Modern Cross-Platform Development. Pearson. ISBN-13: 978-0136600410.
9. Troelsen, A., Japikse, P. (2021). Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming. Apress. ISBN-13: 978-1484277584.
10. Griffiths, D. F., Higham, D. J. (2017). Numerical Methods for Ordinary Differential Equations: Initial Value Problems. Springer. ISBN-13: 978-3319569408.
11. Anton, H., Rorres, C. (2015). Elementary Linear Algebra: Applications Version. Wiley. ISBN-13: 978-1118474228.
12. Trefethen, L. N., Bau III, D. (1997). Numerical Linear Algebra. SIAM. ISBN-13: 978-0898713619.

13. Lay, D. C. (2016). *Linear Algebra and Its Applications*. Pearson. ISBN-13: 978-0321982384.
14. Stoer, J., Bulirsch, R. (2013). *Introduction to Numerical Analysis*. Springer. ISBN-13: 978-3662447412.
15. Golub, G. H., Van Loan, C. F. (2012). *Matrix Computations*. Johns Hopkins University Press. ISBN-13: 978-1421407944.

## APPENDIX A

C# code for the Quiz Form (QuizForm.cs)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.Design.AxImporter;

namespace SystemLinearEquations
{
    public partial class QuizForm : Form
    {
        public QuizForm()
        {
            InitializeComponent();
            drop1Label.AllowDrop = true;
            drop2Label.AllowDrop = true;
            drop3Label.AllowDrop = true;
            drop4Label.AllowDrop = true;
            this.AutoScaleMode = AutoScaleMode.Font;
        }

        private void submitButton_Click(object sender, EventArgs e)
        {
            if (optionC.Checked)
            {
                MessageBox.Show("Correct! The system with no solution is  $x + 2y = 4$ ,  $2x - y = 1$ .",
                    "Result", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
    }
}
```

```

// Hide the first question components
questionLabel.Visible = false;
optionA.Visible = false;
optionB.Visible = false;
optionC.Visible = false;
optionD.Visible = false;
submitButton.Visible = false;

//Showing and positioning Question 2 components
question2Label.Location = questionLabel.Location;
question2Label.Visible = true;
answerTextBox.Visible = true;
submit2Button.Visible = true;
}
else
{
    MessageBox.Show("Incorrect. The correct answer is  $x + 2y = 4$ ,  $2x - y = 1$ .", "Result",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}
private void submit2Button_Click(object sender, EventArgs e)
{
    string userInput = answerTextBox.Text;
    if (userInput.Trim() == "2")
    {
        MessageBox.Show("Correct! Gaussian Elimination Method is based on finding a triangular
matrix by performing elementary row operations.", "Result", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        question2Label.Visible = false;
        answerTextBox.Visible = false;
        submit2Button.Visible = false;
        question3Label.Location = question2Label.Location;
        question3Label.Visible = true;
        step1Label.Visible = true;
        step2Label.Visible = true;
        step3Label.Visible = true;
    }
}

```

```
        step4Label.Visible = true;
        drop1Label.Visible = true;
        drop2Label.Visible = true;
        drop3Label.Visible = true;
        drop4Label.Visible = true;
        submit3Button.Visible = true;
    }
    else
    {
        MessageBox.Show("Incorrect. The correct answer is Gaussian Elimination Method.",
"Result", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void step1Label_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        step1Label.DoDragDrop(step1Label.Text, DragDropEffects.Move);
    }
}

private void step2Label_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        step2Label.DoDragDrop(step2Label.Text, DragDropEffects.Move);
    }
}

private void step3Label_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        step3Label.DoDragDrop(step3Label.Text, DragDropEffects.Move);
    }
}
```

```
    }  
  }  
  private void step4Label_MouseDown(object sender, MouseEventArgs e)  
  {  
    if (e.Button == MouseButtons.Left)  
    {  
      step4Label.DoDragDrop(step4Label.Text, DragDropEffects.Move);  
    }  
  }  
  
  private void drop1Label_DragDrop(object sender, DragEventArgs e)  
  {  
    drop1Label.Text = (string)e.Data.GetData(DataFormats.Text);  
  }  
  private void drop2Label_DragDrop(object sender, DragEventArgs e)  
  {  
    drop2Label.Text = (string)e.Data.GetData(DataFormats.Text);  
  }  
  private void drop3Label_DragDrop(object sender, DragEventArgs e)  
  {  
    drop3Label.Text = (string)e.Data.GetData(DataFormats.Text);  
  }  
  private void drop4Label_DragDrop(object sender, DragEventArgs e)  
  {  
    drop4Label.Text = (string)e.Data.GetData(DataFormats.Text);  
  }  
  
  private void drop1Label_DragEnter(object sender, DragEventArgs e)  
  {  
    if (e.Data.GetDataPresent(DataFormats.Text))  
    {  
      e.Effect = DragDropEffects.Move;  
    }  
  }  
}
```

```

    }
}
private void drop2Label_DragEnter(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.Text))
    {
        e.Effect = DragDropEffects.Move;
    }
}
private void drop3Label_DragEnter(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.Text))
    {
        e.Effect = DragDropEffects.Move;
    }
}
private void drop4Label_DragEnter(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.Text))
    {
        e.Effect = DragDropEffects.Move;
    }
}
private void submit3Button_Click(object sender, EventArgs e)
{
    if (drop1Label.Text == "Form a matrix from the system of linear equations." &&
        drop2Label.Text == "Use row operations to transform the matrix into row echelon form."
&&
        drop3Label.Text == "Solve the linear system using back-substitution." &&
        drop4Label.Text == "Interpret the solution.")
    {
        MessageBox.Show("Correct! Well done!");
        question3Label.Visible = false;
        step1Label.Visible = false;
        step2Label.Visible = false;
    }
}

```



```

step3Label.Visible = false;
step4Label.Visible = false;
drop1Label.Visible = false;
drop2Label.Visible = false;
drop3Label.Visible = false;
drop4Label.Visible = false;
submit3Button.Visible = false;
question4Label.Location = question3Label.Location;
question4Label.Visible = true;
option4A.Visible = true;
option4B.Visible = true;
option4C.Visible = true;
systemImage1.Visible = true;
submit4Button.Visible = true;

}
else
{
    MessageBox.Show("Incorrect. Please try again.");
}
}
private void submit4Button_Click(object sender, EventArgs e)
{
    if (option4A.Checked)
    {
        MessageBox.Show("Correct! This system is consistent and independent.", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

        submit4Button.Visible = false;
        systemImage1.Visible = false;
        systemImage2.Visible = true;
        submit5Button.Visible = true;

    }
    else

```

```

    {
        MessageBox.Show("Incorrect. The correct answer is 'Consistent, independent'.", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }

}

private void submit5Button_Click(object sender, EventArgs e)
{
    if (option4C.Checked)
    {
        MessageBox.Show("Correct! This system is inconsistent.", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        submit5Button.Visible = false;
        systemImage2.Visible = false;
        systemImage3.Visible = true;
        submit6Button.Visible = true;
    }
    else
    {
        MessageBox.Show("Incorrect. The correct answer is 'inconsistent'.", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void submit6Button_Click(object sender, EventArgs e)
{
    if (option4B.Checked)
    {
        MessageBox.Show("Correct! This system is consistent, dependent.", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        submit6Button.Visible = false;
        systemImage3.Visible = false;
        option4A.Visible = false;
        option4B.Visible = false;
        option4C.Visible = false;
        question4Label.Visible = false;
        question5Label.Location = question4Label.Location;
    }
}

```

```

question5Label.Visible = true;
option5A.Visible = true;
option5B.Visible = true;
option5C.Visible = true;
option5D.Visible = true;
submit7Button.Visible = true;
}
else
{
    MessageBox.Show("Incorrect. The correct answer is 'Consistent, dependent'.", "Result",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}
private void submit7Button_Click(object sender, EventArgs e)
{
    if (option5B.Checked)
    {
        MessageBox.Show("Correct! The solution is x = 3, y = 1.5.", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        question5Label.Visible = false;
        option5A.Visible = false;
        option5B.Visible = false;
        option5C.Visible = false;
        option5D.Visible = false;
        submit7Button.Visible = false;
        question6Label.Location = question5Label.Location;
        question6Label.Visible = true;
        option6A.Visible = true;
        option6B.Visible = true;
        option6C.Visible = true;
        option6D.Visible = true;
        submit8Button.Visible = true;
    }
    else
    {
        MessageBox.Show("Incorrect. The correct answer is x = 3, y = 1.5.", "Result",

```

```

MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void submit8Button_Click(object sender, EventArgs e)
{
    if (option6C.Checked)
    {
        MessageBox.Show("Correct! The solution is (c.)", "Result", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        question6Label.Visible = false;
        option6A.Visible = false;
        option6B.Visible = false;
        option6C.Visible = false;
        option6D.Visible = false;
        submit8Button.Visible = false;
        question7Label.Location = question6Label.Location;

        question7Label.Visible = true;
        option7A.Visible = true;
        option7B.Visible = true;
        option7C.Visible = true;
        option7D.Visible = true;
        submit9Button.Visible = true;
    }
    else
    {
        MessageBox.Show("Incorrect. The correct answer is (c.)", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void submit9Button_Click(object sender, EventArgs e)
{
    if (option7A.Checked)
    {
        MessageBox.Show("Correct! The solution is (a.)", "Result", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
}

```

```

question7Label.Visible = false;
option7A.Visible = false;
option7B.Visible = false;
option7C.Visible = false;
option7D.Visible = false;
submit9Button.Visible = false;
question8Label.Location = question7Label.Location;
question8Label.Visible = true;
option8A.Visible = true;
option8B.Visible = true;
option8C.Visible = true;
option8D.Visible = true;
submit10Button.Visible = true;
}
else
{
    MessageBox.Show("Incorrect. The correct answer is (a.)", "Result",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void submit10Button_Click(object sender, EventArgs e)
{
    if (option8A.Checked)
    {
        MessageBox.Show("Correct! The solution is (a.)", "Result", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        question8Label.Visible = false;
        option8A.Visible = false;
        option8B.Visible = false;
        option8C.Visible = false;
        option8D.Visible = false;
        submit10Button.Visible = false;
        question9Label.Location = question8Label.Location;
        question9Label.Visible = true;
        option9A.Visible = true;
    }
}

```

```

    option9B.Visible = true;
    option9C.Visible = true;
    option9D.Visible = true;
    submit11Button.Visible = true;
}
else
{
    MessageBox.Show("Incorrect. The correct answer is (a.)", "Result",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void submit11Button_Click(object sender, EventArgs e)
{
    if (option9B.Checked)
    {
        MessageBox.Show("Correct! The solution is (b.)", "Result", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        question9Label.Visible = false;
        option9A.Visible = false;
        option9B.Visible = false;
        option9C.Visible = false;
        option9D.Visible = false;
        submit11Button.Visible = false;
        question10Label.Location = question9Label.Location;
        question10Label.Visible = true;
        checkBox10A.Visible = true;
        checkBox10B.Visible = true;
        checkBox10C.Visible = true;
        checkBox10D.Visible = true;
        submit12Button.Visible = true;
    }
    else
    {
        MessageBox.Show("Incorrect. The correct answer is (b.)", "Result",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

```
    }  
  }  
  
  private void submit12Button_Click(object sender, EventArgs e)  
  {  
    if (checkBox10A.Checked && checkBox10B.Checked && checkBox10C.Checked &&  
!checkBox10D.Checked)  
    {  
      MessageBox.Show("Correct! Well done!", "Result", MessageBoxButtons.OK,  
MessageBoxIcon.Information);  
      checkBox10A.Visible = false;  
      checkBox10B.Visible = false;  
      checkBox10C.Visible = false;  
      checkBox10D.Visible = false;  
      submit12Button.Visible = false;  
      question10Label.Visible = false;  
      congrats.Visible = true;  
    }  
    else  
    {  
      MessageBox.Show("Incorrect. Please try again.", "Result", MessageBoxButtons.OK,  
MessageBoxIcon.Warning);  
    }  
  }  
}
```

## APPENDIX B

C# code for the Solver Form (SolverForm.cs)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SystemLinearEquations
{
    public partial class SolverForm : Form
    {
        public SolverForm()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private double[] SolveLinearSystem(double[,] A, double[] b)
        {
            // Augmented matrix creation
            double[,] Ab = AugmentedMatrix(A, b);

            // Gaussian elimination (Iterate through each row)
            for (int k = 0; k < 3; k++)
            {
                // Find pivot row
                int maxRow = k;
                double maxVal = Ab[k, k];
                for (int i = k + 1; i < 3; i++)
                {
                    if (Math.Abs(Ab[i, k]) > Math.Abs(maxVal))
                    {
                        maxRow = i;
                        maxVal = Ab[i, k];
                    }
                }

                // Swap the current row
                if (maxRow != k)
                {
                    for (int j = k; j < 4; j++)

```



```

    {
        double temp = Ab[k, j];
        Ab[k, j] = Ab[maxRow, j];
        Ab[maxRow, j] = temp;
    }
}

// Eliminate coefficients
for (int i = k + 1; i < 3; i++)
{
    double factor = Ab[i, k] / Ab[k, k];
    for (int j = k; j < 4; j++)
    {
        Ab[i, j] -= factor * Ab[k, j];
    }
}

// Check if the system has no solution or infinite solutions
if (Ab[2, 3] != 0 && (Ab[2, 0] == 0 && Ab[2, 1] == 0 && Ab[2, 2] == 0))
{
    return null;
}

// Back-substitute to find solution
double[] x = new double[3];
for (int i = 2; i >= 0; i--)
{
    double sum = 0;
    for (int j = i + 1; j < 3; j++)
    {
        sum += Ab[i, j] * x[j];
    }
    x[i] = (Ab[i, 3] - sum) / Ab[i, i];
}

return x;
}

private double[,] AugmentedMatrix(double[,] A, double[] b)
{
    double[,] Ab = new double[3, 4];
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            Ab[i, j] = A[i, j];
        }
        Ab[i, 3] = b[i];
    }
}

```

```

    return Ab;
}

private void SolveButton_Click(object sender, EventArgs e)
{
    double[,] A = new double[3, 3];
    double[] b = new double[3];

    // User coefficient input
    try
    {
        A[0, 0] = double.Parse(textBox11.Text);
        A[0, 1] = double.Parse(textBox12.Text);
        A[0, 2] = double.Parse(textBox13.Text);
        A[1, 0] = double.Parse(textBox21.Text);
        A[1, 1] = double.Parse(textBox22.Text);
        A[1, 2] = double.Parse(textBox23.Text);
        A[2, 0] = double.Parse(textBox31.Text);
        A[2, 1] = double.Parse(textBox32.Text);
        A[2, 2] = double.Parse(textBox33.Text);
    }
    catch (FormatException)
    {
        MessageBox.Show("Invalid input. Please enter numeric values for all coefficients.");
        return;
    }

    // User vector input
    try
    {
        b[0] = double.Parse(textBoxB1.Text);
        b[1] = double.Parse(textBoxB2.Text);
        b[2] = double.Parse(textBoxB3.Text);
    }
    catch (FormatException)
    {
        MessageBox.Show("Invalid input. Please enter numeric values for all constants.");
        return;
    }

    // Call function
    double[] solution = SolveLinearSystem(A, b);
    if (solution != null)
    {
        solutionLabel.Text = string.Format("x = {0:F3}, y = {1:F3}, z = {2:F3}", solution[0],
solution[1], solution[2]);
    }
    else
    {

```

```
        solutionLabel.Text = "No solution or infinite solutions.";
    }
}
}
```