

Implementation of Gaussian Elimination Method for Solving System of Linear Equations using Visual Basic

Abubakar Muhammad and Zainab Aliyu Musa

Computer Science Department

Federal Polytechnic, Bauchi

Nigeria

faqeer@gwanisoftware.com and zynerb@gmail.com

Janet Folasade Taiwo

Department of Science Education,

Faculty of Technology Education, ATBU

Bauchi Nigeria janettaiwo01@gmail.com

Abstract

Simultaneous equations are used in solving many algebra word problems and its application is enormously used in different fields of human endeavours. Different methods are used in solving system of simultaneous linear equations out of these methods there is Gaussian elimination method. This method is named after a Mathematician Carl Friedrich Gauss (1777–1855), ever since it has been used to solve system of simultaneous linear equation. This research intends to study this method of solving system of simultaneous linear equations and attempt to design a software that will handle all the procedures taken in that method to provide solution. Visual Basic is used to design the program that will solve a system of 3 X 3 simultaneous equations. Scope of this research is restricted to system of simultaneous linear equations having three equations and three unknowns (3 X 3). The program converts the three equations into matrices and then conducts the elementary row operation and produces diagonal matrix, from it, the solution is then calculated. This software can be used in teaching Gaussian elimination procedures in class by simulating different problems with it. Time and human efforts will be saved in manual conducting of Gaussian elimination procedure, by the inception of this software and research.

Keywords: Gaussian elimination, simultaneous equation, Visual Basic, Gauss, algebra.

1. INTRODUCTION

Gaussian elimination (also known as row reduction) is an algorithm for solving systems of linear equations. It is usually understood as a sequence of operations performed on the associated matrix of coefficients. This method can also be used to find the rank of a matrix, to calculate the determinant of a matrix, and to calculate the

inverse of an invertible square matrix. The method is named after Carl Friedrich Gauss (1777–1855).

To perform row reduction on a matrix, one uses a sequence of elementary row operations to modify the matrix until the lower left-hand corner of the matrix is filled with zeros, as much as possible. There are three types of elementary row operations: 1) Swapping two rows, 2) Multiplying a row by a non-zero number, 3) Adding a multiple of one row to another row. Using these operations, a matrix can always be transformed into an upper triangular matrix, and in fact one that is in row echelon form. Once all of the leading coefficients (the left-most non-zero entry in each row) are 1, and in every column containing a leading coefficient has zeros elsewhere, the matrix is said to be in reduced row echelon form.

This final form is unique; in other words, it is independent of the sequence of row operations used. For example, in the following sequence of row operations (where multiple elementary operations might be done at each step), the third and fourth matrices are the ones in row echelon form, and the final matrix is the unique reduced row echelon form.

1.1 System of Simultaneous Linear Equations

What is a system of simultaneous linear equations? First, ask —what is a linear equation? It is an equation in one or more variables where each term's degree is not more than 1. That means a variable x may appear, but neither any higher power of x , such as x^2 , nor any product of variables, such as xy , may appear. It has to be a pretty simple equation like

$$3x + 2y - 5z = 8.$$

In fact, any linear equation can be put in the form

$$c_1x_1 + c_2x_2 + \dots + c_nx_n = c_0.$$

Where n is the number of variables, the variables are x_1, x_2, \dots, x_n , and c_0, c_1, \dots, c_n are constants.

A system is just a collection of such linear equations, and to solve a system look for the values of the variables which make all the equations true simultaneously. For instance, if x and y are the variables, then an example system of linear equations is $5x - 2y = 4$ and $x + 2y = 8$

There are various ways of solving this system, and they lead to the unique solution where $x = 2$ and $y = 3$. We'll look next at a common algorithm for solving systems of simultaneous equations called elimination.

1.2 Solutions for simultaneous linear equations above two

The solution method known as Gauss elimination has two stages. In the first stage the equations are replaced by a system of equations having the same solution but which are in triangular form.

In the second stage the new system is solved by back-substitution. The first step is to write the equations in matrix form. This gives:

$$\left(\quad \right) \left(\quad \right) \left(\quad \right)$$

Then for conciseness we combine the matrix of coefficients with the column vector of right-hand sides to produce the augmented matrix

1.3 Gaussian Elimination Method

The Gaussian elimination method is used to solve systems of three or more equations. There are two ways to implement this method.

First Method

1. Transform the equations of a system so that in each equation there will be one unknown less than the previous.

$$+ \begin{cases} 3x + 2y + z = 1 \\ 5x + 3y + 4z = 2 \\ x + y - z = 1 \end{cases}$$

2. Take the equation with the coefficient of x: 1 or -1, as the first equation.

If not possible with x, do with y or z, and rearrange the order of the unknowns:

$$\begin{cases} x + y - z = 1 & 3x + 2y + z = 1 \\ 5x + 3y + 4z = 2 \end{cases}$$

3. Perform the elimination method with the 1st and 2nd equation to eliminate the term of x in the 2nd equation. Then, in the second equation, place the result of the operation:

$$E'2 = E2 - 3E1$$

4. Perform the same with the 1st and 3rd equation to eliminate the term of x:

$$E'3 = E3 - 5E1$$

$$5x + 3y + 4z = 2$$

$$\left\{ \begin{array}{l} -5x - 5y + 5z = -5 \\ -2y + 9z = -3 \end{array} \right.$$

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

$$\left\{ \begin{array}{l} x + y - z = 1 \\ -2y + 9z = -3 \end{array} \right.$$

5. Perform the elimination method with the 2nd and 3rd equation:

$$\mathbf{E'3 = E'3 - 2E'2}$$

$$\begin{array}{l} -2y \\ \underline{2y} \end{array} \left\{ \begin{array}{l} +9z = -3 \\ -8z = 4 \end{array} \right.$$

$$Z = 1$$

6. So, we obtain another equivalent system:

$$- \left\{ \begin{array}{l} X + y - z = 1 \\ y + 4z = -2 \\ z = 1 \end{array} \right.$$

7. Solve the system:

$$z = 1$$

$$-y + 4 \cdot 1 = -2 \quad y = 6$$

$$x + 6 - 1 = 1 \quad x = -4$$

Second Method

1. Transform the system into a matrix, in which place the coefficients of the variables and the independent terms (separated by a straight line).

Example:

$$\begin{array}{l} 3x + 2y + z = 1 \quad 5x + 3y + 4z \\ = 2x + y - z = 1 \end{array}$$

1.4 Matrix

In mathematics, a matrix (plural matrices) is a rectangular array [1]—of numbers, symbols, or expressions, arranged in rows and columns [2][3]—that is interpreted and manipulated in certain prescribed ways. One such way is to state the order of the matrix. For example, the order of the matrix below is 2×3 (read "two by three"), because there are two rows and three columns.

$$\begin{pmatrix} 1 & 9 & -13 \\ 20 & 5 & -6 \end{pmatrix}$$

The individual items in a matrix are called its elements or entries. [4] Provided that they are the same size (have the same number of rows and the same number of columns), two matrices can be added or subtracted element by element. The rule for matrix multiplication, however, is that two matrices can be multiplied only when the number of columns in the first equals the number of rows in the second. A major application of matrices is to represent linear transformations, that is, generalizations of linear functions such as $f(x) = 4x$. For example, the rotation of vectors in three dimensional space is a linear transformation which can be represented by a rotation matrix R : if v is a column vector (a matrix with only one column) describing the position of a point in space, the product Rv is a column vector describing the position of that point after a rotation. The product of two transformation matrices is a matrix that represents the composition of two linear transformations. Another application of matrices is in the solution of systems of linear equations. If the matrix is square, it is possible to deduce some of its properties by computing its determinant. For example, a square matrix has an inverse if and only if its determinant is not zero. Insight into the geometry of a linear transformation is obtainable (along with other information) from the matrix's eigenvalues and eigenvectors.

Applications of matrices are found in most scientific fields. In every branch of physics, including classical mechanics, optics, electromagnetism, quantum mechanics, and quantum electrodynamics, they are used to study physical phenomena, such as the motion of rigid bodies. In computer graphics, they are used to project a 3-dimensional image onto a 2-dimensional screen. In probability theory and statistics, stochastic matrices are used to describe sets of probabilities; for instance, they are used within the Page Rank algorithm that ranks the pages in a Google search.[5] Matrix calculus generalizes classical analytical notions such as derivatives and exponentials to higher dimensions.

Matrices can be used to compactly write and work with multiple linear equations, i.e., systems of linear equations. For example, if A is an m -by- n matrix, x designates a column vector (i.e., $n \times 1$ -matrix) of n variables x_1, x_2, \dots, x_n , and b is an $m \times 1$ -column vector, then the matrix equation

$$Ax = b$$

is equivalent to the system of linear equations

$$A_{1,1}x_1 + A_{1,2}x_2 + \dots + A_{1,n}x_n = b_1$$

$$A_{m,1}x_1 + A_{m,2}x_2 + \dots + A_{m,n}x_n = b_m \text{ .[24]}$$

1.5 Application of Gaussian Elimination in solving Simultaneous Linear System of Equations

Engineers often need to solve large systems of linear equations; for example in determining the forces in a large framework or finding currents in a complicated electrical circuit. The method of Gauss elimination provides a systematic approach to their solutions. Many other scientific areas rely on Gaussian Elimination method to provide solutions to scientific problems capable of being translated into simultaneous linear equations.

2. SYSTEM OF THREE SIMULTANEOUS LINEAR EQUATIONS

Systems of three simultaneous equations are three algebraic equations having three unknown variables which are to have their values computed during solution provision. Consider the system

$$\begin{aligned} x_1 + 3x_2 + 5x_3 &= 14 & \dots\dots\dots(1) & \quad 2x_1 - x_2 - \\ 3x_3 &= 3 & \dots\dots\dots(2) & \\ 4x_1 + 5x_2 - x_3 &= 7 & \dots\dots\dots(3) & \end{aligned}$$

Unknown variables in the three equations are x_1 , x_2 and x_3 .

2.1 Forming the coefficient matrix

The first step is to write the equations in matrix form. This gives:

$$\begin{bmatrix} 1 & 3 & 5 \\ 2 & -1 & -3 \\ 4 & 5 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 3 \\ 7 \end{bmatrix}.$$

Then for conciseness we combine the matrix of coefficients with the column vector of right-hand sides to produce the **augmented matrix**

$$\left[\begin{array}{ccc|c} 1 & 3 & 5 & 14 \\ 2 & -1 & -3 & 3 \\ 4 & 5 & -1 & 7 \end{array} \right]$$

2.2 Iterations

The next stage after reducing equations into matrices is iteration stage and is now accomplished by means of **row operations**. There are three possible operations:

- i. interchange two rows;
- ii. multiply or divide a row by a non-zero constant factor;
- iii. add to, or subtract from, one row a multiple of another row.

2.3 Solution matrix

Once the coefficient matrix is reduced into diagonal matrix, iteration comes to an end and a solution can be obtained by solving the last equation which is having only one unknown, then substitute its value into equation two to obtain the value of the second unknown. Finally, the values of the two unknown are then substituted into the first equation to obtain the value of the first unknown. This process is known as

backsubstitution. Some equations have infinite number of solutions and are therefore unstable, example is the equations shown below:

$$\begin{aligned}x_1 + x_2 - 3x_3 &= 3 \\2x_1 - 3x_2 + 4x_3 &= -4 \\x_1 - x_2 + x_3 &= -1\end{aligned}$$

3. DESIGN OF THE SOFTWARE

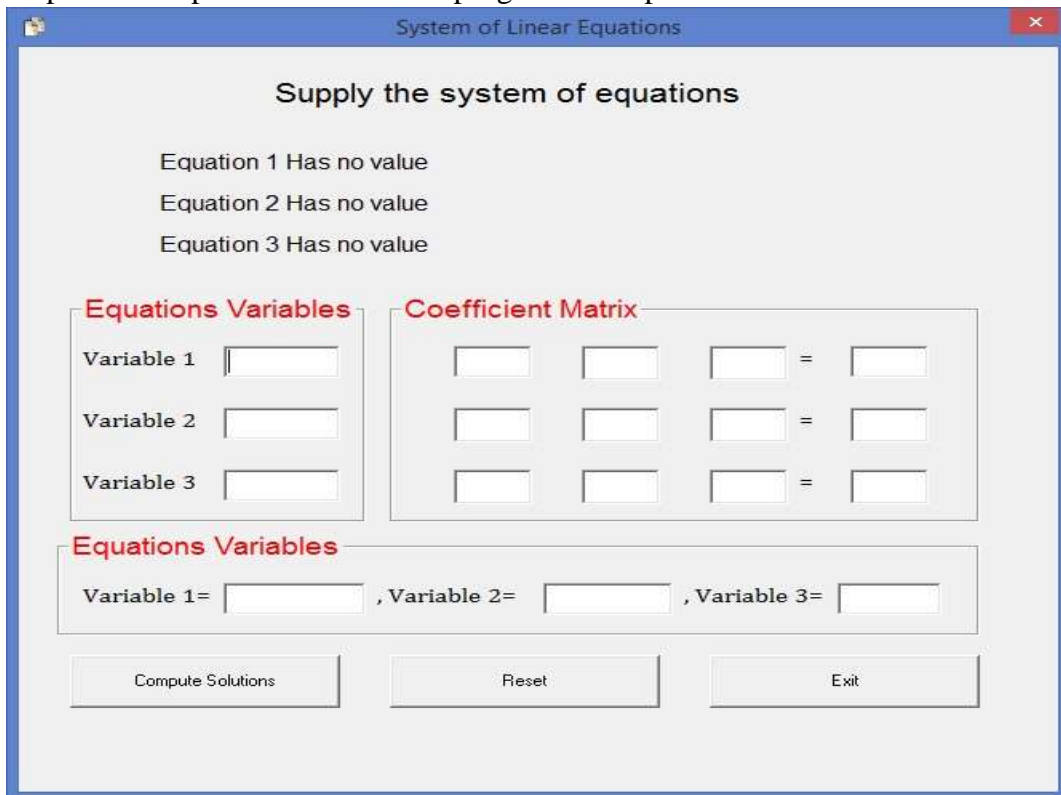
This research intends to design a program that will solve 3 X 3 systems of simultaneous linear equations using Gaussian Elimination procedure. This section examines the process of designing this program.

3.1 Choice of programming Language

Visual Basic is chosen for the design of the software due its GUI support.

3.2 Interfaces

The input and output interfaces of the program are captured and shown below.



System of Linear Equations

Supply the system of equations

$+1X+1Y+1Z = 6$
 $+1X-1Y+1Z = 2$
 $+1X+2Y-1Z = 2$

Equations Variables

Variable 1

Variable 2

Variable 3

Coefficient Matrix

<input type="text" value="+1"/>	<input type="text" value="+1"/>	<input type="text" value="+1"/>	=	<input type="text" value="6"/>
<input type="text" value="+1"/>	<input type="text" value="-1"/>	<input type="text" value="+1"/>	=	<input type="text" value="2"/>
<input type="text" value="+1"/>	<input type="text" value="+2"/>	<input type="text" value="-1"/>	=	<input type="text" value="2"/>

Equations Variables

X = , Y = , Z =

Compute Solutions

Reset

Exit

System of Linear Equations

Supply the system of equations

$+1X+1Y+1Z = 6$
 $+1X-1Y+1Z = 2$
 $+1X+2Y-1Z = 2$

Equations Variables

Variable 1

Variable 2

Variable 3

Coefficient Matrix

<input type="text" value="+1"/>	<input type="text" value="+1"/>	<input type="text" value="+1"/>	=	<input type="text" value="6"/>
<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="-2"/>	=	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="-4"/>	=	<input type="text" value="-4"/>

Equations Variables

X = , Y = , Z =

Compute Solutions

Reset

Exit

Fig. 3 Diagonal matrix with solution

3.3 Testing of system

The designed system is tested with system of equations with known solutions and is found to be working well. However, it was observed that in some systems of equations alternate solutions were obtained from the system apart from the ones manually computed.

6. CONCLUSION

This research is able to successfully designed a program that solve system of simultaneous equations of three unknowns using Gaussian Elimination.

7. ACKNOWLEDGEMENTS

All consulted references are duly acknowledged.

8. REFERENCES

Gaussian Elimination retrieved from

https://en.wikipedia.org/wiki/Gaussian_elimination Gaussian

Elimination retrieved from

http://www.ditutor.com/system/gaussian_elimination.html

Matrix retrieved from [https://en.wikipedia.org/wiki/Matrix_\(mathematics\)](https://en.wikipedia.org/wiki/Matrix_(mathematics))

Simultaneous retrieved from

<http://aleph0.clarku.edu/~djoyce/ma105/simultaneous.html>

Appendix A

Source Code

```
Dim Done As Boolean
Private Sub Command1_Click()
Done = True
'changing Row two
Dim Temp(20) As Double
```

```

If (Val(Txt21.Text) * Val(Txt31.Text)) > 0 Then
    Temp(1) = Val(Txt31.Text) * Val(Txt21.Text) - Val(Txt21.Text) *
Val(Txt31.Text)
Else
    Temp(1) = Val(Txt31.Text) * Val(Txt21.Text) + Val(Txt21.Text) *
Val(Txt31.Text)
End If
If (Val(Txt22.Text) * Val(Txt31.Text)) > 0 Then
    Temp(2) = Val(Txt32.Text) * Val(Txt21.Text) - Val(Txt22.Text) *
Val(Txt31.Text)
Else
    Temp(2) = Val(Txt32.Text) * Val(Txt21.Text) + Val(Txt22.Text) *
Val(Txt31.Text)
End If
If (Val(Txt23.Text) * Val(Txt31.Text)) > 0 Then
    Temp(3) = Val(Txt33.Text) * Val(Txt21.Text) - Val(Txt23.Text) *
Val(Txt31.Text)
Else
    Temp(3) = Val(Txt33.Text) * Val(Txt21.Text) + Val(Txt23.Text) *
Val(Txt31.Text)
End If
If (Val(Txt24.Text) * Val(Txt31.Text)) > 0 Then
    Temp(4) = Val(Txt34.Text) * Val(Txt21.Text) - Val(Txt24.Text) *
Val(Txt31.Text)
Else

```

```

Val(Txt21.Text)      Val(Txt24.Text)
Val(Txt31.Text)
  End If
  Txt21 = Temp(1)
  Txt22 = Temp(2)
  Txt23 = Temp(3)
  Txt24 = Temp(4)
  'changing A31
  If (Val(Txt11.Text) * Val(Txt31.Text)) > 0 Then
    Temp(1) = Val(Txt31.Text) * Val(Txt11.Text) - Val(Txt11.Text) *
Val(Txt31.Text)
  Else
    Temp(1) = Val(Txt31.Text) * Val(Txt11.Text) + Val(Txt11.Text) *
Val(Txt31.Text)
  End If
  If (Val(Txt12.Text) * Val(Txt31.Text)) > 0 Then
    Temp(2) = Val(Txt32.Text) * Val(Txt11.Text) - Val(Txt12.Text) *
Val(Txt31.Text)
  Else
    Temp(2) = Val(Txt32.Text) * Val(Txt11.Text) + Val(Txt12.Text) *
Val(Txt31.Text)
  End If
  If (Val(Txt13.Text) * Val(Txt31.Text)) > 0 Then
    Temp(3) = Val(Txt33.Text) * Val(Txt11.Text) - Val(Txt13.Text) *
Val(Txt31.Text)
  Else
    Temp(3) = Val(Txt33.Text) * Val(Txt11.Text) + Val(Txt13.Text) *
Val(Txt31.Text)
  End If
  If (Val(Txt14.Text) * Val(Txt31.Text)) > 0 Then
    Temp(4) = Val(Txt34.Text) * Val(Txt11.Text) - Val(Txt14.Text) *
Val(Txt31.Text)
  Else

```

```

Val(Txt11.Text)      Val(Txt14.Text)
Val(Txt31.Text)
  End If
  Txt31 = Temp(1)
  Txt32 = Temp(2)
  Txt33 = Temp(3)
  Txt34 = Temp(4)
  'changing A32
  If (Val(Txt21.Text) * Val(Txt32.Text)) > 0 Then
    Temp(1) = Val(Txt31.Text) * Val(Txt22.Text) - Val(Txt21.Text) *
Val(Txt32.Text)
  Else
    Temp(1) = Val(Txt31.Text) * Val(Txt22.Text) + Val(Txt21.Text) *
Val(Txt32.Text)
  End If
  If (Val(Txt22.Text) * Val(Txt32.Text)) > 0 Then
    Temp(2) = Val(Txt32.Text) * Val(Txt22.Text) - Val(Txt22.Text) *
Val(Txt32.Text)
  Else
    Temp(2) = Val(Txt32.Text) * Val(Txt22.Text) + Val(Txt22.Text) *
Val(Txt32.Text)
  End If
  If (Val(Txt23.Text) * Val(Txt32.Text)) > 0 Then
    Temp(3) = Val(Txt33.Text) * Val(Txt22.Text) - Val(Txt23.Text) *
Val(Txt32.Text)
  Else
    Temp(3) = Val(Txt33.Text) * Val(Txt22.Text) + Val(Txt23.Text) *
Val(Txt32.Text)
  End If
  If (Val(Txt24.Text) * Val(Txt32.Text)) > 0 Then
    Temp(4) = Val(Txt34.Text) * Val(Txt22.Text) - Val(Txt24.Text) *
Val(Txt32.Text)
  Else

```

```

Val(Txt22.Text) Val(Txt24.Text)
Val(Txt32.Text)
End If
Txt31 = Temp(1)
Txt32 = Temp(2)
Txt33 = Temp(3)
Txt34 = Temp(4)
Temp(1) = Val(Txt34.Text) / Val(Txt33.Text)
Temp(2) = Temp(1) * Val(Txt23.Text)
If Temp(2) < 0 Then
    Temp(2) = (Val(Txt24.Text) + (-1 * Temp(2))) / Val(Txt22.Text)
Else
    Temp(2) = (Val(Txt24.Text) - Temp(2)) / Val(Txt22.Text)
End If
Temp(3) = Val(Txt12.Text) * Temp(2) + Temp(1) * Val(Txt13.Text)
If Temp(3) < 0 Then
    Temp(3) = (Val(Txt14.Text) + (-1 * Temp(3))) / Val(Txt11.Text)
Else
    Temp(3) = (Val(Txt14.Text) - Temp(3)) / Val(Txt11.Text)
End If
TxtSol1 = Temp(1)
TxtSol2 = Temp(2)
TxtSol3 = Temp(3)
End Sub
Private Sub Command2_Click()
Dim YesExit As String
YesExit = MsgBox("Are you sure you want to quit", vbQuestion + vbYesNo,
"Gaussian Elimination Method")
If YesExit = vbYes Then
    End
End If
End Sub

```

```

Private Sub Command3_Click()
Done = False
Txt11 = ""
Txt12 = ""
Txt13 = ""
Txt14 = ""
Txt21 = ""
Txt22 = ""
Txt23 = ""
Txt24 = ""
Txt31 = ""
Txt32 = ""
Txt32 = ""
Txt33 = ""
Txt34 = ""
End Sub

Private Sub Timer1_Timer()
If Done Then Exit Sub
If TxtVar1 <> "" Then
    Label13.Caption = TxtVar1 & " = "
    If (Txt11 <> "") Then
        LblEqns1.Caption = Txt11 & TxtVar1
    End If
    If (Txt21 <> "") Then
        LblEqns2.Caption = Txt21 & TxtVar1
    End If
    If (Txt31 <> "") Then
        LblEqns3.Caption = Txt31 & TxtVar1
    End If
End If
If TxtVar2 <> "" Then

```

```
Label12.Caption = "," & TxtVar2 & " = "  
If (Txt12 <> "") Then  
    LblEqns1.Caption = LblEqns1 & Txt12 & TxtVar2  
End If  
If (Txt22 <> "") Then  
    LblEqns2.Caption = LblEqns2 & Txt22 & TxtVar2  
End If  
If (Txt32 <> "") Then  
    LblEqns3.Caption = LblEqns3 & Txt32 & TxtVar2  
End If  
End If  
If TxtVar3 <> "" Then  
    Label11.Caption = "," & TxtVar3 & " = "  
    If (Txt13 <> "") Then  
        LblEqns1.Caption = LblEqns1 & Txt13 & TxtVar3  
    End If  
    If (Txt23 <> "") Then  
        LblEqns2.Caption = LblEqns2 & Txt23 & TxtVar3  
    End If  
    If (Txt33 <> "") Then  
        LblEqns3.Caption = LblEqns3 & Txt33 & TxtVar3  
    End If  
End If  
If (Txt14 <> "") Then  
    LblEqns1.Caption = LblEqns1 & " = " & Txt14  
End If  
If (Txt24 <> "") Then  
    LblEqns2.Caption = LblEqns2 & " = " & Txt24  
End If  
If (Txt34 <> "") Then  
    LblEqns3.Caption = LblEqns3 & " = " & Txt34 End If
```

```

End Sub
Private Sub Txt11_LostFocus()
Dim k As Double k =
Val(Txt11.Text)
If (k > 0) And (Mid(Txt11, 1, 1) <> "+") Then Txt11 = "+" & Txt11
End Sub
Private Sub Txt12_Change()
If (Txt12 <> "") And (Txt11 = "") Then
MsgBox "You must first supply the first coefficient before supplying the second!",
vbCritical + vbOKOnly, "Gaussian Elimination Method"
Txt12 = ""
Txt11.SetFocus
End If
End Sub
Private Sub Txt12_LostFocus()
Dim k As Double k =
Val(Txt12.Text)
If (k > 0) And (Mid(Txt12, 1, 1) <> "+") Then Txt12 = "+" & Txt12
End Sub
Private Sub Txt13_Change()
If (Txt13 <> "") And (Txt12 = "") Then
Txt13 = ""
Txt12.SetFocus
MsgBox "You must first supply the second coefficient before supplying the third!",
vbCritical + vbOKOnly, "Gaussian Elimination Method"
End If
End Sub
Private Sub Txt13_LostFocus()
Dim k As Double k =
Val(Txt13.Text)
If (k > 0) And (Mid(Txt13, 1, 1) <> "+") Then Txt13 = "+" & Txt13
End Sub

```



```

Private Sub Txt14_Change()
If (Txt14 <> "") And (Txt13 = "") Then
    Txt14 = ""
    Txt13.SetFocus
    MsgBox "You must first supply the coefficient of the equation before supplying the
solution vector!", vbCritical + vbOKOnly, "Gaussian Elimination Method"
End If
End Sub
Private Sub Txt21_LostFocus()
Dim k As Double k =
Val(Txt21.Text)
If (k > 0) And (Mid(Txt21, 1, 1) <> "+") Then Txt21 = "+" & Txt21
End Sub
Private Sub Txt22_Change()
If (Txt22 <> "") And (Txt21 = "") Then
    MsgBox "You must first supply the first coefficient before supplying the second!",
vbCritical + vbOKOnly, "Gaussian Elimination Method"
    Txt22 = ""
    Txt21.SetFocus
End If
End Sub
Private Sub Txt22_LostFocus()
Dim k As Double k =
Val(Txt22.Text)
If (k > 0) And (Mid(Txt22, 1, 1) <> "+") Then Txt22 = "+" & Txt22
End Sub
Private Sub Txt23_Change()
If (Txt23 <> "") And (Txt22 = "") Then
    Txt23 = ""
    Txt22.SetFocus
    MsgBox "You must first supply the second coefficient before supplying the third!",
vbCritical + vbOKOnly, "Gaussian Elimination Method"

```

```

End If
End Sub
Private Sub Txt23_LostFocus()
Dim k As Double k =
Val(Txt23.Text)
If (k > 0) And (Mid(Txt23, 1, 1) <> "+") Then Txt23 = "+" & Txt23
End Sub
Private Sub Txt24_Change()
If (Txt24 <> "") And (Txt23 = "") Then
    Txt24 = ""
    Txt23.SetFocus
    MsgBox "You must first supply the coefficient of the equation before supplying the
solution vector!", vbCritical + vbOKOnly, "Gaussian Elimination Method"
End If
End Sub
Private Sub Txt31_LostFocus()
Dim k As Double k =
Val(Txt31.Text)
If (k > 0) And (Mid(Txt31, 1, 1) <> "+") Then Txt31 = "+" & Txt31
End Sub
Private Sub Txt32_Change()
If (Txt32 <> "") And (Txt31 = "") Then
    MsgBox "You must first supply the first coefficient before supplying the second!",
vbCritical + vbOKOnly, "Gaussian Elimination Method"
    Txt32 = ""
    Txt31.SetFocus
End If
End Sub
Private Sub Txt32_LostFocus()
Dim k As Double
k = Val(Txt32.Text)

```

```

If (k > 0) And (Mid(Txt32, 1, 1) <> "+") Then Txt32 = "+" & Txt32
End Sub
Private Sub Txt33_Change()
If (Txt33 <> "") And (Txt32 = "") Then
    Txt33 = ""
    Txt32.SetFocus
    MsgBox "You must first supply the second coefficient before supplying the third!",
vbCritical + vbOKOnly, "Gaussian Elimination Method"
End If
End Sub
Private Sub Txt33_LostFocus()
Dim k As Double k =
Val(Txt33.Text)
If (k > 0) And (Mid(Txt33, 1, 1) <> "+") Then Txt33 = "+" & Txt33
End Sub
Private Sub Txt34_Change()
If (Txt134 <> "") And (Txt33 = "") Then
    Txt34 = ""
    Txt33.SetFocus
    MsgBox "You must first supply the coefficient of the equation before supplying the
solution vector!", vbCritical + vbOKOnly, "Gaussian Elimination Method"
End If
End Sub
Private Sub TxtVar2_Change()
If (TxtVar2 <> "") And (TxtVar1 = "") Then
    MsgBox "You must first supply the first variable before supplying the second!",
vbCritical + vbOKOnly, "Gaussian Elimination Method"
    TxtVar2 = ""
    TxtVar1.SetFocus
End If
End Sub
Private Sub TxtVar3_Change()

```

```
If (TxtVar3 <> "") And (TxtVar2 = "") Then
    MsgBox "You must first supply the second variable before supplying the third!",
vbCritical + vbOKOnly, "Gaussian Elimination Method"
    TxtVar3 = ""
    TxtVar2.SetFocus
End If
End Sub
```