**EEE 206 - PROGRAMMING**

**PROJECT REPORT**

**PROJECT NO:3**

**Linear Equations**

Submitted by:

*Ahmet Berke Karaca (EEE)*

*Rumeysa Türkmen (EEE)*

*Zeynep Sude Koç (EEE)*

Project Mentor:

Dr. Doğan Çörüş

Faculty of Engineering and Natural Sciences

Kadir Has University

Spring 2022

**TABLE OF CONTENTS**

[1 INTRODUCTION 3](#_Toc104755649)

[1.1 Project Definition 3](#_Toc104755650)

[1.2 Linear Equations 3](#_Toc104755651)

[2 LEVEL-1 5](#_Toc104755652)

[3 LEVEL-2 8](#_Toc104755653)

[3.1 Variables 8](#_Toc104755654)

[3.2 Scanning the First Equation 8](#_Toc104755655)

[3.3 Solving Two Variables Linear Equation 9](#_Toc104755656)

[3.4 Solving Three Variables Linear Equation 11](#_Toc104755657)

[3.5 Simulations 12](#_Toc104755658)

[4 CONLUSION 16](#_Toc104755659)

[REFERENCES 17](#_Toc104755660)

# 1 INTRODUCTION

## 1.1 Project Definition

The goal in this project is to create a program that solves a system of linear equations. What is expected from the code in this program is that our program reads the file, decodes the relevant system and reports the values ​​of each variable. In the program we created, the lines of code are written in detail to solve the problem step by step and to find the value of each variable. In the Level-1 and Level-2 sections, the working stages of the code are presented in detail.

## 1.2 Linear Equations

A linear equation is one in which the variable's maximum power is always 1. A linear equation's graph is always a straight line. A one-degree equation is another name for it. Ax + B = 0 is the standard form of a linear equation in one variable. x is a variable, A is a coefficient, and B is a constant in this equation. A linear equation with two variables is written in the standard form Ax + By = C. A and B are coefficients, and C is a constant, and x and y are variables.

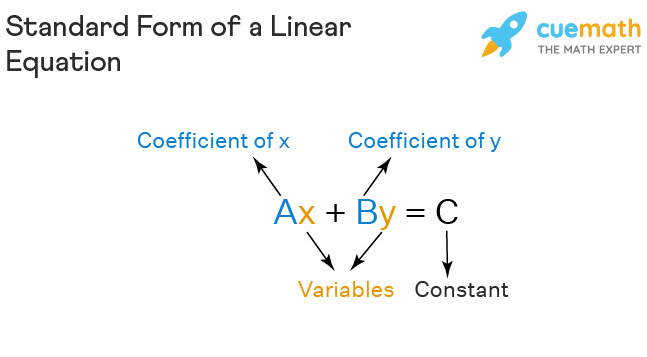


Figure 1.1 Standard form of a linear equation

A weighing balance with equal weights on both sides is what an equation is. It still holds true if we add or subtract the same integer from both sides of an equation. It is also correct to multiply or divide the same integer on both sides of an equation. We find the value of the unknown variable by bringing the variables to one side of the equation and the constant to the other. This is how to solve a one-variable linear equation.

While solving linear equations in this project, we adapted our code according to this logic; We multiplied one side with a (-) coefficient by adapting it according to the coefficient of the two unknown equations. Then, by adding these two equations, we are left with an equation with one unknown. For example, if x is found from the x and y variables, the other variable y is also found by substituting x in the equation. With this logic, we provided the solution of linear equations.

# 2 LEVEL-1

In this level of our project, we create a program which solves systems of linear equations. We assume that the system has a unique solution, and our code solves linear equations with two variables.

As can be seen in equations 2.1 and 2.2, our program gets an input for coefficients a1, b1, c1 for the first equation, after user entered those, our program asks user to enter the coefficients a2, b2, c2.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.1 Defining the variables and getting inputs for the first equation

In figure 2,1 you can see we defined the variables as a float. After that, we wrote down the code for the getting input from the user. In the last line in figure 2,1, our code prints out the equation that the user created.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.2 Simulation of the first equation

It can be seen that in figure 2.2, when the code is run, firstly our program asks for the coefficients of first linear equation. In our example we chose them 3, 1 and 2, respectively. After that, program shows it as a linear equation form.

In the next step in our code, we followed the same steps for the second linear equation. This part of the code can be seen in figure 2.3

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.3 Code for the second linear equation

When the user continues to run the code, the program will ask to enter coefficients of second linear equation.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.4 Simulation of the second linear equation

Figure 2.4 shows that what happens when the user enter the coefficients of the second linear equation. As we mentioned, until there we followed the same steps for the both equations.

After that we defined x and y values in terms of coefficients. The logic is stands for simple algebra. This is shown in figure 2.5.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.5 Defining x, y and equations

Additionally, we defined first and second equations to show the user as a float.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.6 Last step of our code

Lastly, we printed out *eq1, eq2* and *x, y* values. Simulation of the final version of our sample, can be seen in figure 2.7.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.7 Last version of the simulation

As can be seen in figure 2.7, our program solves the linear equations in a form of shown in equation 2.1 and 2.2.

# 3 LEVEL-2

In this part of the report, we will introduce the level-2 code. Our level-2 code is capable to solve two and three variables linear equations successfully. On the other hand, it also satisfies the level-2 requirement which is reading the input data of the system from a txt file. In addition, our code also can determine if there is no unique solution for the given system of linear equations for two variable equations.

## 3.1 Variables

We start with defining our variables. Since we will have fractions after the mathematical operations, we define our values as floats. And we decided to name them as:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.1 Defining the variables

As we can see in the figure above, we name the coefficients of the linear equations first. Next, we define our variables for the sub equations we will obtain from the three variable linear equations. And, lastly, we define our variables in the related equations as x, y, and z.

## 3.2 Scanning the First Equation

After this process is done, we will scan the equation in the first line of the file. By scanning this equation, we will determine that if our system is neither a two variable or three variable one. If the system is a three variable one, all the coefficients of the first equation (and the other two equations’ coefficients too) must be other than 0. That means regardless of saving an equation with two or three variables, the users should load the equation to be entered into the file in the form ax+by+cz=d. If the system has a bivariate structure, the coefficient of one of the variables must be zero. Depends on the existence a coefficient 0 in the first equation the code will determine if it is two or three variables and will process the required solution.

So, we scan the first line equation first

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.2 Scanning the first equation

As we can see from the code, we use *fseek* and *fscanf* functions respectively. What the code does in the code is to point the necessary float coefficient in the equation with a cursor by managing the flow in the file with the *fseek* function. Then, after each necessary *fseek* step, the required float coefficient is assigned to the previously defined coefficient variables with the *fscanf* function.

## 3.3 Solving Two Variables Linear Equation

After it is done, we will use if statement to determine if there is any coefficient 0 in the equations. In other words, we will see if the system is a two variable or three variable one. Then, If it is a two variable, we will determine which coefficient is 0. This will make us capable to solve all possible two variable linear equations in the given file. Because all linear equations require different solution in terms of multiplications of the different coefficients.

So, the required code is

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.3 Soluntion for a two variable equation system

As it can be seen from the code, The code determines if there is any 0 coefficient first, and it solves the equation if c1 is 0 for the equation. For the rest of the code, of course, the other possible equations’ (where a1 or b1 is 0) solutions are made by the code.

Besides, the code also understands if the equation has a unique solution. Or not by checking the coefficients ratios. For example, for c1 is 0, if a1/a2 and b1/b2 ratios are not equal, the system has a unique solution. If the a1/a2, b1/b2 and d1/d2 ratios are all equal, the system has infinitely many solutions. And, lastly, if while a1/a2 - b1/b2 ratios are equal and a1/a2 - d1/d2 ratios are not equal, the system has no solution. So, the code will be

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure X.4 Coding the possible results for two variable equations

## 3.4 Solving Three Variables Linear Equation

If there is no 0 coefficient in the first equation, the system is a three variables equation. So, under an *else* statement, we will eliminate the variable z by using the coefficient formulas and obtain two variables sub equations. After that, we will solve the linear equations for these sub equations. Lastly, we will find the variables x, y and z. The required code will be following:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.5 Soluntion for a three variable equation system

## 3.5 Simulations

In this part, we will test our code with our txt file. We will test our code with a two variable equation first.

* 1.2x-2.5y+0z=12
* 10x+12.5y+0z=120

So, our txt file is

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.6 txt file for the first simulation of a two variable equation system

If we run our code, here is what we obtain

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.7 The console result for the first simulation

Now, let’s see what we will obtain in the console when the system has many solutions. To do that we will make our txt file as

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.8 txt file for the second simulation of a two variable equation system

So, in the console, we obtain:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.9 Infinitely many solution result on the console

As we can see from figure 3.9, the code gave the console result as infinitely many solutions.

Now, we will test our code with three variable equations. We modify our txt file as

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure X.10 txt file for the simulation of a three variable equation system

So, the console is

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.11 The console result for the three variable linear equation system

As we can see from the last console, again, the console printed our equations’ result successfully.

# 4 CONLUSION

In this project, we created a program that solves a system of linear equations. We have completed the functions that our 1st and 2nd level code should provide (our program reads the file, decodes the relevant system and reports the values ​​of each variable) and presented our problem-solving stages in detail.

# REFERENCES

[1] Linear equations - definition, formula, graph, examples. Cuemath. (n.d.). Retrieved May 23, 2022, from <https://www.cuemath.com/algebra/linear-equations/>