SPL-1 Project Report

Equation Solver

Submitted by

Sakib Al Hasan

BSSE Roll No: 1209

Exam Roll No: 60926

BSSE Session: 2019-20

Submitted to

Dr. Naushin Nower

Associate Professor

Institute of Information Technology

Institute of Information Technology

University of Dhaka

[30-05-2022]

Table of Contents

1. Introduction

Equation solver solves mathematical equations. There are huge equations in mathematics, solve all equations in a big project. So, I made an equation solver that can solve three kinds of equations they are,

(i) linear algebra

(ii) linear programming

(iii) polynomial equation

These types of equations are commonly used in our everyday life. It takes input from the user and solves the equation using some algorithm and prints the result. This program takes input from the I/O console and prints it to the I/O console. This program takes input equations from users in any format. This program can help students find the root of these equations and also helps people to find the maximum and minimum values of linear programming.

2. Background of the project

To understand this project, some prior study was necessary. And should know some topics that are related to this project. Without these topics, it is very troublesome to understand this project.

2.1 Determinant of Matrix

The determinant is defined as a scalar value that is associated with the square matrix. We find to determine when the matrix is only square. We use the row reduction method to find the determination of a matrix, meaning all values of lower triangular become 0. Then multiplication of the diagonal elements of a matrix we find the determined value.

2.2 Cramer's rule

Cramer's rule is a specific formula used for solving a system of linear equations containing as many equations as unknowns, efficient whenever the system of the linear equation has a unique solution. The solution obtained using Cramer's rule will be in terms of the determinants of the coefficient matrix and matrices obtained from it by replacing one column with the column vector of the right-hand sides of the equations.

2.3 Simplex Method

A simplex method is an approach to solving linear programming models by hand using slack variables, tableaus, and pivot variables as a means of finding the optimal solution to an optimization problem. Simplex tableau is used to perform row operations on the linear programming model as well as for checking optimality. By using this method we can find the maximum and minimum values of linear programming. It is used when we calculate 3 or upper dimensional equations.

2.4 Bairstow Method

Bairstow's method is an efficient algorithm for finding the roots of a real polynomial of arbitrary degree. It is to use Newton's method to adjust the coefficients *u* and *v* in the quadratic until its roots are also roots of the polynomial being solved. The roots of the quadratic may then be determined, and the polynomial may be divided by the quadratic to eliminate those roots. This process is then iterated until the polynomial becomes quadratic or linear, and all the roots have been determined.

3. Project Description

3.1 Separate number and variable

Users can input equations in any format. So, we should tokenize this number and variable and the power of the variable. Then store it in a matrix array for calculation and find the solution of linear algebra.

3.2 Solution of linear algebra

After tokenizing numbers and variables, we make a matrix and find determine this matrix. Then replace one column of the matrix with its constant vector and also find determine of this matrix. We found a determined value for each variable. Now we apply Cramer's rule to find the unique solution to these equations.

3.3 Infinite/ No solution exits

If the determined value of the first matrix is equal to zero then there are infinitely many solutions or no solution exists. Now we check the constant vector, if the constant vector is a multiplicand of the first matrix then there are infinitely many solutions exits. Otherwise, there are no solution exists.

3.4 Standard form

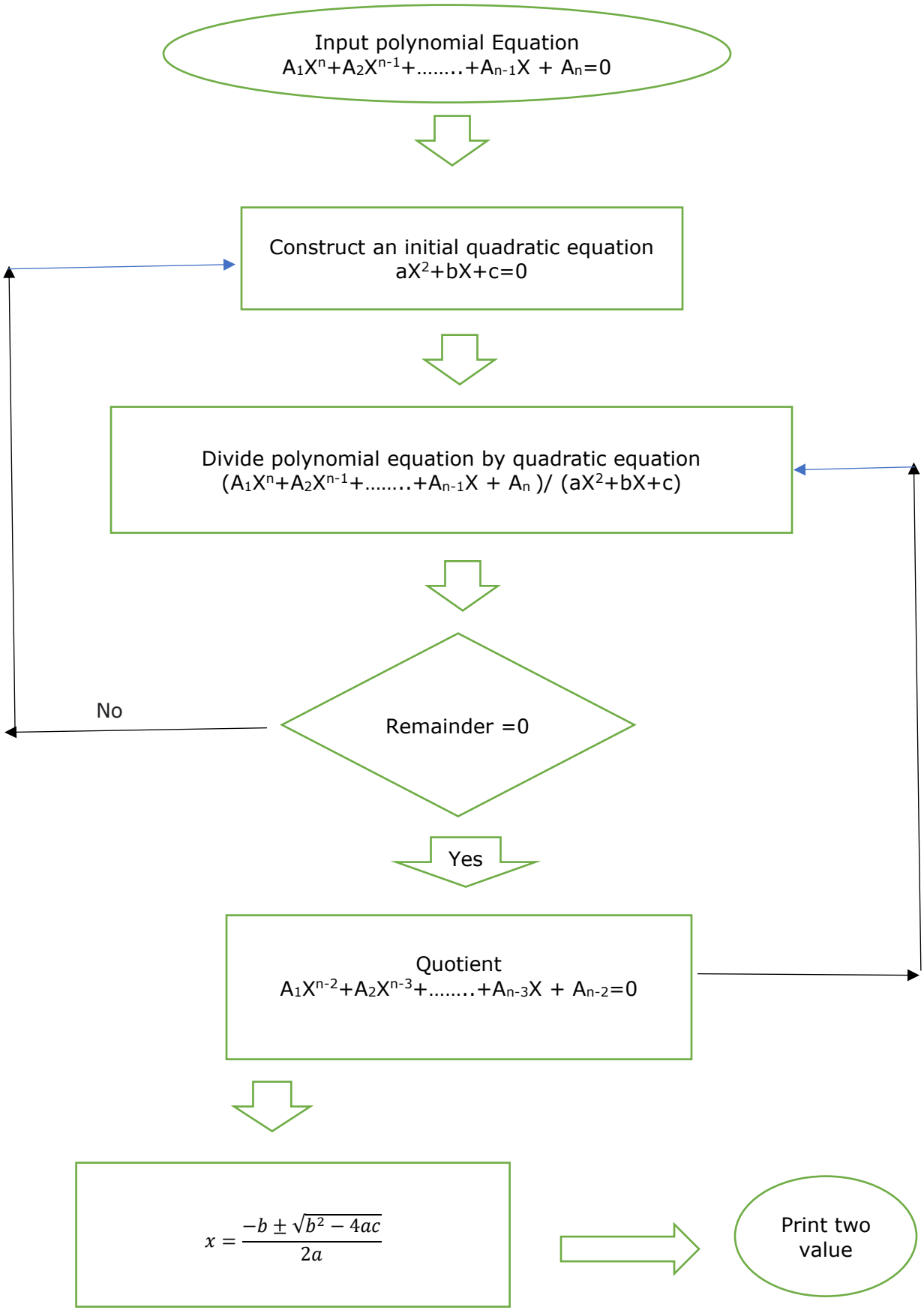
To solve linear programming we need to convert linear programming equations into standard form. We add slug variable if there are less than(<) sign, if there are greater than(>) sign we subtract append value. If there are negative variables we convert/replace them with positive variables.

3.5 Solution of linear programming

After converting linear programming to standard form we make a table and calculate this table with the key row and key column. Also, handle slug variables and append variables. There is also an objective function, if it is a minimum value then it becomes negative and finds a minimum value.

3.6 Polynomial equation

We use Bairstow's method for solving a polynomial equation. To solve the polynomial equation we use a quadratic equation. We divide the main equation with the quadratic equation, if the remainder is equal to zero then we find two roots from the quadratic equation and the main equation will be lower two degrees. We again use the new equation as the main equation and find a new quadratic equation which is also the root of the main equation. We proceed with this until our equation becomes quadratic or with less power. Here is the flowchart of this process.


Input polynomial Equation
$A_1X^n+A_2X^{n-1}+\ldots\ldots..+A_{n-1}X + A_n=0$
Construct an initial quadratic equation
$aX^2+bX+c=0$
Divide polynomial equation by quadratic equation
$(A_1X^n+A_2X^{n-1}+\ldots\ldots..+A_{n-1}X + A_n)/ (aX^2+bX+c)$
Remainder =0
No
Yes
Quotient
$A_1X^{n-2}+A_2X^{n-3}+\ldots\ldots..+A_{n-3}X + A_{n-2}=0$
$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
Print two value

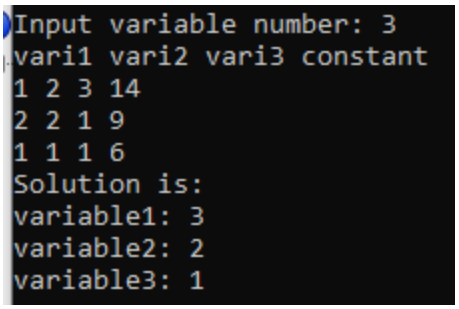
4. Implement and Testing

I use C++ language to implement this project. I never use the built-in library function in this project rather than use my own library. I use project in code blocks to gather all code in the project so that no file becomes large. Here are three testing parts for linear algebra.

First linear algebra, we use Cramer's rule for finding unique solutions. We use the row reduction method for finding determine the matrix. Here are some code snippets for finding determination.

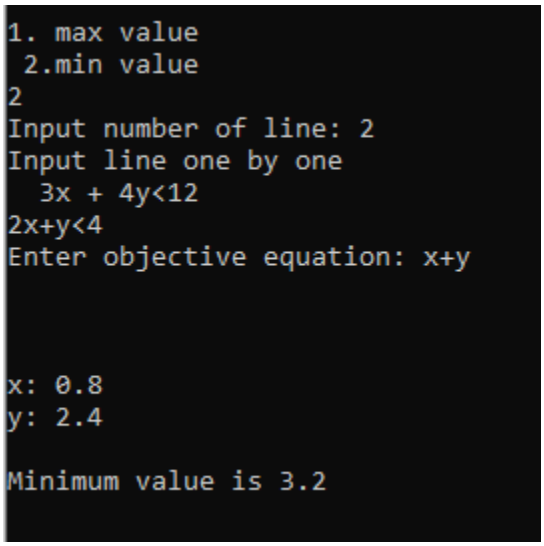
```
for(size_t i=0;i<number;i++)
   {
      for(size_t j=number-1;j>i;j--)
      {
         if(arr[j][i] == 0){
            continue;
         }
         else{
            if(arr[j-1][i] == 0){
               //exchange row
               for(size_t x =0;x<number;x++){
                  double temp = arr[j][x];
                  arr[j][x] = arr[j-1][x];
                  arr[j-1][x] = temp;
               }
               continue;
            }

           double req_ratio = arr[j][i] / arr[j-1][i];

            for(size_t k =0;k<number;k++){
               arr[j][k] = arr[j][k] - req_ratio * arr[j-1][k];
            }
         }
      }
   }
```

Some output of linear algebra:



Find solution of three value

In the polynomial equation, we use the simplex method and covert the equation into standard form. Then using tableau we find the max or min value and find the critical value of these equations.

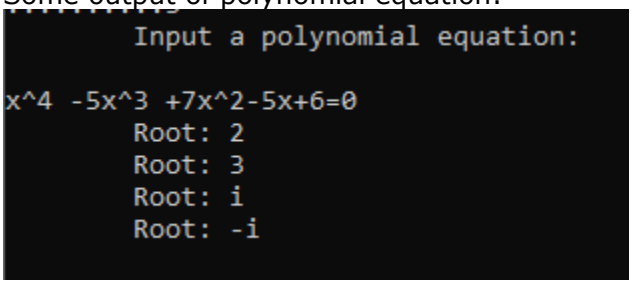


Find minimum value

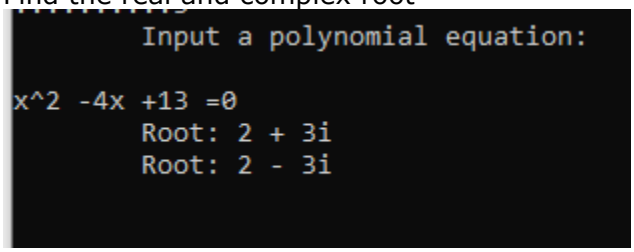
In the polynomial equation, we use the Bairstow method to find all real roots and also find all rational roots. We make a quadratic equation from the main equation and find a solution from the quadratic equation. Here is a code snippet for finding solution from a quadratic equation.

```
if(determine<0){
        cout<<"\tRoot: "<<(determine/(2*x))<<"i"<<endl;
         cout<<"\tRoot: "<<(determine/(2*x))<<"-i"<<endl;
}
else{
                //there are coefficient
        cout<<"\tRoot: "<<((-p)/(2*x))<<" + "<<(determine/(2*x))<<"i"<<endl;
        cout<<"\tRoot: "<<((-p)/(2*x))<<" - "<<(determine/(2*x))<<"i"<<endl;
}
```

Some output of polynomial equation:



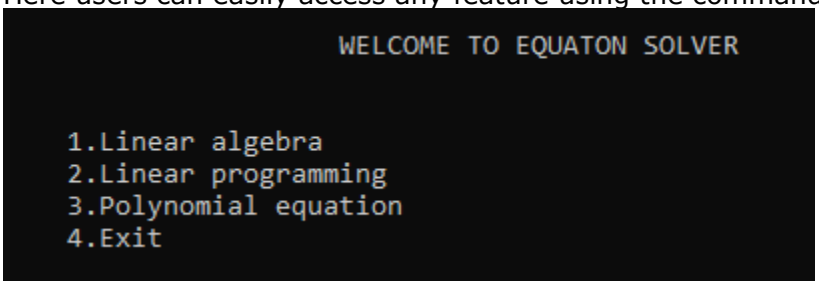
Find the real and complex root



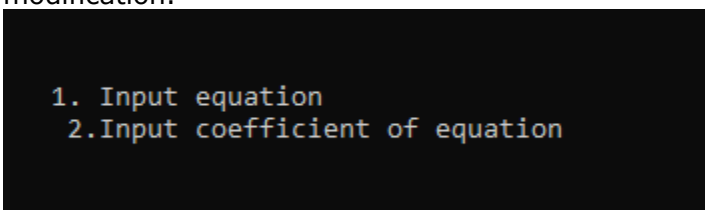
Find only complex roots

5. User Interface

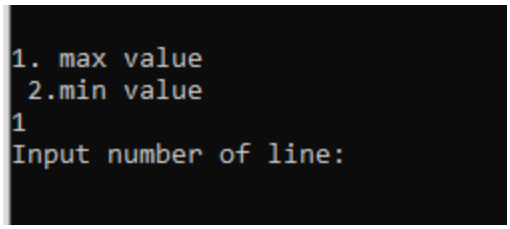
Here users can easily access any feature using the command line.



In linear programming, the user can input a matrix or whole equation without any modification.



In linear programming, there are options for finding max or min values.



The output results are printed in the console which is shown in the previous.

6. Challenges Faced

The Challenge of the project is when this program takes input from the user and its needs to separate variables and numbers. We need to tokenize this variable and number, and also find the power of each variable when solving the polynomial equation.
In linear algebra there we need to find determine the matrix and use Cramer's rule for finding the roots of these equations. It also tries to find if there are infinitely many solutions or if no solution exists.
In the polynomial equation, there are we use the Bairstow algorithm for finding read and rational roots and removing epsilon error from roots.

In linear programming, there is use simplex method for finding max or min value and maintaining a table. For creating a table we need to handle metrics. We also need to convert these equations into standard form.

7. Conclusion

There are many equations in mathematics, and my project try solves some selective equations. I try my best to find solutions properly these equations. I use different types of

equations to test this solution. All tests have become successful and it makes a strong proof that this project worked successfully.

I tried to learn and implement everything from scratch. Working on this project has helped me have a better idea about windows and network programming. I have fulfilled all the functionalities that I was committed to from the very beginning of the project. Being new to this programming paradigm, this SPL has challenged my limits. I enjoyed learning on the go and implementing it. I am happy that I could complete the project successfully and have taken on the challenges. I find myself more confident while solving yet the unseen problem, thinking critically, and being more confident in my future projects.

Reference

(i) Bairstow method: https://math.iitm.ac.in/public_html/sryedida/caimna/transcendental/polynomial%20methods/brs%20method.html

(ii) Polynomial equation : https://www.calculatorsoup.com/calculators/math/math-equation-solver.php

(iii) Linear programming: https://en.wikipedia.org/wiki/Simplex_algorithm

(iv) Linear algebra: https://www.analyzemath.com/linear-algebra/determinants/find-determinant-using-row-reduction.html