NUMERICAL METHOD

Project by

Aman Shaw	*12017009001029*
Kaushik Dey	*12017009001099*
Atanu Das	*12017009001328*
Nitesh Fatepuria	*12017009001416*
Somdeep Jana	*12017009001009*

University Of Engineering & Management

Section: 2A Year: 2019 Semester: 4th

TABLE OF CONTENTS

No.	Content Name	Page No.
01.	Introduction & Objective	3
02.	Project Environment ○ Hardware Environment ○ Software Environment ○ Running Environment	4
03.	Software Development	5-8
	○ Creating Project ○ Program Flow	5
	○ Operations Algorithm	6 - 8
	○ Bisection Method	6 - 7
	○ Newton-Raphson Method	7
	○ Regula-Falsi Method	7 - 8
04.	Conclusion	9
05.	References	10

Introduction

It is an offline application where we can get the possible root(s) of a function. This is a menu driven program. Three type of operation we can perform in this program namely Bisection Method, Newton-Raphson Method, Regula-Falsi Method. This program also shows every steps in every operation. It also helps us to compare runtime of all the operations and determine which one of them is fastest and accurate. All of this operation will mainly give assume values.

Objective

As a software developer our main objective is to build a program that is very easy to use and give a complete overview of all the process. This program is created for those functions for which to get a root we have to take multiple number of assumptions. For some functions, getting the near right value of a root may be very difficult for human, but for this program it is a matter of moment.

Project Environment

To build this software first we have to setup a favorable environment for testing and developing flawlessly. Below we list up The Hardware and software environment.

Hardware Environment:

CPU	:	Ryzen 5 2600X 3.6 GHz
RAM	:	2 x 8 GB 3200 MHz
Storage	:	250 GB NVMeM.2 SSD
GPU	:	RTX 2070 8GB GDDR6

Software Environment:

OS	:	Windows 10 Build 17763
DevSoft	:	MS Visual Studio 2017
Database	:	MS Access 2007

Running Requirement:

CPU	:	Dual Core 1.4 GHz or Higher
RAM	:	1 GB or more
Storage	:	at least 50 MB free storage
OS	:	Windows 95 or Higher

Software Development

There are many steps involve in development of this Program. The very first step is to creating the project file in DevC++.

Creating Project:

1. *Open DevC++*
2. *Click File-> New-> Project*
3. *Select Type c file-> Enter Filename.*
4. *Press OK*

Program Flow:

Before moving to creating the project any further we have to determine what the program will do and how one operation will come one after another.

1. *Ask user for correction decimal point*
2. *Move to a new screen*
3. *Present all the operations to perform and ask the user to choose an option.*
   a. *If Bisection Method is selected the go to new screen for Bisection Method*
   b. *If Regula-Falsi Method is selected the go to new screen for Regular-Falsi Method*
   c. *If Newton-Raphson selected then go to new screen for* Newton-Raphson.
4. *After this function return to main menu.*

Operations Algorithm:

There are three different algorithm attach to all this operations, those are all discussed below

Bisection Method:

*The method is also called the interval halving method, the binary search method or the dichotomy method. This method is used to find root of an equation in a given interval that is value of 'x' for which f(x) = 0.The method is based on The Intermediate Value Theorem which states that if f(x) is a continuous function and there are two real numbers a and b such that f(a)*f(b) 0 and f(b) < 0), then it is guaranteed that it has at least one root between them.*

***Assumptions**:*

1. *f(x) is a continuous function in interval [a, b]*
2. *f(a) * f(b) < 0*

***Steps**:*

1. *Find middle point c= (a + b)/2 .*
2. *If f(c) == 0, then c is the root of the solution.*
3. *Else f(c) != 0*
4. *If value f(a)*f(c)<0 then root lies between a and c.Else If f(b)*f(c) < 0 then root lies between b and c. So we recur b and c.*

5. *Else given function doesn't follow one of assumptions.*

Newton-Raphson Method:

Given a function f(x) on floating number x and an initial guess for root, find root of function in interval. Here f(x) represents algebraic or transcendental equation.

Starting from initial guess x1, the Newton Raphson method uses below formula to find next value of x, i.e., xn+1 from previous value xn.

$$X_{n+1} = X_n - \frac{F(x_n)}{F'(x_n)}$$

***Steps**:*

Input: initial x, func(x), derivFunc(x)
Output: Root of Func()

1. *Compute values of func(x) and derivFunc(x) for given initial x*
2. *Compute h: h = func(x) / derivFunc(x)*
3. *While h is greater than allowed error ε*
4. *h = func(x) / derivFunc(x)*
5. *x = x – h*

Regula-Falsi Method:

*Given a function f(x) on floating number x and two numbers 'a' and 'b' such that f(a)*f(b) < 0 and f(x) is continuous in [a, b]. Here f(x) represents*

algebraic or transcendental equation. Find root of function in interval [a, b] (Or find a value of x such that f(x) is 0).

***Steps**:*

1. *Write equation of the line connecting the two points.*
   a. *y – f(a) = ((f(b)-f(a))/(b-a))*(x-a)*
   b.
   c. *Now we have to find the point which touches x axis.*
   d. *For that we put y = 0.*
   e.
   f. *so x = a - (f(a)/(f(b)-f(a))) * (b-a)*
   g. *x = (a*f(b) - b*f(a)) / (f(b)-f(a))*
   h.
   i. *This will be our c that is c = x.*
2. *If f(c) == 0, then c is the root of the solution.*
3. *Else f(c) != 0*
4. *If value f(a)*f(c) < 0 then root lies between a and c. So we recur for a and c*
5. *Else If f(b)*f(c) < 0 then root lies between b and c. So we recur b and c.*
6. *Else given function doesn't follow one of assumptions.*

Conclusion

This program is an attempt to overcome the human inefficiency of getting a root for an unusual function in efficient and time consuming way. No human can run a loop for more than 1000 time in a moment but a computer can. This program use very limited memory space and storage space. Running system's limitation are the only limitation, this program is having. Lastly we want to say this programs is the proof that the main reason for creating a computer is calculation.

References

To make this project we gather lots of information from many places and in this section we like to acknowledge all of their helps. All the information is with the author's permission.

1. *https://www.geeksforgeeks.org/program-for-bisection-method/*
2. *https://www.geeksforgeeks.org/program-for-newton-raphson-method/*
3. *https://www.geeksforgeeks.org/program-for-method-of-false-position/*