# ALGORITHM FOR MATHEMATICAL SIMPLIFICATION

**A mini project report submitted in partial fulfilment of the requirement for the Award of the Degree of**

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

***by***

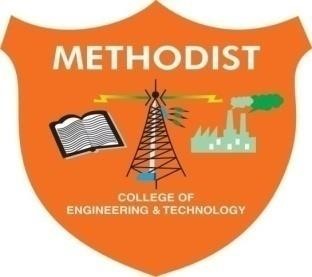
**Satyam Sharma (160717733061)**

**L.Sai Prakash (160717733062)**

**K.Dhanush Reddy (160717733106)**

***Under the Guidance of***

**Dr. R. Ch. A. Naidu, Professor, CSE**.



**Department of Computer Science and Engineering Methodist College of Engineering and Technology**

**King Koti, Abids, Hyderabad-500001.**

# 2018-2019

**Methodist College of Engineering and Technology King Koti, Abids, Hyderabad-500001.**

**Department of Computer Science and Engineering**



# DECLARATION BY THE CANDIDATE

We, **Satyam sharma (160717733061), L.Sai Prakash(160717733062) and K.Dhanush Reddy(160717733106)** students of Methodist College of Engineering and Technology, pursuing Bachelor’s degree in Computer Science and Engineering, here by declare that this mini project report entitled **“ALGORITHM FOR MATHEMATICAL SIMPLIFICATION”,** carried out under the guidance of Dr**.R.Ch.A.Naidu** submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science. This is a record work carried out by us and the results embodied in this project have not been reproduced /copied from any source

|  |  |
| --- | --- |
| **Satyam sharma(160717733061)** |  |
| **L.Sai Prakash (160717733062)** |  |
| **K.Dhanush Reddy(160717733106)** |  |

**Methodist College of Engineering and Technology**

**King Koti, Abids, Hyderabad-500001.**

## Department of Computer Science and Engineering



# CERTIFICATE BY THE SUPERVISOR

This is to certify that the project report entitled **“ALGORITHM FOR MATHEMATICAL SIMPLIFICATION”** being submitted **Satyam sharma(160717733061), L.Sai Prakash(160717733062) and K.Dhanush Reddy(160717733106)**  submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering, during the academic year 2019, is a bonafide record of work carried out by them.

**Dr.R.Ch.A.Naidu**

Professor, CSE.

Date:

## Methodist College of Engineering and Technology

**King Koti, Abids, Hyderabad-500001.**

**Department of Computer Science and Engineering**



### **CERTIFICATE BY THE HEAD OF THE DEPARTMENT**

This is to certify that this project report entitled “**ALGORITHM FOR MATHEMATICAL SIMPLIFICATION**” by **Satyam sharma (160717733061), L.Sai Prakash(160717733062) and K.Dhanush Reddy(160717733106) ,**submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering of the Osmania University, Hyderabad, during the academic year 2019,is a bonafide record of work carried out by them.

**Mrs.P.Lavanya**

Head of the Department,CSE

Date:

### **ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our project guide **Dr.R.Ch.A.Naidu, Professor,** for giving us the opportunity to work on this topic. It would never be possible for us to take this project to this level without his innovative ideas, relentless support and encouragement.

We would like to thank our project coordinators **Dr. B. Jayalakshmi, Associate Professor, CSE and Mr Uday kumar, Assistant Professor, CSE.** They helped us by being an example of high vision and pushing towards greater limits of achievement.

Our sincere thanks to **Mrs.P.Lavanya, Head of the Department of Computer Science and Engineering,** for her valuable guidance and encouragement which has played a major role in the completion of the project and for helping us by being an example of high vision and pushing towards greater limits of achievement.

We would like to express a deep sense of gratitude towards **Dr. G. Ravinder Reddy, Principal, Methodist College of Engineering and Technology,** for always being an inspiration and for encouraging us in every possible way.

We would like to express a deep sense of gratitude towards **Dr.Lakshmipathi Rao, Director, Methodist College of Engineering and Technology,** for always being an inspiration and for encouraging us in every possible way

We are indebted to the Department of Computer Science & Engineering and Methodist College of Engineering and Technology for providing us with all the required facility to carry our work in a congenial environment. We extend our gratitude to the CSE Department staff for providing us the needful time whenever requested.

We would like to thank our parents for allowing us to realize our potential, all the support they have provided us over the years was the greatest gift anyone has ever given us and also for teaching us the value of hard work and education. Our parents has offered us with tremendous support and encouragement, thanks to our parents for all the moral support and the amazing opportunities they have given us over the years.

**Abstract**

* Solving complex mathematical equations is both time consuming and mostly inaccurate when done manually.
* The main aim of this project is to develop an algorithm which makes Mathematical Simplification of complex equations.
* It provides user the flexibility to choose an method.
* It provides the user the flexibility to give an user the values of his choice.
* It provides very speed computation for any complex equations with user requested values.
* It help’s an user to solve n number of equations in m number of methods at same time .

**Contents**

**List of Figures .......................................................................................i**

**1.Introduction.........................................................................................1**

1.1 Existing System.................................................................................1

1.2 Proposed System................................................................................1

1.3 Advantages.........................................................................................2

1.4 Disadvantages.....................................................................................2

**2.System Analysis………......................................................................3**

2.1 Hardware requirements......................................................................3

2.2 Software requirements.......................................................................3

**3.Design and Implementation..............................................................4**

3.1 Block Diagram...................................................................................4

3.2 Algorithm….......................................................................................5

3.3 Flow chart..........................................................................................6

3.4 Code...................................................................................................7

**4.Result…...............................................................................................15**

4.1 Screenshots.........................................................................................15

**5. Conclusion and Future Scope ..........................................................19**

**References................................................................................................20**

**LIST OF FIGURES**

Fig 1.1 Block diagram

Fig 1.2.Flowchart

**1 . INTRODUCTION**

“Algorithm For Mathematical Simplication” is a algorithm designed for helping the user to solve complex mathematical equations without doing them manually .The algorithm mainly build with the vision of providing an easy mathematical simplication approach with peak accuracy. The algorithm also provides the user the flexibility to apply multiple methods at the same time , thus giving an user the smart way to choose the best method for the given equation

This program mainly consists of Three Mathematical methods which include

1.Newton Raphson Method

2.Bisction method

3.Newton Interpolation method

\_1\_

**1.1 EXISTING SYSTEM**

In existing system, solving mathematical equations are done manually.

Solving the mathematical equations are complex and time consuming inaccurate.

Selecting an appropriate method for an mathematical simplification of a particular equation consumes a lot of time and energy of an user with no guaranteed accurate outcome resulting in false predictions

* 1. **PROPOSED SYSTEM**
* User friendly interface
* Less error
* Peak accuracy
* Fast access to methods
* Quick simplification
* More storage capacity

\_2\_

**1.3 ADVANTAGES**

* Eco-Friendly: Paperwork can be avoided
* Time efficient and user friendly
* Easy access to different Mathematical methods like Newton raphson, Bisection method, Newton Interpolation method .
  1. **DISADVANTAGES**
* The functions which are called inside the respective method are needed to be defined in it.
* It consists only a finite number of mathematical methods

\_3\_

## 2.SYSTEMREQUIREMENTS

**2.1 SOFTWARE REQUIREMENTS:**

(1) OS: Windows 10 / LINUX / MacOS

(2) Language: C++

**2.2HARDWARE REQUIREMENTS:**

(1) Intel core i5,i3

(2) RAM 4gb

(3) Hard disk 1Tb

\_4\_

**3.DESIGN AND IMPLEMENTATION**

**3.1 Block diagram**

Fig 1.1 block diagram

\_5\_

**3.2 ALGORITHM**

* **Step 1:-**Start
* **Step 2:-** Displays the methods

(I)Newton Raphson Method

(II)Bisection Method

(III)Newton Interpolation

* **Step 3:-** The user has to choose an option.
* **Step 4:-** If case (I) is selected then Newton Raphson method is performed and go to step 5.

If case (II) is selected then Bisection method is performed and go to step 5.

If case (III) is selected then Newton Interpolation method is performed and

go to step 5.

* **Step 5:-** Displays the message ‘ Do you want to continue?’

If Yes, go to **Step 2**

If No, go to **Step 6**

* **Step 6:-** Stop

\_6\_

**3.3 FLOWCHART**

Fig 1.2

\_7\_

**3.4 Code**

//step1: including necessary headers and necessary libraries

#include<iostream>

#include<cmath>

#include<iomanip>

#include<stdio.h>

using namespace std;

//step2:here we are only going to declare the functions , we are going to use

//newton functions declaration

double nf(double x);

double fprime(double x);

double nrm();

//bisection functions declaration

double f(double x);

//interpolation declaration

double Li(int i,int n,double x[],double X);

double Pn(int n, double x[], double y[], double X);

//functions where the logics are written for above mentioned methods

double nrm();

double bsm();

double ipm();

//step3:we are defining our function now

double nf(double x) //define the function here, ie give the equation

{

\_8\_

double a=pow(x,3.0)-x-11.0; //write the equation whose roots are to be determined

return a;

}

double fprime(double x)

{

double b=3\*pow(x,2.0)-1.0; //write the first derivative of the equation

return b;

}

double f(double x) //define the function here, ie give the equation

{

double a=x\*x-4.0; //write the equation whose root is to be determined

return a;

}

/\*Function to evaluate Li(x)\*/

double Li(int i,int n,double x[],double X){

int j;

double prod=1;

for(j=0;j<=n;j++){

if(j!=i)

prod=prod\*(X-x[j])/(x[i]-x[j]);

}

return prod;

}

/\*Function to evaluate Pn(x) where Pn is the Lagrange interpolating polynomial of degree n\*/

double Pn(int n, double x[], double y[], double X){

double sum=0;

\_9\_

int i;

for(i=0;i<=n;i++){

sum=sum+Li(i,n,x,X)\*y[i];

}

return sum;

}

//nrm method for logic

double nrm(){

double x,x1,e,nfx,fx1;

cout.precision(4); //set the precision

cout.setf(ios::fixed);

cout<<"Enter the initial guess\n"; //take an intial guess

cin>>x1;

cout<<"Enter desired accuracy\n"; //take the desired accuracy

cin>>e;

nfx=nf(x);

fx1=fprime(x);

cout <<"x{i}"<<" "<<"x{i+1}"<<" "<<"|x{i+1}-x{i}|"<<endl;

do

{

x=x1; /\*make x equal to the last calculated value of x1\*/

nfx=nf(x); //simplifying f(x)to fx

fx1=fprime(x); //simplifying fprime(x) to fx1

x1=x-(nfx/fx1); /\*calculate x{1} from x, fx and fx1\*/

cout<<x<<" "<<x1<<" "<<abs(x1-x)<<endl;

} while (fabs(x1-x)>=e);

/\*if |x{i+1}-x{i}| remains greater than the desired accuracy, continue the loop\*/

cout<<"The root of the equation is "<<x1<<endl;

return 0;

}

\_10\_

//bsm method for logic

double bsm(){

cout.precision(5); //set the precision

cout.setf(ios::fixed);

double a,b,c,e,fa,fb,fc; //declare some needed variables

a:cout<<"Enter the initial guesses:\na="; //Enter the value of a(set a label('a:') for later use with goto)

cin>>a;

cout<<"\nb="; //Enter the value of b

cin>>b;

cout<<"\nEnter the degree of accuracy desired"<<endl; //Enter the accuracy cin>>e; //e stands for accuracy

int iter=0;

if (f(a)\*f(b)>0) //Check if a root exists between a and b

{ //If f(a)\*f(b)>0 then the root does not exist between a and b

cout<<"Please enter a different intial guess"<<endl;

goto a; //go back to 'a' ie 17 and ask for different values of a and b

}

else //else a root exists between a and b

{

cout<<"Iter"<<setw(14)<<"a"<<setw(18)<<"b"<<setw(18)<<"c"<<setw(18)<<"f(c)"<<setw(18)<<"|a-b|"<<endl;

cout<<"--------------------------------------------------------------------------------------------\n";

while (fabs(a-b)>=e)

/\*if the mod of a-b is greater than the accuracy desired keep bisecting the interval\*/

{

c=(a+b)/2.0; //bisect the interval and find the value of c

fa=f(a);

fb=f(b);

fc=f(c);

iter++;

cout<<iter<<setw(18)<<a<<setw(18)<<b<<setw(18)<<c<<setw(18)<<fc<<setw(18)<<fabs(a-b)<<endl;/\*print the values of a,b,c and fc after each iteration\*/

if (fc==0) //if f(c)=0, that means we have found the root of the equation

{

\_11\_

cout<<"The root of the equation is "<<c<<endl;; /\*print the root of the equation and end program\*/

return 0;

}

if (fa\*fc>0) //if f(a)xf(c)>0, that means no root exist between a and c

{

a=c; /\*hence make a=c, ie make c the starting point of the interval and b the end point\*/

}

else if (fa\*fc<0)

{

b=c; /\*this means that a root exist between a and c therfore make c the end point of the interval\*/

}

}

}

//The loop ends when the difference between a and b becomes less than the desired accuracy ie now the value stored in 'c' can be called the approximate root of the equation

cout<<"The root of the equation is "<<c<<endl;; //print the root

return 0;

}

//ipm method for logic

double ipm(){

int i,n; //n is the degree

cout << "Enter the number of data-points:\n";

cin >> n; //no. of data-points is n+1

n = n-1;

//Arrays to store the (n+1) x and y data-points of size n+1

double x[n+1];

double y[n+1];

cout<<"Enter the x data-points:\n";

for(i=0;i<n+1;i++){

\_12\_

cin>>x[i];

}

cout<<"Enter the y data-points:\n";

for(i=0;i<n+1;i++){

cin>>y[i];

}

double X; //value of x for which interpolated value is required

cout<<"Enter the value of x for which you want the interpolated value of y(x):\n";

cin>>X;

cout<<"The interpolated value is : " << Pn(n,x,y,X);

}

//step4:here we are going to write our main functions

//now we will write up our main function

int main(){

cout <<"........................................................................"<<endl;

cout <<" select a method"<< endl;

cout <<".............................................................................."<<endl;

cout << "1.Newton-Raphson Method \t 2.bisection method \t 3.LAGRANGE INTERPOLATION \t"<< endl;

cout <<"................................................................................"<<endl;

char ch;

int choice;

do

{

cout <<"enter you choice of method \n" ;

cin >> choice ;

switch(choice)

{

case 1:nrm();

break;

case 2:bsm();

break;

case 3:ipm();

break;

\_13\_

}

cout <<"................................................................................"<<endl;

cout << "enter y to continue" << endl;

cin >> ch;

cout <<"................................................................................"<<endl;

}while(ch=='Y'||ch=='y');

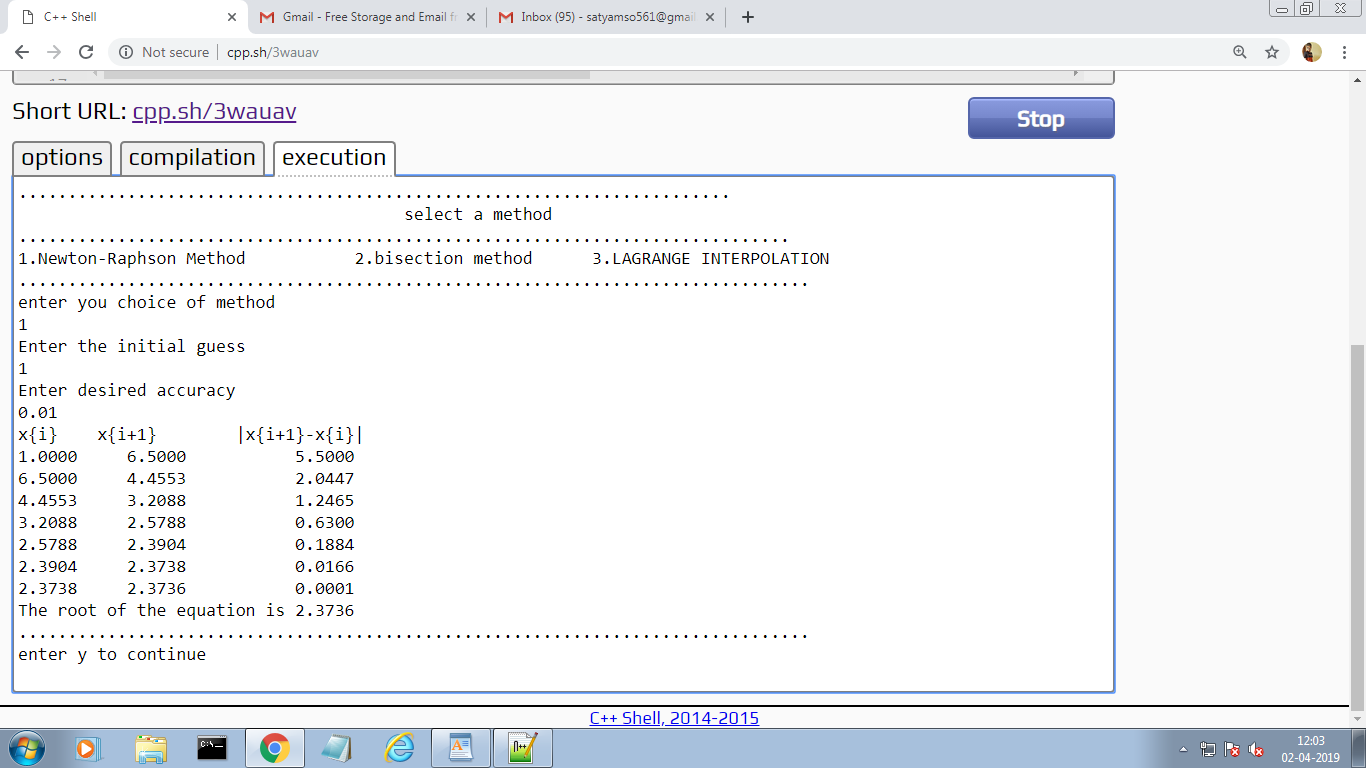
}

\_14\_

**RESULT**

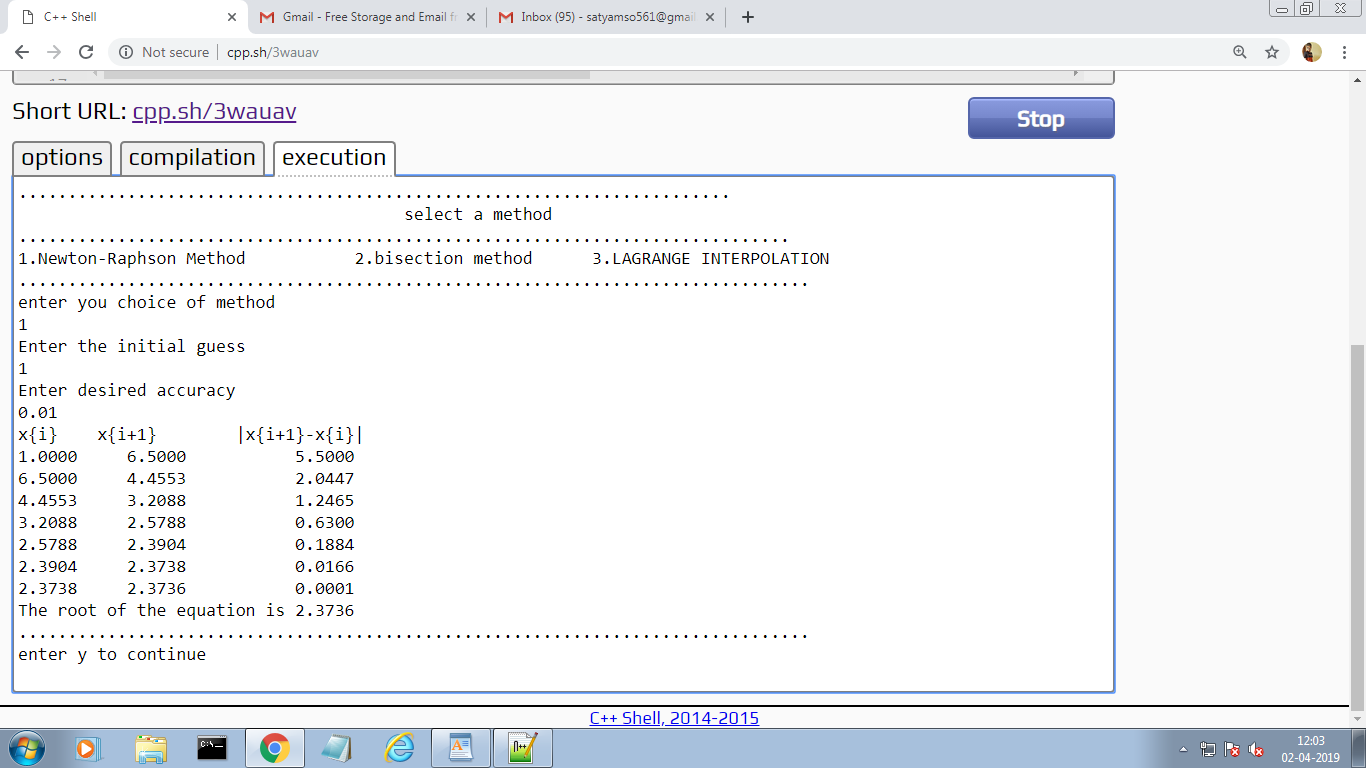
**4.1 SCREENSHOTS**

**Fig 4.1.1.Main Menu**

****

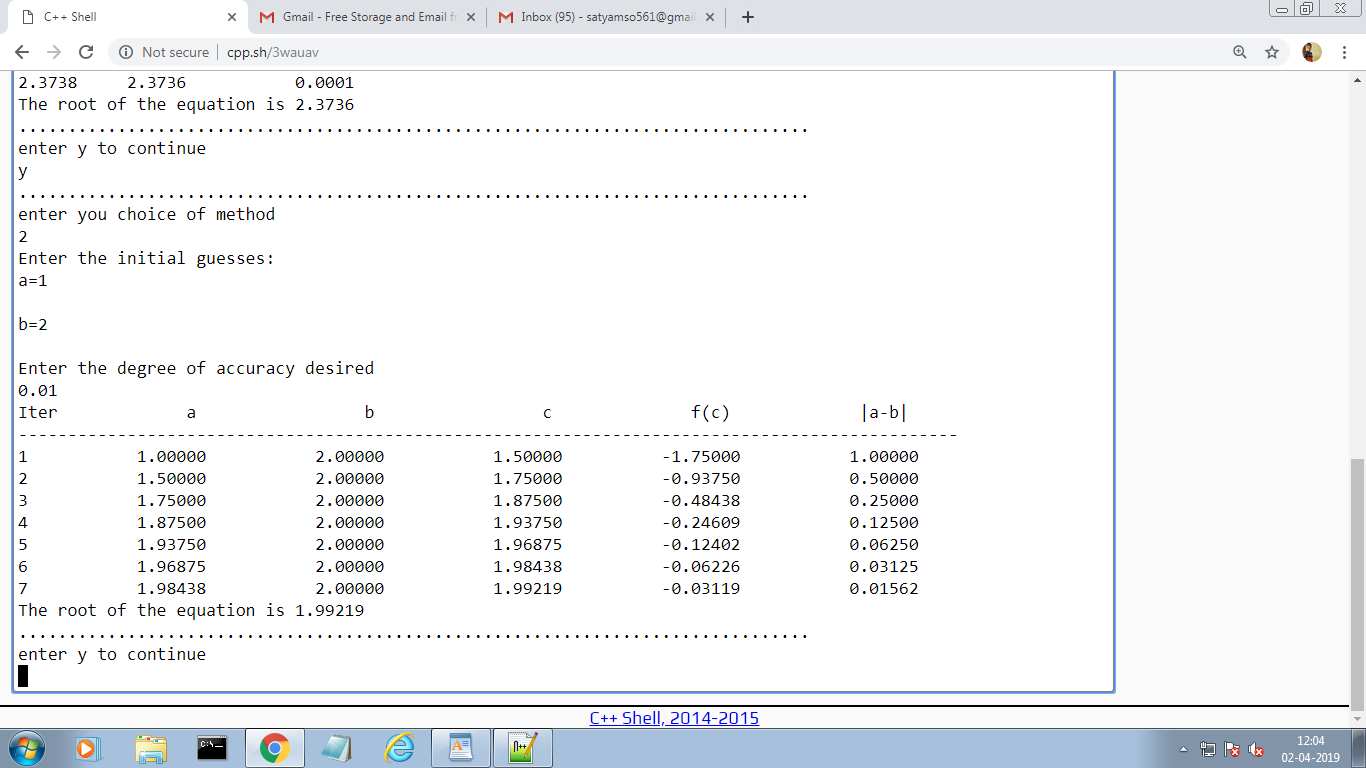
\_15\_

**Fig 4.1.2.Newton Raphson Method**

****

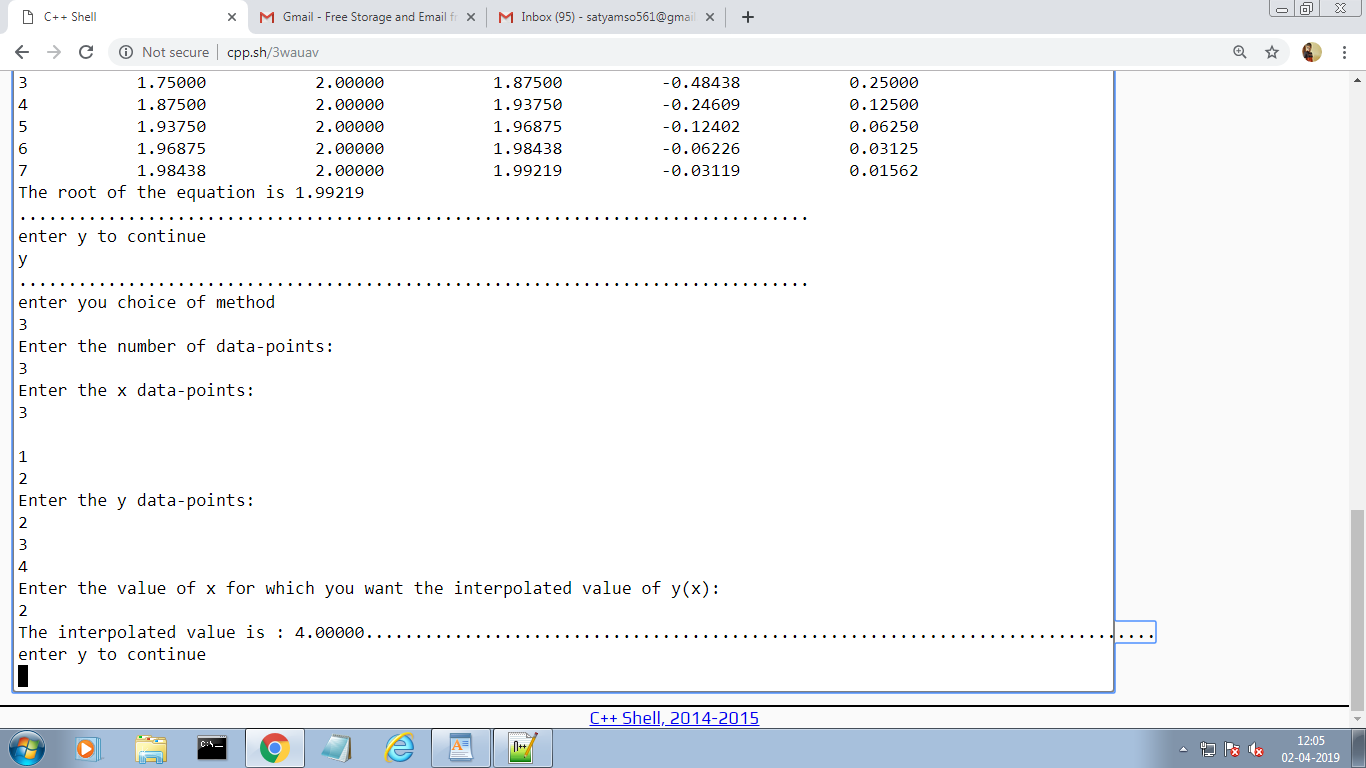
\_16\_

**Fig 4.1.3 Bisection method**

****

\_17\_

**Fig4.1.4 Newton Interpolation Method**

****

\_18\_

**CONCLUSION AND FUTURE SCOPE**

* Our ALGORITHM FOR MATHEMATICAL SIMPLIFICATION is mainly build with the vision of providing an easy mathematical simplification approach with peak accuracy to the user. It mainly used to solve complex problems with a modern approach .
* This is mainly used for students for solving problems of the methods mentioned in it.

Our project will be able to implement in future after making some changes and modifications. The modifications that can done in our project are:

* Adding more Number of Mathematical Methods to the code.
* Adding a option to take functions from the user in future can also be done.
* Efficiency and accuracy can be improved

\_19\_

## REFERENCES

* [Introductory Methods of Numerical Analysis by S.S. Sastry](http://www.flipkart.com/introductory-methods-numerical-analysis-english-5th/p/itmdwu92fzutjczw?pid=9788120345928&affid=sandeepgfg)
* [https://en.wikipedia.org/wiki/Newton’s\_method](https://en.wikipedia.org/wiki/Newton's_method)
* <https://en.wikipedia.org/wiki/bisection_method>
* [https://en.wikipedia.org/wiki/Newton’s\_interpolation\_method](https://en.wikipedia.org/wiki/Newton's_interpolation_method)

\_20\_