1. Write a program to implement bisection method for solving non-linear equations.

Bisection Method is one of the simplest, reliable, easy to implement and convergence guarenteed method for finding real root of non-linear equations.

Bisection method is bracketing method and starts with two initial guesses say x0 and x1 such that x0 and x1 brackets the root i.e. f(x0)f(x1)<0

Bisection method is based on the fact that if f(x) is real and continuous function, and for two initial guesses x0 and x1 brackets the root such that: f(x0)f(x1) <0 then there exists atleast one root between x0 and x1.

Root is obtained in Bisection method by successive halving the interval i.e. If x0 and x1 are two guesses then we compute new approximated root as:

x2 = (x0 + x1)/2

Now we have following three different cases:

If f(x2)=0 then the root is x2.

If f(x0)f(x2)<0 then root lies between x0 and x2.

If f(x0)f(x2)> 0 then root lies between x1 and x2.

And then process is repeated until we find the root within desired accuracy.

**Algorithm**

Step 1 : Start

Step 2 : Define function f(x)

Step 3 : Choose initial guesses x0 and x1 such that f(x0)f(x1) < 0

Step 4 : Choose pre-specified tolerable error e.

Step 5 : Calculate new approximated root as x2 = (x0 + x1)/2

Step 6 : Calculate f(x0)f(x2)

a. if f(x0)f(x2) < 0 then x0 = x0 and x1 = x2

b. if f(x0)f(x2) > 0 then x0 = x2 and x1 = x1

c. if f(x0)f(x2) = 0 then goto (8)

Step 7 : if |f(x2)| > e then goto (5) otherwise goto (8)

Step 8 : Display x2 as root.

Step 9 : Stop

**Implementation Program Code**

#include<stdio.h>

#include<conio.h>

#include<math.h>

#define f(x) pow(x,2) - 4 \* x - 10

void main(){

float x0, x1, x2, f0, f1, f2, e;

int step = 1;

up:

printf("\nEnter two initial guesses:\n");

scanf("%f%f", &x0, &x1);

printf("Enter tolerable error:\n");

scanf("%f", &e);

f0 = f(x0);

f1 = f(x1);

if(f0 \* f1 > 0.0){

printf("Incorrect Initial Guesses.\n");

goto up;

}

printf("\nStep\t\tx0\t\tx1\t\tx2\t\tf(x2)\n");

do{

x2 = (x0 + x1)/2;

f2 = f(x2);

printf("%d\t\t%f\t%f\t%f\t%f\n",step, x0, x1, x2, f2);

if(f0 \* f2 < 0){

x1 = x2;

f1 = f2;

}

else{

x0 = x2;

f0 = f2;

}

step = step + 1;

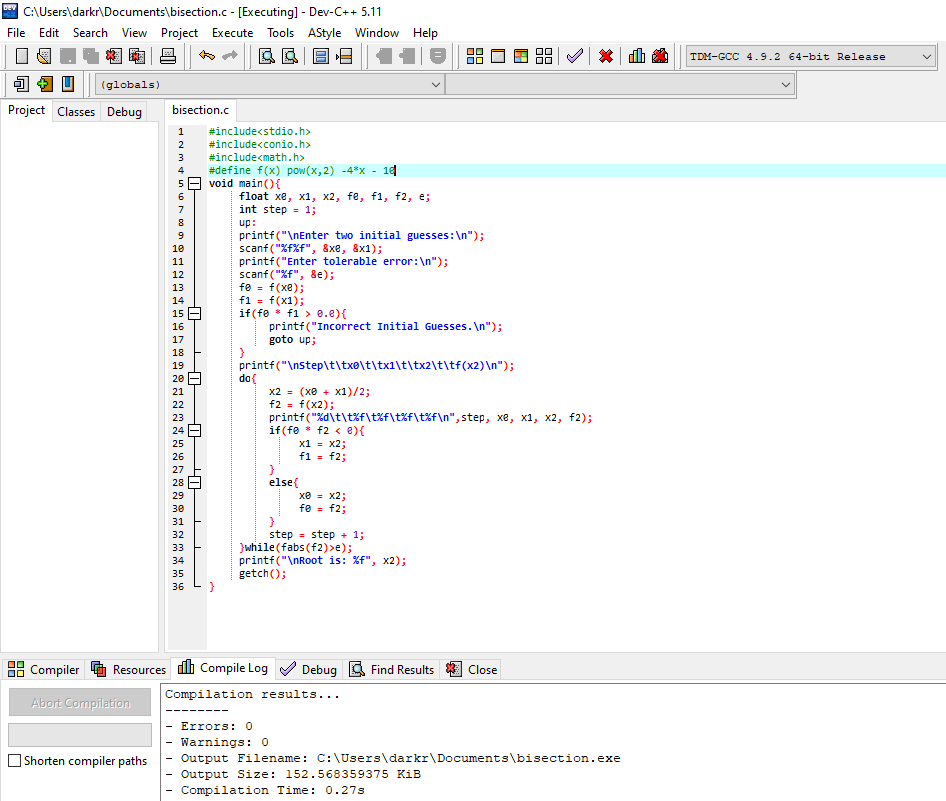
}while(fabs(f2)>e);

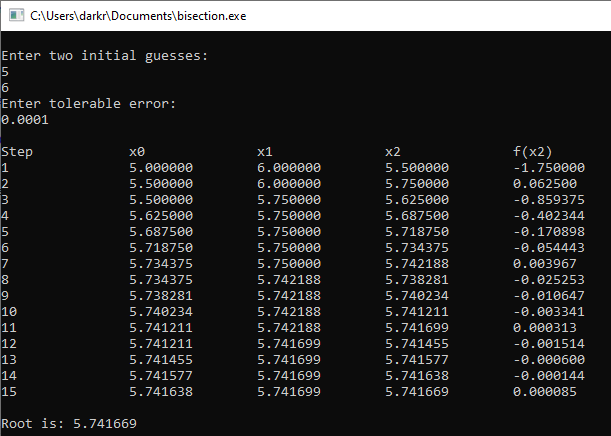
printf("\nRoot is: %f", x2);

getch();

}

**Program Code Screenshot**



**Output :**

1. Write a program to implement Newton Raphson method for solving non-linear equations.

Newton Raphson Method is an open method and starts with one initial guess for finding real root of non-linear equations.

In Newton Raphson method if x0 is initial guess then next approximated root x1 is obtained by following formula:

x1 = x0 - f(x0) / g(x0)

And an algorithm for Newton Raphson method involves repetition of above process i.e. we use x1 to find x2 and so on until we find the root within desired accuracy.

**Algorithm**

Step 1 : Start

Step 2 : Define function as f(x)

Step 3 : Define first derivative of f(x) as g(x)

Step 4 : Input initial guess (x0), tolerable error (e) and maximum iteration (N)

Step 5 : Initialize iteration counter i = 1

Step 6 : If g(x0) = 0 then print "Mathematical Error" and goto (12) otherwise goto (7)

Step 7 : Calculate x1 = x0 - f(x0) / g(x0)

Step 8 : Increment iteration counter i = i + 1

Step 9 : If i >= N then print "Not Convergent" and goto (12) otherwise goto (10)

Step 10 : If |f(x1)| > e then set x0 = x1 and goto (6) otherwise goto (11)

Step 11 : Print root as x1

Step 12 : Stop

**Implementation Program Code**

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<stdlib.h>

#define f(x) 3\*x - cos(x) -1

#define g(x) 3 + sin(x)

void main(){

float x0, x1, f0, f1, g0, e;

int step = 1, N;

printf("\nEnter initial guess:\n");

scanf("%f", &x0);

printf("Enter tolerable error:\n");

scanf("%f", &e);

printf("Enter maximum iteration:\n");

scanf("%d", &N);

printf("\nStep\t\tx0\t\tf(x0)\t\tx1\t\tf(x1)\n");

do{

g0 = g(x0);

f0 = f(x0);

if(g0 == 0.0) {

printf("Mathematical Error.");

exit(0);

}

x1 = x0 - f0/g0;

printf("%d\t\t%f\t%f\t%f\t%f\n",step,x0,f0,x1,f1);

x0 = x1;

step = step+1;

if(step > N) {

printf("Not Convergent.");

exit(0);

}

f1 = f(x1);

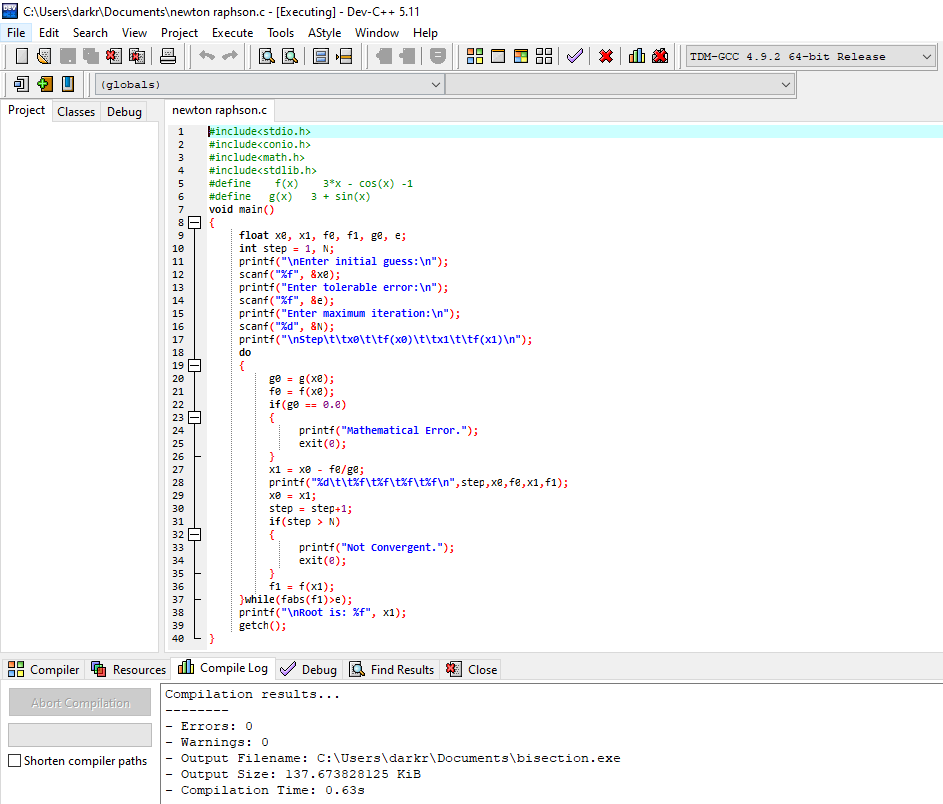
}while(fabs(f1)>e);

printf("\nRoot is: %f", x1);

getch();

}

**Program code Screenshot**



**Output :**



1. Write a program to implement Fixed point iteration method for solving non-linear equations.

Fixed point iteration method is open and simple method for finding real root of non-linear equation by successive approximation. It requires only one initial guess to start. Since it is open method its convergence is not guaranteed. This method is also known as Iterative Method

To find the root of nonlinear equation f(x)=0 by fixed point iteration method, we write given equation f(x)=0 in the form of x = g(x).

If x0 is initial guess then next approximated root in this method is obtaine by:

x1 = g(x1)

And similarly, next to next approximated root is obtained by using value of x1 i.e.

x2 = g(x2)

And the process is repeated until we get root within desired accuracy.

Note: While expressing f(x)=0 to x = g(x) we can have many different forms. For convergence, following criteraia must be satisfied.

g'(x)|< 1

**Algorithm**

Step 1 : Start

Step 2 : Define function f(x)

Step 3 : Define function g(x) which is obtained from f(x)=0 such that x = g(x) and |g'(x) < 1|

Step 4 : Choose initial guess x0, Tolerable Error e and Maximum Iteration N

Step 5 : Initialize iteration counter: step = 1

Step 6 : Calculate x1 = g(x0)

Step 7 : Increment iteration counter: step = step + 1

Step 8 : If step > N then print "Not Convergent" and goto (12) otherwise goto (10)

Step 9 : Set x0 = x1 for next iteration

Step 10 : If |f(x1)| > e then goto step (6) otherwise goto step (11)

Step 11 : Display x1 as root.

Step 12 : Stop

**Implementation Program Code**

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include <stdlib.h>

#define f(x) cos(x)-3\*x+1

#define g(x) (1+cos(x))/3

void main(){

int step=1, N;

float x0, x1, e;

printf("Enter initial guess: ");

scanf("%f", &x0);

printf("Enter tolerable error: ");

scanf("%f", &e);

printf("Enter maximum iteration: ");

scanf("%d", &N);

printf("\nStep\tx0\t\tf(x0)\t\tx1\t\tf(x1)\n");

do{

x1 = g(x0);

printf("%d\t%f\t%f\t%f\t%f\n",step, x0, f(x0), x1, f(x1));

step = step + 1;

if(step>N){

printf("Not Convergent.");

exit(0);

}

x0 = x1;

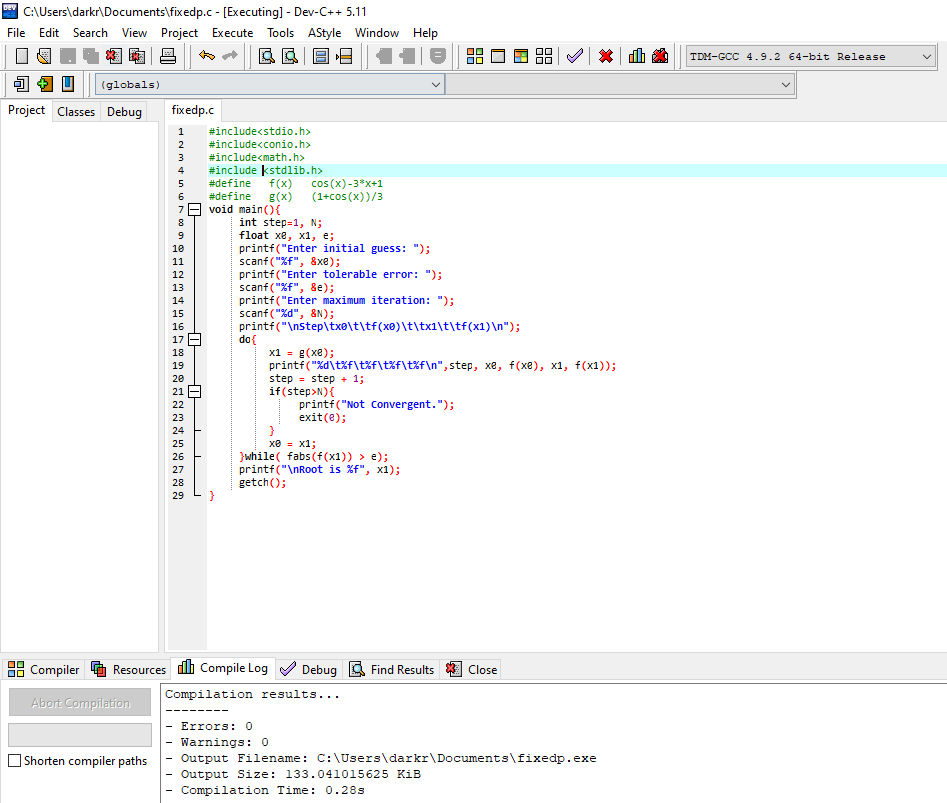
}while( fabs(f(x1)) > e);

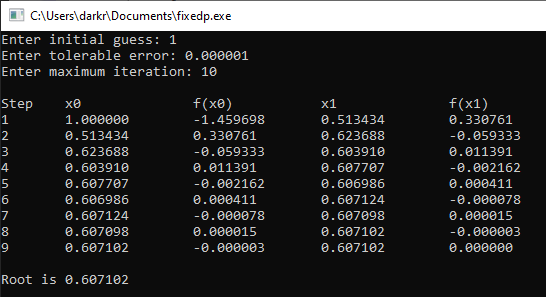
printf("\nRoot is %f", x1);

getch();

}

**Program code Screenshot**



**Output :**