# Experiment No.: 01

# Name of the Experiment:

Determining the root of a non-linear equation using Bisection Method.

# Objectives:

* Getting introduced with Bisection Method.
* Determining the roots of non-linear equations in C.
* Determining the roots of non-linear equations in Microsoft Excel.
* Making comparison of experimental results in C and in Microsoft Excel.

# Theory:

The bisection method is one of the simplest and most reliable of iterative methods for the solution of nonlinear equations. This method is also known as binary chopping or half interval method. It relies on the fact that if *f*(*x*) is real and continuous in the interval *a* < *x* < *b*, and *f*(*a*) and *f*(*b*) are of opposite signs, that is,

*f*(*a*) . *f*(*b*) < 0

Then there is at least one real root in the interval between *a* and *b*. That is,

x0=(x1+x2)/2

Now there exist following three conditions:

1. If f(x0) =0, we have a root at x0.

2. If f (x0) f(x1) < 0, there is a root between x0 and x1

3. If f (x0) f(x2) < 0, there is a root between x0 and x2

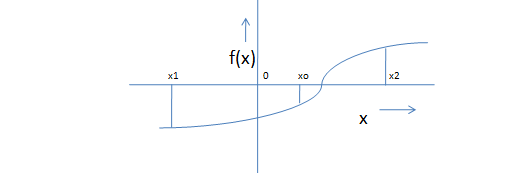


Figure: Illustration of Bisection Method

# Algorithm for Bisection Method:

1. Decide initial values for x1 and x2 and stopping criterion, E.

2. Computing f1=f(x1) and f2=f(x2)

3. If f1\* f2>0, x1 and x2 do not bracket any root and go to step 7.

4. Compute x0=(x1+x2)/2 and compute f0=f(x0)

5. If f1\*f0 <0 then

set x2=x0

else

set x1=x0

set f1=f0

6. If absolute value of (x2 – x1)/x2 is less than error E, then

root= (x1 +x2)/2

write the value of root,

go to step 7

else

go to step 4

7. Stop.

-------------------------------------------------------------------------------------------------------------

# C code of Bisection Method:

/\* Write a C program to find out a real root of the following non-linear equation using Bisection method:

x2 - 4x - 10 = 0

Done by: XXXXX, Class Roll: XXXXX, Exam Roll: XXXXX

Date:

\*/

/\*

#include<stdio.h>

#include<math.h>

#define ERROR 0.000001

double F(double x)

{

double y;

y=(x)\*(x)-4\*(x)-10;

return(y);

}

main()

{

int s, count;

double a, b, root;

printf("\n");

printf("SOLUTION BY BISECTION METHOD \n");

printf("\n");

printf("Input starting values: ");

scanf("%lf%lf",&a,&b);

/\*calling the subroutine bim() \*/

bim(&a, &b, &root, &s, &count);

if(s == 0)

{

printf("\n");

printf("Starting points do not bracket any root \n");

printf("Check whether they bracket EVEN roots");

printf("\n");

}

else

{

printf("\nRoot = %lf \n", root);

printf("F(root) = %lf\n", F(root));

printf("\n");

printf("Iterations = %d\n", count);

printf("\n");

}

}

/\*End of main program \*/

/\* ------------------------------------------------------------------------------------------------------ \*/

/\* Defining the subroutine bim() \*/

bim(double \*a, double \*b, double \* root, int \*s, int \*count)

{

double x1, x2, x0, f0, f1, f2;

x1 = \*a;

x2 = \*b;

f1 = F(x1);

f2 = F(x2);

/\*Test if initial values bracket a SINGLE root \*/

if(f1 \* f2 > 0)

{

\*s = 0;

return ; /\*Program terminated\*/

}

else

{

\*count = 0;

begin:

x0 = (x1 + x2)/2.0;

f0 = F(x0);

if(f0 == 0)

{

\*s = 1;

\*root = x0;

return ;

}

if(f1 \* f0 < 0)

{

x2 = x0;

f2 = f0;

}

else

{

x1 = x0;

f1 = f0;

}

/\*Testfor accuracy and repeat the process,if necessary \*/

if(fabs(x2 - x1) < ERROR)

{

\*s = 1;

\*root = (x1 + x2) / 2.0;

return ; /\*Iteration ends \*/

}

else

{

\*count = \*count + 1;

goto begin;

}

}

}

/\*End of subroutine bim ()\*/

**Output:**

SOLUTION BY BISECTION METHOD

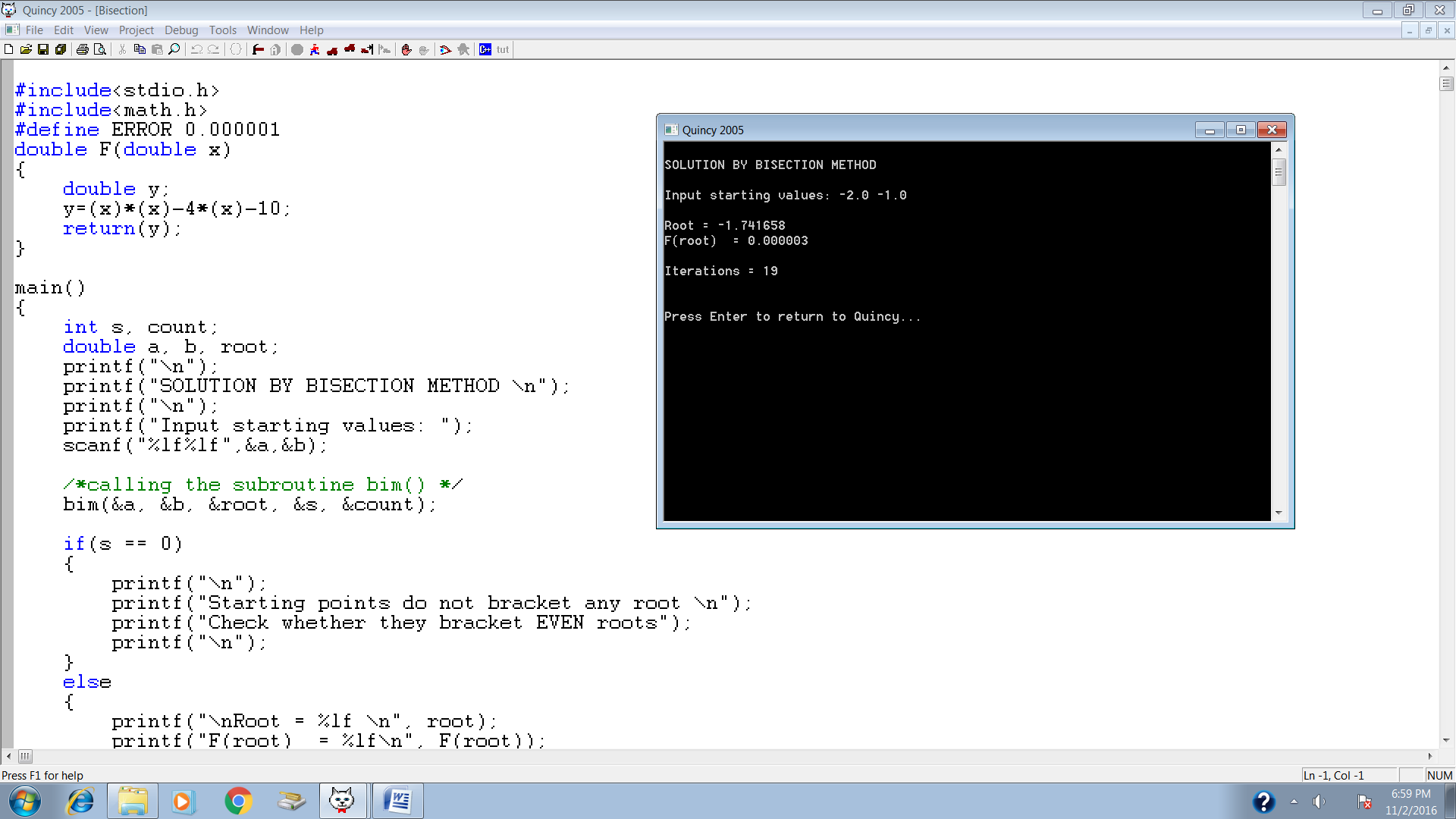
Input starting values: -2.0 -1.0

Root = -1.741658

F(root) = 0.000003

Iterations = 19

Press Enter to return to Quincy...



# Bisection Method in Microsoft Excel:

Experiment Name: Find the root of the following equation using Bisection Method:



= 6

Plot the function:

|  |  |
| --- | --- |
| **x** | **f(x)** |
| -6 | 50 |
| -5 | 35 |
| -4 | 22 |
| -3 | 11 |
| -2 | 2 |
| -1 | -5 |
| 0 | -10 |
| 1 | -13 |
| 2 | -14 |
| 3 | -13 |
| 4 | -10 |
| 5 | -5 |
| 6 | 2 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **x1** | **x2** | **x0** | **f(x1)** | **f(x2)** | **f(x0)** | **f(x1)f(x0)** | **f(x2)f(x0)** |
| -2 | -1 | -1.5 | 2 | -5 | -1.75 | -3.5 | 8.75 |
| -2 | -1.5 | -1.75 | 2 | -1.75 | 0.0625 | 0.125 | -0.109375 |
| -1.75 | -1.5 | -1.625 | 0.0625 | -1.75 | -0.859375 | -0.0537109 | 1.50390625 |
| -1.75 | -1.625 | -1.6875 | 0.0625 | -0.859375 | -0.4023438 | -0.0251465 | 0.34576416 |
| -1.75 | -1.6875 | -1.71875 | 0.0625 | -0.402344 | -0.1708984 | -0.0106812 | 0.06875992 |
| -1.75 | -1.71875 | -1.734375 | 0.0625 | -0.170898 | -0.0544434 | -0.0034027 | 0.00930429 |
| -1.75 | -1.734375 | -1.742188 | 0.0625 | -0.054443 | 0.0039673 | 0.00024796 | -0.000216 |
| -1.742188 | -1.734375 | -1.738281 | 0.0039673 | -0.054443 | -0.0252533 | -0.0001002 | 0.00137487 |
| -1.742188 | -1.7382813 | -1.740234 | 0.0039673 | -0.025253 | -0.0106468 | -4.224E-05 | 0.00026887 |
| -1.742188 | -1.7402344 | -1.741211 | 0.0039673 | -0.010647 | -0.0033407 | -1.325E-05 | 3.5568E-05 |
| -1.742188 | -1.7412109 | -1.741699 | 0.0039673 | -0.003341 | 0.000313 | 1.2419E-06 | -1.046E-06 |
| -1.741699 | -1.7412109 | -1.741455 | 0.000313 | -0.003341 | -0.0015139 | -4.739E-07 | 5.0575E-06 |
| -1.741699 | -1.7414551 | -1.741577 | 0.000313 | -0.001514 | -0.0006004 | -1.88E-07 | 9.0901E-07 |
| -1.741699 | -1.7415771 | -1.741638 | 0.000313 | -0.0006 | -0.0001437 | -4.499E-08 | 8.6285E-08 |
| -1.741699 | -1.7416382 | -1.741669 | 0.000313 | -0.000144 | 8.467E-05 | 2.6505E-08 | -1.217E-08 |
| -1.741669 | -1.7416382 | -1.741653 | 8.467E-05 | -0.000144 | -2.952E-05 | -2.499E-09 | 4.2417E-09 |

# Result:

After 1st iteration the root is -1.5

After 2nd iteration the root is -1.75

After 3rd iteration the root is -1.625

After 5th iteration the root is -1.71875

After 10th iteration the root is -1.74121

After 15th iteration the root is -1.74167

Approximately the root is -1.74166

# Discussion:

The root is not totally accurate. The root has been taken when the interval between x1 and x2 is equal to 1.91E-06. After 20th iteration the difference is 1.91E-06. This is the error of this calculation. The amount of error is too little that it can be avoided. So, -1.74166 can be considered as the root of the equation