

# CBNST LAB PRACTICAL

Date: September 14, 2021

Name : Deepankar Sharma  
Course: BCA  
Student Id : 20041299  
University Roll No: 2092014

## Bisection Method

### Algorithm:

1. Start
2. Define function  $f(x)$
3. Input
  - a. Lower and Upper guesses  $x_0$  and  $x_1$
  - b. tolerable error  $e$
4. If  $f(x_0)*f(x_1) > 0$   
    print "Incorrect initial guesses"  
    goto 3  
End If
5. Do  
     $x_2 = (x_0+x_1)/2$   
  
    If  $f(x_0)*f(x_2) < 0$   
         $x_1 = x_2$   
    Else  
         $x_0 = x_2$   
    End If  
  
    while  $\text{abs}(f(x_2)) > e$
6. Print root as  $x_2$
7. Stop

### Code:

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#define phi(x) (pow(2.718282, -1 * x) - sin(x))
```

# CBNST LAB PRACTICAL

Date: September 14, 2021

```
double differential(double x0)
{
    const double delta = 1.0e-10;
    double x1 = x0 - delta;
    double x2 = x0 + delta;

    double y1 = phi(x1);
    double y2 = phi(x2);

    // printf("gradient= %f\n", grad);
    return (y2 - y1) / (x2 - x1);
    // return (pow(-2.718282, -1*x)-cos(x));
}

int main()
{
    int k = 0;
    double x1, x0, f0, f1, x2, f2;
    int step = 1, N;
    double allErr;

    printf("Enter the allowed Error: ");
    scanf("%lf", &allErr);
    printf("Enter the interval lower limit (initial guess 'a'): ");
    scanf("%lf", &x0);
    printf("Enter the interval upper limit (initial guess 'b'): ");
    scanf("%lf", &x1);

    f0 = phi(x0);
    f1 = phi(x1);
    if (f0 * f1 > 0.0)
    {
        printf("\n\nIncorrect Initial Guesses !!!!!\n");
    }

    printf("Enter maximum iteration: ");
    scanf("%d", &N);
    {
        {
            printf("\nStep\t\tta\t\ttb\t\ttf(a)\t\ttf(b)\t\ttc=(a+b)/2\t\ttf(c)\n\n");
```

# CBNST LAB PRACTICAL

Date: September 14, 2021

```
do
{
    x2=(x0+x1)/2;
    f2= phi(x2);

    printf("%d\t\t%f\t%f\t%f\t%f\t%f\t%f\t%f\n", step, x0,
x1, f0, f1, x2, f2);

    step = step + 1;

    if (step > N)
    {
        printf("%d iterations Completed !!!!!\n", N);
        exit(0);
    }

    if (f0 * f2 < 0)
    {
        x1 = x2;
        f1 = f2;
    }
    else
    {
        x0 = x2;
        f0 = f2;
    }

} while (fabs(f1) > allErr);

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

# CBNST LAB PRACTICAL

Date: September 14, 2021

The screenshot displays the Visual Studio Code interface with a C++ program implementing the Bisection Method. The code is in a file named `01_BisectionMethod.cpp`. The program defines a function `phi` and uses it to calculate the root of a function `f` using the Bisection Method. The terminal output shows the execution of the program, including the input of the allowed error, the interval limits, and the maximum iterations. The output also displays a table of the iterative process, showing the values of `a`, `b`, `f(a)`, `f(b)`, `c=(a+b)/2`, and `f(c)` for each step.

```
01_BisectionMethod.cpp - Unit 01 - Visual Studio Code
01_BisectionMethod.cpp x 02_NewtonRaphsonMethod.cpp
01_BisectionMethod.cpp > main()
10 double x1 = x0 - delta;
11 double x2 = x0 + delta;
12
13 double y1 = phi(x1);
14 double y2 = phi(x2);
15
16 // printf("gradient= %f\n", grad);
17 return (y2 - y1) / (x2 - x1);

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS E:\03 Semester\CBNST\Unit 01> cd "E:\03 Semester\CBNST\Unit 01"; if ($?) { g++ 01_BisectionMethod.cpp -o 01_BisectionMethod }; if ($?) { .\01_BisectionMethod }
Enter the allowed Error: 0.0001
Enter the interval lower limit (initial guess 'a'): 0
Enter the interval upper limit (initial guess 'b'): 1
Enter maximum iteration: 12

Step    a          b          f(a)        f(b)        c=(a+b)/2    f(c)
1       0.000000   1.000000   1.000000   -0.473592   0.500000     0.127195
2       0.500000   1.000000   0.127195   -0.473592   0.750000    -0.289272
3       0.500000   0.750000   0.127195   -0.289272   0.625000    -0.049836
4       0.500000   0.625000   0.127195   -0.049836   0.562500     0.036480
5       0.562500   0.625000   0.036480   -0.049836   0.593750    -0.007221
6       0.562500   0.593750   0.036480   -0.007221   0.578125     0.014495
7       0.578125   0.593750   0.014495   -0.007221   0.585938     0.003683
8       0.585938   0.593750   0.003683   -0.007221   0.589844    -0.001817
9       0.585938   0.589844   0.003683   -0.001817   0.587891     0.000891
10      0.587891   0.589844   0.000891   -0.001817   0.588867    -0.000464
11      0.587891   0.588867   0.000891   -0.000464   0.588379     0.000213
12      0.588379   0.588867   0.000213   -0.000464   0.588623    -0.000125

12 iterations Completed !!!!!
PS E:\03 Semester\CBNST\Unit 01>
```