**PROGRAM NO.1**

**NAME: KESHAV BHATT**

**COURSE: BTECH**

**BRANCH/SEMESTER: CSE / 5th**

**ROLL NO: 26**

**DATE:**

1. **OBJECTIVE**: WAP to find the roots of non-linear equation using Bisection method.

**2. METHOD**:

Step1: Define a function f(x)= x \* x \* x - x - 1=0

Step2: Enter two values a and b such that (f(a)\*f(b))<0

Step3:

Loop:

Iterate using

x = (a + b) / 2

if ((f0\*f2) < 0)

a=x

else b= x

Step4: if ((fabs(x-a)<=0.0001) || (fabs(x-b)<= 0.0001)) { break; }

Stop the procedure as desired solution of given equation x is achieved.

1. **PROGRAM:**

#include <stdio.h>

#include <math.h>

float f(float x)

{

    return (x \* x \* x - x - 1);

}

void bisect(float \*x, float a, float b, int \*itr)

{

    \*x = (a + b) / 2;

    ++(\*itr);

    printf("iteration no. %3dx=%75f\n", \*itr, \*x);

}

int main()

{

    int itr = 0, maxitr;

    float a, b, aerr, x, x1;

    printf("enter the value of a,b,allowed error,maximum iteration\n");

    scanf("%f%f%f%d", &a, &b, &aerr, &maxitr);

    bisect(&x, a, b, &itr);

    do{

        if (f(a) \* f(x) < 0){ b = x;}

        else

            a = x;

        bisect(&x1, a, b, &itr);

        if (fabs(x - x1) < aerr)

        {

            printf("after %d iteration,root=%f", itr, x1);

            return 0;

        }

        x = x1;

    } while (itr < maxitr);

    printf("solution does not converge");

    return (1);

}

**4. OUTPUT:-**

Enter the value of a, b, allowed error, maximum iteration

1

2

0.0001

15

iteration no. 1x=

1.500000

iteration no. 2x=

1.250000

iteration no. 3x=

1.375000

iteration no. 4x=

1.312500

iteration no. 5x=

1.343750

iteration no. 6x=

1.328125

iteration no. 7x=

1.320313

iteration no. 8x=

1.324219

iteration no. 9x=

1.326172

iteration no. 10x=

1.325195

iteration no. 11x=

1.324707

iteration no. 12x=

1.324951

iteration no. 13x=

1.324829

iteration no. 14x=

1.324768

after 14 iteration,root=1.324768

**PROGRAM NO.2**

**NAME: KESHAV BHATT**

**COURSE: BTECH**

**BRANCH/SEMESTER: CSE / 5th**

**ROLL NO: 26**

**DATE:**

1. **OBJECTIVE**: WAP to find the roots of non-linear equation using False position method.

**2. METHOD:**

**Step 1:**

Define a function f(x) = cos(x) - x = 0.

**Step 2:**

Enter two values x0 and x1 such that f(x0) \* f(x1) < 0.

**Step 3:**

Loop:

* Calculate the next approximation: x2=x0−(x1−x0)(f(x1)−f(x0))×f(x0)x2 = x0 - \frac{(x1 - x0)}{(f(x1) - f(x0))} \times f(x0)x2=x0−(f(x1)−f(x0))(x1−x0)​×f(x0)
* If f(x0) \* f(x2) < 0, then set x1 = x2.
* Else, set x0 = x2.

**Step 4:**

If the absolute difference fabs(x3 - x2) <= allowed error, stop the iteration as the desired root is found.

1. **PROGRAM:**

#include<stdio.h>

#include<math.h>

float f(float x)

{

return cos(x) – x;

}

void regula (float \*x, float x0, float x1, float fx0, float fx1, int \*itr)

{

\*x = x0 - ((x1 - x0) / (fx1 - fx0))\*fx0;

++(\*itr);

printf("Iteration no. %3d X = %7.5f \n", \*itr, \*x);

}

void main ()

{

int itr = 0, maxmitr;

float x0,x1,x2,x3,allerr;

printf("\nEnter the values of x0, x1, allowed error and maximum iterations:\n");

scanf("%f %f %f %d", &x0, &x1, &allerr, &maxmitr);

regula (&x2, x0, x1, f(x0), f(x1), &itr);

do

{

if (f(x0)\*f(x2) < 0)

x1=x2;

else

x0=x2;

regula (&x3, x0, x1, f(x0), f(x1), &itr);

if (fabs(x3-x2) < allerr)

{

printf("After %d iterations, root = %6.4f\n", itr, x3);

return 0;

}

x2=x3;

}

while (itr<maxmitr);

printf("Solution does not converge or iterations not sufficient:\n");

return 1;

}

**4. OUTPUT:-**

Enter the values of x0, x1, allowed error and maximum iterations:

0

1

0.0001

10

Iteration no. 1 X = 0.68507

Iteration no. 2 X = 0.73630

Iteration no. 3 X = 0.73895

Iteration no. 4 X = 0.73908

Iteration no. 5 X = 0.73908

After 5 iterations, root = 0.7391

**NAME: KESHAV BHATT**

**COURSE: BTECH**

**BRANCH/SEMESTER: CSE / 5th**

**ROLL NO: 26**

**DATE:**

1. **OBJECTIVE**: Write a program to find the roots of a non-linear equation using the **Newton-Raphson Method**.

**2. METHOD:**

**Step 1:**Define a function f(x)=x3−2x−5f(x) = x^3 - 2x - 5f(x)=x3−2x−5 and its derivative f′(x)=3x2−2f'(x) = 3x^2 - 2f′(x)=3x2−2.

**Step 2:**  
Enter an initial guess x0x\_0x0​, allowed error, and the maximum number of iterations.

**Step 3:**  
**Loop:**

* Calculate the next approximation using the Newton-Raphson formula:

x1=x0−f(x0)f′(x0)x\_1 = x\_0 - \frac{f(x\_0)}{f'(x\_0)}x1​=x0​−f′(x0​)f(x0​)​

* Print the current iteration and the value of x1x\_1x1​.
* If ∣h∣<|h| <∣h∣< allowed error, where h=f(x0)f′(x0)h = \frac{f(x\_0)}{f'(x\_0)}h=f′(x0​)f(x0​)​, stop the iteration and consider x1x\_1x1​ as the root.

**Step 4:**  
If the maximum number of iterations is reached without satisfying the stopping condition, print a message indicating that the solution did not converge.

**3. PROGRAM:**

#include <stdio.h>

#include <math.h>

float f(float x)

{

    return x \* x \* x - 2 \* x - 5;

}

float df(float x)

{

    return 3 \* x \* x - 2;

}

void main()

{

    int itr, maxmitr;

    float h, x0, x1, allerr;

    printf("\nEnter x0, allowed error, and maximum iterations\n");

    scanf("%f %f %d", &x0, &allerr, &maxmitr);

    for (itr = 1; itr <= maxmitr; itr++)

    {

        h = f(x0) / df(x0);

        x1 = x0 - h;

        printf(" At Iteration no. %3d, x = %9.6f\n", itr, x1);

        if (fabs(h) < allerr)

        {

            printf("After %3d iterations, root = %8.6f\n", itr, x1);

            return;

        }

        x0 = x1;

    }

    printf("The required solution does not converge or iterations are insufficient\n");

}

**4. OUTPUT:-**

Enter x0, allowed error, and maximum iterations

2

0.0001

10

At Iteration no. 1, x = 2.100000

At Iteration no. 2, x = 2.094568

At Iteration no. 3, x = 2.094552

After 3 iterations, root = 2.094552

**PROGRAM NO.4**

**NAME: KESHAV BHATT**

**COURSE: BTECH**

**BRANCH/SEMESTER: CSE / 5th**

**ROLL NO: 26**

**DATE:**

**1. OBJECTIVE**: WAP to find the roots of non-linear equation using Iteration method .

**2. METHOD**:

**Step 1: Define the Function**

* The function f(x)=cos(x)−3x+1f(x) = \cos(x) - 3x + 1f(x)=cos(x)−3x+1 is transformed to x=ϕ(x)=cos(x)+13x = \phi(x) = \frac{\cos(x) + 1}{3}x=ϕ(x)=3cos(x)+1​ for iteration.

**Step 2: Input Values**

* User provides aaa, bbb, allowed error, and maximum iterations. The initial guess x0x\_0x0​ is set as x0=a+b2x\_0 = \frac{a + b}{2}x0​=2a+b​.

**Step 3: Convergence Check**

* Check if ∣ϕ′(x0)∣>1|\phi'(x\_0)| > 1∣ϕ′(x0​)∣>1. If true, the iteration will not converge, and the program exits.

**Step 4: Iteration Process**

* Use xnew=ϕ(xold)x\_{\text{new}} = \phi(x\_{\text{old}})xnew​=ϕ(xold​) to update xxx and continue iterating.

**Step 5: Stopping Criterion**

* Stop when the difference between consecutive values ∣xnew−xold∣|x\_{\text{new}} - x\_{\text{old}}|∣xnew​−xold​∣ is less than the allowed error or when maximum iterations are reached.

**Step 6: Output**

* Print the root if the solution converges or report failure if it doesn’t.

**3. PROGRAM:**

#include <stdio.h>

#include <math.h>

float fun(float x) {

    return cos(x) - 3 \* x + 1;

}

float phi(float x) {

    return (cos(x) + 1) / 3;

}

float phidiff(float x) {

    return -sin(x) / 3;

}

int main()

{

    int max\_iterations;

    float a, b, x0, allerr;

    printf("Enter a , b ,allowed error and maximum iterations::\n");

    scanf("%f %f %f %d", &a, &b,&allerr, &max\_iterations);

    x0 = (a + b) / 2;

    if (fabs(phidiff(x0)) > 1)

    {

        printf("This phi function is not valid for this value of x0.\n");

        printf("Either change phi(x) or x.\n");

        return 0;

    }

    float prev = 0.0;

    int iterations = 0;

    do

    {

        prev = x0;

        x0 = phi(x0);

        iterations++;

        printf("At Iteration no. %3d, x = %9.6f\n", iterations, x0);

        if (fabs(x0 - prev) < allerr)

        {

            printf("After %3d iterations, root = %8.6f\n", iterations, x0);

            return 0;

        }

    } while (iterations < max\_iterations);

    printf("The required solution does not converge or iterations are insufficient\n");

    return 0;

}

**4. OUTPUT:-**

Enter a , b ,allowed error and maximum iterations:

0

1

0.0001

15

At Iteration no. 1, x = 0.62586199gf

At Iteration no. 2, x = 0.603486

At Iteration no. 3, x = 0.607787

At Iteration no. 4, x = 0.606971

At Iteration no. 5, x = 0.607126

At Iteration no. 6, x = 0.607097

After 6 iterations, root = 0.607097