**Page No. 1**

**PROGRAM NO.1**

**NAME: Daksh Purohit**

**COURSE: B.TECH**

**BRANCH/SEMESTER: CSE/5 Sem**

**ROLL NO: 23**

**DATE: 5/09/24**

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 - 2\*x -5=0

Step2: Enter two values x1 and x2 such that (f(x1)\*f(x2))<0

Step3:

Loop:

Iterate using

x = (x1 + x2) / 2

if ((f(x1)\*f(x2)) < 0)

x1=x

else x2= x

Step4: if ((fabs(x-x1)<=0.0001) || (fabs(x-x2)<= 0.0001)) { break; }

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

**3. PROGRAM:**

#include<stdio.h>

#include<math.h>

#define EPSILON 0.0001

float findValueAt(float x)

{

   return x\*x\*x - 2\*x -5;

}

float bisect(float x1,float x2)

{

    return (x1+x2)/2;

}

int main()

{

    int maxIteration,i=1;

    float x1,x2,x3,x;

    printf("Enter Maximum no of Iterations\n");

    scanf("%d",&maxIteration);

**Page No. 2**

   do

   {

        printf("Enter the value of x1 and x2(starting boundary->Initial Roots)");

        scanf("%f%f",&x1,&x2);

        if(findValueAt(x1)\*findValueAt(x2)>0)

        {

           printf("Roots are Invalid\n");

           continue;

        }

        else

        {

            printf("Roots Lie between %f and %f\n",x1,x2);

            break;

        }

    } while(1);

    x = bisect(x1,x2);

    do

    {

      if(findValueAt(x)\*findValueAt(x1)<0)

         x2=x;

      else

         x1=x;

      printf("Iterations=%d  Roots=%f\n",i,x);

      x3 = bisect(x1,x2);

      if(fabs(x3-x)<EPSILON)

      {

          printf("Root=%f  Total Iterations=%d",x,i);

          return 0;

      }

      x=x3;

      i++;

    }while(i<=maxIteration);

    printf("Root=%f  Total Iterations=%d",x,--i);

    return 0;

}

**Page No. 3**

**4. OUTPUT:-**

Enter Maximum no of Iterations

12

Enter the value of x1 and x2(starting boundary->Initial Roots) 2 3

Roots Lie between 2.000000 and 3.000000

Iterations=1 Roots=2.500000

Iterations=2 Roots=2.250000

Iterations=3 Roots=2.125000

Iterations=4 Roots=2.062500

Iterations=5 Roots=2.093750

Iterations=6 Roots=2.109375

Iterations=7 Roots=2.101563

Iterations=8 Roots=2.097656

Iterations=9 Roots=2.095703

Iterations=10 Roots=2.094727

Iterations=11 Roots=2.094238

Iterations=12 Roots=2.094482

Root=2.094604 Total Iterations=12

**Page No. 4**

**PROGRAM NO.2**

**NAME: Daksh Purohit**

**COURSE: B.TECH**

**BRANCH/SEMESTER: CSE/5 Sem**

**ROLL NO: 23**

**DATE: 5/09/24**

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

**2. METHOD**:

Step1: Define a function f(x)= x\*x\*x - 2\*x -5=0

Step2: Enter two values a and b such that (f(x1)\*f(x2))<0

Step3:

Loop:

**Iterate using:**

x=(x1\*f(x2)-x2\*f(x1))/(f(x2)-f(x1))

if ((f(x)\*f(x1)) < 0)

x1=x

else x2= x

**Else**:  
Set x1=xx\_1 = xx1​=x

Step4: if ((fabs(x-x1)<=0.0001) || (fabs(x-x2)<= 0.0001)) { break; }

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

**3. PROGRAM:**

#include<stdio.h>

#include<math.h>

#define EPSILON 0.0001

float findValueAt(float x)

{

   return x\*x\*x - 2\*x -5;

}

float findX(float x1,float x2)

{

      return (x1\*findValueAt(x2) - x2 \* findValueAt(x1)) / (findValueAt(x2) - findValueAt(x1));

}

int main()

{

    int maxIteration,i=1;

    float x1,x2,x3,x;

    printf("Enter Maximum no of Iterations\n");

    scanf("%d",&maxIteration);

   do

   {

        printf("Enter the value of x1 and x2(starting boundary->Initial Roots)");

        scanf("%f%f",&x1,&x2);

        if(findValueAt(x1)\*findValueAt(x2)>0)

        {

           printf("Roots are Invalid\n");

           continue;

        }

        else

        {

            printf("Roots Lie between %f and %f\n",x1,x2);

            break;

        }

    } while(1);

    x = findX(x1,x2);

    do

    {

      if(findValueAt(x)\*findValueAt(x1)<0)

         x2=x;

      else

         x1=x;

      printf("Iterations=%d  Root=%f\n",i,x);

      x3 = findX(x1,x2);

      if(fabs(x3-x)<EPSILON)

      {

          printf("Root=%f  Total Iterations=%d",x,i);

          return 0;

      }

      x=x3;

      i++;

    }while(i<=maxIteration);

    printf("Root=%f  Total Iterations=%d",x,--i);

    return 0;

}

**Page No. 6**

**4. OUTPUT:-**

Enter Maximum no of Iterations

14

Enter the value of x1 and x2(starting boundary->Initial Roots) 2 3

Roots Lie between 2.000000 and 3.000000

Iterations=1 Roots=2.500000

Iterations=2 Roots=2.250000

Iterations=3 Roots=2.125000

Iterations=4 Roots=2.062500

Iterations=5 Roots=2.093750

Iterations=6 Roots=2.109375

Iterations=7 Roots=2.101563

Iterations=8 Roots=2.097656

Iterations=9 Roots=2.095703

Iterations=10 Roots=2.094727

Iterations=11 Roots=2.094238

Iterations=12 Roots=2.094482

Iterations=13 Roots=2.094604

Root=2.094604 Total Iterations=13

**Page No. 6**

**PROGRAM NO.3**

**NAME: Daksh Purohit**

**COURSE: B.TECH**

**BRANCH/SEMESTER: CSE/5 Sem**

**ROLL NO: 23**

**DATE: 5/09/24**

**OBJECTIVE**: WAP to find the roots of non-linear equation using Newton’s Raphson method.

**2. METHOD**:

Step1: Define a function f(x)= x\*x\*x - 3\*x -5=0

Step2: Enter two values x1 and x2 such that (f(x1)\*f(x2))<0

Step3: Find x0:x0 is any point between x1 & x2

If(abs(f(x1))<abs(f(x2)))

x0=x1

else x0=x2

Step4:

Loop:

**Iterate using:**

 x = x0 - (f(x0)/differentiate(x0))    
**Else**:  
Set x1=xx\_1 = xx1​=x

Step5: if ((fabs(x-x0)<=0.0001)) { break; }

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

**3. PROGRAM:**

#include<stdio.h>

#include<math.h>

#define EPSILON 0.0001

float f(float x)

{

    return x\*x\*x - 3\*x -5;

}

float differentiate(float x)

{

    return 3\*x\*x - 3;

}

int main()

**Page No. 7**

{

    int maxIteration,i;

    float x1,x2,x,x0;

    printf("Enter Maximum no of Iterations\n");

    scanf("%d",&maxIteration);

   do

   {

        printf("Enter the value of x1 and x2(starting boundary)");

        scanf("%f%f",&x1,&x2);

        if(f(x1)\*f(x2)>0)

        {

           printf("Boundary Values are Invalid\n");

           continue;

        }

        else

        {

            printf("Roots Lie between %f and %f\n",x1,x2);

            break;

        }

    } while(1);

    if(fabs(f(x1)) < fabs(f(x2)))

        x0 = x1;

    else

        x1 = x2;

    for(i=1;i<=maxIteration;i++)

    {

      x = x0 - (f(x0)/differentiate(x0));

      if(fabs(x-x0)<EPSILON)

      {

            printf("Iterations=%d  Final Root=%f\n",i,x);

           return 0;

      }

      printf("Iterations=%d  Roots=%f\n",i,x);

      x0=x;

    }

    printf("Root=%f  Total Iterations=%d",x,--i);

    return 0;}

**Page No. 8**

**4. OUTPUT:-**

Enter Maximum no of Iterations

4

Enter the value of x1 and x2(starting boundary) 2 3

Roots Lie between 2.000000 and 3.000000

Iterations=1 Roots=2.333333

Iterations=2 Roots=2.280555

Iterations=3 Roots=2.279020

Iterations=4 Final Root=2.279019