Birla Institute of Technology, Mesra

NUMERICAL METHOD LAB
MA(204)

MODULE 1 LAB FILE

PREPARED BY: -

Robin Roy
B.Tech/15114/19
EEE

BISECTION METHOD

ALGORITHM: -

I. Start
II. Define function F(x) and error e=0.001.
III. Enter the initial guesses A and B.
IV. Calculate f1 & f2 where, f1=F(A) & f2=F(B).
V. Now check if f1*f2<0. If yes then go to step 5, else go to step 10.
VI. Calculate new approximated root as $x_0 = \frac{A+B}{2}$ and its corresponding value of function as f3.
VII. Then check
- If f3*f1 < 0 then B = x
- If f3*f1 > 0 then A = x

VIII. Repeat step 6 & 7 till (B-A) > e.
IX. Display x as root.
X. Stop

PROGRAM FOR ALGEBRIC: -

Find a real root of the equation **f (x) = x^3 + x - 1 = 0**, using Bisection method.

```
#include <stdio.h>

#include<math.h>

#define e 0.001

#define F(x) x*x*x + x -1


int main()

{

        int i;

        float A,B,x0;

        double f1,f2,f3;

        printf("\nENTER THE VALUE OF A:");

        scanf("%f",&A);
```

```
        printf("\nENTER THE VALUE OF B:");
        scanf("%f",&B);
        f1 = F(A);
        f2 = F(B);

        if (f1*f2>0)
        printf("REAL ROOT DOES NOT EXIST BETWEEN %f and %f",A,B);

else
        {
        printf("ITERATION NO.  \tVALUE OF A  \tVALUE OF B   \tVALUE OF X
\tVALUE OF F(X)");
            for(i=1;(B-A)>=e;i++)
             {
                  x0 = (A+B)/2;
                  f3 = F(x0);
                  printf("\n\t%d \t %f \t %f \t %f \t %f",i,A,B,x0,F(x0));
                  if(f3*f1<0)
                  {
                  B=x0;
                    }
                  else
                  {
                  A=x0;
                    }
            }
      printf("\n\n\nHENCE THE ROOT OF THE EQUATION IS %f",x0);
```

```
    }
        return 0;
}
```

Output: -

```
ENTER THE VALUE OF A: 0

ENTER THE VALUE OF B: 1
ITERATION NO.   VALUE OF A      VALUE OF B      VALUE OF X      VALUE OF F(X)
        1        0.000000        1.000000        0.500000        -0.375000
        2        0.500000        1.000000        0.750000        0.171875
        3        0.500000        0.750000        0.625000        -0.130859
        4        0.625000        0.750000        0.687500        0.012451
        5        0.625000        0.687500        0.656250        -0.061127
        6        0.656250        0.687500        0.671875        -0.024830
        7        0.671875        0.687500        0.679688        -0.006314
        8        0.679688        0.687500        0.683594        0.003037
        9        0.679688        0.683594        0.681641        -0.001646
        10       0.681641        0.683594        0.682617        0.000694


HENCE THE ROOT OF THE EQUATION IS 0.682617
--------------------------------
Process exited after 8.679 seconds with return value 0
Press any key to continue . . .
```

PROGRAM FOR TRANSIDENTAL: -

Find a root of **f (x) = x*e^x – 1 = 0**, using Bisection method, correct to three decimal places.

```
#include <stdio.h>

#include<math.h>

#define e 0.0001

#define F(x) x*exp(x) - 1


int main()

{

        int i;
```

```
float A,B,x0;
double f1,f2,f3;
printf("\nENTER THE VALUE OF A:");
scanf("%f",&A);
printf("\nENTER THE VALUE OF B:");
scanf("%f",&B);
f1 = F(A);
f2 = F(B);

if (f1*f2>0)
printf("REAL ROOT DOES NOT EXIST BETWEEN %f and %f",A,B);
else
{
        printf("ITERATION NO.  \tVALUE OF A  \tVALUE OF B   \tVALUE OF X
\tVALUE OF F(X)");
        for(i=1;(B-A)>=e;i++)
        {
        x0 = (A+B)/2;
        f3 = F(x0);
        printf("\n\t%d \t %f \t %f \t %f \t %f",i,A,B,x0,F(x0));
        if(f3*f1<0)
        {
        B=x0;
    }
        else
        {
        A=x0;
```

```
        }
    }
    printf("\n\n\nHENCE THE ROOT OF THE EQUATION IS %f",x0);
      }
       return 0;
}
```

Output: -

```
ENTER THE VALUE OF A: 0

ENTER THE VALUE OF B: 1
ITERATION NO.   VALUE OF A      VALUE OF B      VALUE OF X      VALUE OF F(X)
        1        0.000000        1.000000        0.500000        -0.175639
        2        0.500000        1.000000        0.750000        0.587750
        3        0.500000        0.750000        0.625000        0.167654
        4        0.500000        0.625000        0.562500        -0.012782
        5        0.562500        0.625000        0.593750        0.075142
        6        0.562500        0.593750        0.578125        0.030619
        7        0.562500        0.578125        0.570313        0.008780
        8        0.562500        0.570313        0.566406        -0.002035
        9        0.566406        0.570313        0.568359        0.003364
        10       0.566406        0.568359        0.567383        0.000662
        11       0.566406        0.567383        0.566895        -0.000687
        12       0.566895        0.567383        0.567139        -0.000013
        13       0.567139        0.567383        0.567261        0.000325
        14       0.567139        0.567261        0.567200        0.000156


HENCE THE ROOT OF THE EQUATION IS 0.567200
--------------------------------
Process exited after 4.924 seconds with return value 0
Press any key to continue . . .
```

REGULA FALSI METHOD

ALGORITHM: -

I. Start
II. Define function F(x) and error e=0.001.
III. Enter the initial guesses A and B. Also enter the maximum no. of iterations allowed.
IV. Calculate f1 & f2 where, f1=F(A) & f2=F(B).
V. Now check if f1*f2<0. If yes then go to step 5, else go to step 10.
VI. Calculate new approximated root as $x_0 = \frac{A*f2 - B*f1}{f2 - f1}$ and its corresponding value of function as f3.
VII. Then check
- If f3*f1 < 0 then B = x0 & f2 = f3.
- If f3*f1 > 0 then A = x0 & f1 = f3.

VIII. Repeat step 6 & 7 till the loop not reaches to maximum no. of iterations.
IX. Display x0 as root.
X. Stop

PROGRAM FOR ALGEBRIC: -

Find a real root of the equation **f (x) = x^3 – 2x – 5 = 0** by method of False position.

```
#include<stdio.h>

#include<math.h>

#define   F(x)   x*x*x - 2*x -5

#define e 0.001


int main()

{

        float A, B, x0, f1, f2, f3;

         int i,maxitr;

         printf("\nENTER THE VALUE OF A:");

        scanf("%f",&A);
```

```
printf("\nENTER THE VALUE OF B:");
scanf("%f",&B);
printf("\nENTER THE Maximum no. of iterations allowed:");
scanf("%d",&maxitr);
 f1 = F(A);
 f2 = F(B);
 if( f1*f2 >0)
 {
        printf("REAL ROOT DOES NOT EXIST BETWEEN %f and %f",A,B);
 }
 else
 {
 printf("\nITERATION NO. \t \tVALUE OF A  \tVALUE OF B   \tVALUE OF X
\tVALUE OF F(X)");
 for(i=1;i<=maxitr;i++)
 {
        x0 = ((A*f2)- (B*f1))/(f2-f1);
        f3 = F(x0);
        printf("\n\t%d\t\t%f\t%f\t%f\t%f\n",i, A, B, x0, f3);
       if(f1*f3 < 0)
        {
                B = x0;
                f2=f3;
        }
        else
        {
                A = x0;
```

```
                    f1=f3;
                }
        }
    printf("\n\n\nHENCE THE ROOT OF THE EQUATION IS %f",x0);
}
        return 0;
}
```

Output: -

```
ENTER THE VALUE OF A: 2

ENTER THE VALUE OF B: 3

ENTER THE Maximum no. of iterations allowed: 6

ITERATION NO.          VALUE OF A      VALUE OF B      VALUE OF X      VALUE OF F(X)
       1               2.000000        3.000000        2.058824        -0.390799

       2               2.058824        3.000000        2.081264        -0.147205

       3               2.081264        3.000000        2.089639        -0.054677

       4               2.089639        3.000000        2.092740        -0.020200

       5               2.092740        3.000000        2.093884        -0.007450

       6               2.093884        3.000000        2.094306        -0.002745


HENCE THE ROOT OF THE EQUATION IS 2.094306
--------------------------------
Process exited after 7.16 seconds with return value 0
Press any key to continue . . .
```

PROGRAM FOR TRANSIDENTAL: -

Determine the root of the equation **cos x – x*e^x = 0** by the method of False position.

```
#include<stdio.h>

#include<math.h>

#define   F(x)   cos(x) - x*exp(x)

#define e 0.001


int main()

{

        float A, B, x0, f1, f2, f3;

         int i,maxitr;


        printf("\nENTER THE VALUE OF A:");

        scanf("%f",&A);

        printf("\nENTER THE VALUE OF B:");

        scanf("%f",&B);

        printf("\nENTER THE Maximum no. of iterations allowed:");

        scanf("%d",&maxitr);

         f1 = F(A);

         f2 = F(B);

         if( f1*f2 >0)

         {

                printf("REAL ROOT DOES NOT EXIST BETWEEN %f and %f",A,B);

         }

         else

         {
```

```
        printf("\nITERATION NO. \t \tVALUE OF A  \tVALUE OF B   \tVALUE OF X
\tVALUE OF F(X)");

        for(i=1;i<=maxitr;i++)

        {

                x0 = ((A*f2)- (B*f1))/(f2-f1);

                f3 = F(x0);

                printf("\n\t%d\t\t%f\t%f\t%f\t%f\n",i, A, B, x0, f3);


                if(f1*f3 < 0)

                {

                        B = x0;

                        f2=f3;

                }

                else

                {

                        A = x0;

                        f1=f3;

                }

        }

      printf("\n\n\nHENCE THE ROOT OF THE EQUATION IS %f",x0);

}

        return 0;

}
```

Output: -

```
ENTER THE VALUE OF A: 0

ENTER THE VALUE OF B: 1

ENTER THE Maximum no. of iterations allowed: 7

ITERATION NO.          VALUE OF A      VALUE OF B      VALUE OF X      VALUE OF F(X)
       1               0.000000        1.000000        0.314665        0.519871

       2               0.314665        1.000000        0.446728        0.203545

       3               0.446728        1.000000        0.494015        0.070802

       4               0.494015        1.000000        0.509946        0.023608

       5               0.509946        1.000000        0.515201        0.007760

       6               0.515201        1.000000        0.516922        0.002539

       7               0.516922        1.000000        0.517485        0.000829


HENCE THE ROOT OF THE EQUATION IS 0.517485
--------------------------------
Process exited after 18.06 seconds with return value 0
Press any key to continue . . .
```

SECANT METHOD

ALGORITHM: -

I. Start
II. Define function F(x) and error e=0.001.
III. Enter the initial guesses A and B. Also enter the maximum no. of iterations allowed.
IV. Calculate f1 & f2 where, f1=F(A) & f2=F(B).
V. Now check if f1*f2<0. If yes then go to step 5, else go to step 10.
VI. Calculate new approximated root as $x_0 = \frac{A*f2 - B*f1}{f2 - f1}$ and its corresponding value of function as f3.
VII. Then change the values as follows:
- Make A = B and,
- B = x_0

VIII. Repeat step 4,6 & 7 till the loop not reaches to maximum no. of iterations.
IX. Display x_0 as root.
X. Stop

PROGRAM FOR ALGEBRIC: -

Find a real root of the equation **f (x) = x^3 – x – 1 = 0** by secant method.

```
#include<stdio.h>

#include<math.h>

#define   F(x)   x*x*x - x -1

#define e 0.001


int main()

{

        float A, B, x0, f1, f2, f3,C;

         int i,maxitr;


        printf("\nENTER THE VALUE OF A:");

        scanf("%f",&A);
```

```
printf("\nENTER THE VALUE OF B:");
scanf("%f",&B);
 f1 = F(A);
 f2 = F(B);
 printf("\nENTER THE Maximum no. of iterations allowed:");
scanf("%d",&maxitr);
 if( f1*f2 >0)
 {
        printf("REAL ROOT DOES NOT EXIST BETWEEN %f and %f",A,B);
 }
 else
 {
 printf("ITERATION NO. \t \tVALUE OF A  \tVALUE OF B   \tVALUE OF X
\tVALUE OF F(X)");
 for(i=1;i<=maxitr;i++)
 {
       f1 = F(A);
    f2 = F(B);
        x0 = ((A*f2)- (B*f1))/(f2-f1);
        f3 = F(x0);

        printf("\n\t%d\t\t%f\t%f\t%f\t%f\n",i, A, B, x0, f3);
        A = B;
        B=x0;

 }
printf("\n\n\nHENCE THE ROOT OF THE EQUATION IS %f",x0);
```

}

return 0;

}

Output: -

```
ENTER THE VALUE OF A: 1

ENTER THE VALUE OF B: 2

ENTER THE Maximum no. of iterations allowed: 5
ITERATION NO.           VALUE OF A      VALUE OF B      VALUE OF X      VALUE OF F(X)
        1               1.000000        2.000000        1.166667        -0.578704

        2               2.000000        1.166667        1.253112        -0.285363

        3               1.166667        1.253112        1.337207        0.053881

        4               1.253112        1.337207        1.323850        -0.003698

        5               1.337207        1.323850        1.324708        -0.000043


HENCE THE ROOT OF THE EQUATION IS 1.324708
--------------------------------
Process exited after 13.53 seconds with return value 0
Press any key to continue . . .
```

PROGRAM FOR TRANSIDENTAL: -

Determine the root of the equation **x – e^(-x)=0** by the secant method.

```
#include<stdio.h>

#include<math.h>

#define   F(x)   x - exp(-x)

#define e 0.001


int main()

{
```

```
float A, B, x0, f1, f2, f3,C;
int i,maxitr;

printf("\nENTER THE VALUE OF A:");
scanf("%f",&A);
printf("\nENTER THE VALUE OF B:");
scanf("%f",&B);
f1 = F(A);
f2 = F(B);
printf("\nENTER THE Maximum no. of iterations allowed:");
scanf("%d",&maxitr);
if( f1*f2 >0)
{
        printf("REAL ROOT DOES NOT EXIST BETWEEN %f and %f",A,B);
}
else
{
printf("ITERATION NO. \t \tVALUE OF A  \tVALUE OF B   \tVALUE OF X
\tVALUE OF F(X)");
for(i=1;i<=maxitr;i++)
{
        f1 = F(A);
    f2 = F(B);
        x0 = ((A*f2)- (B*f1))/(f2-f1);
        f3 = F(x0);

        printf("\n\t%d\t\t%f\t%f\t%f\t%f\n",i, A, B, x0, f3);
```

```
            A = B;
            B=x0;

        }
        printf("\n\n\nHENCE THE ROOT OF THE EQUATION IS %f",x0);
}
        return 0;
}
```

Output: -

```
ENTER THE VALUE OF A: 0

ENTER THE VALUE OF B: 1

ENTER THE Maximum no. of iterations allowed: 5
ITERATION NO.           VALUE OF A      VALUE OF B      VALUE OF X      VALUE OF F(X)
        1               0.000000        1.000000        0.612700        0.070814

        2               1.000000        0.612700        0.563838        -0.005182

        3               0.612700        0.563838        0.567170        0.000042

        4               0.563838        0.567170        0.567143        0.000000

        5               0.567170        0.567143        0.567143        0.000000



HENCE THE ROOT OF THE EQUATION IS 0.567143
--------------------------------
Process exited after 15.79 seconds with return value 0
Press any key to continue . . .
```

NEWTON RAPHSON METHOD

ALGORITHM: -

I. Start
II. Define function F(x) & DF(x) and error e=0.001.
III. Enter the initial guesses x_0. Also enter the maximum no. of iterations allowed.
IV. Now check if $DF(x_0)$ is not equal to 0. If yes then go to step 5, else go to step 10.
V. Calculate new approximated root as $x_{n+1} = x_n - \frac{F(x_n)}{DF(x_n)}$ and its corresponding value of function as f3.
VI. Then change the values $x=xx$.
VII. Repeat step 5 & 6 till the loop not reaches to maximum no. of iterations.
VIII. Display x_1 as root if $|F(x_n)/DF(x_n)|<e$.
IX. Stop

PROGRAM FOR ALGEBRIC: -

Find a real root of the equation **f (x) = x^3 + x - 1 = 0** by NRM.

```
#include <stdio.h>

#include <conio.h>

#include <math.h>

#define e 0.001


float F(float x)

{

   return x*x*x+x-1;

}

float DF (float x)

{

   return 3*x*x+1;

}
```

```
int main()
{
  int maxitr,i;
  float x0,x1,h;
  printf("\nENTER THE INITIAL VALUE OF A:\t");
    scanf("%f",&x0);
  printf("\nENTER THE  MAXIMUM NO. OF ITERATIONS ALLOWED:\t");
  scanf("%d",&maxitr);
  if(DF(x0)!=0)
  {
          printf("\n\nNRM IS APPLICABLE\n\n");
          printf("\t    ITERATION NO.    \t VALUE OF X    \t\t VALUE OF F(X)");
          for(i=1;i<=maxitr;i++)
          {
                  h= F(x0)/DF(x0);
                  x1=x0-h;
                  printf("\n\t\t %d    \t\t  %f    \t\t   %f",i,x1,F(x1));
       if (fabs(h) < e)
       {
           printf("\n\n\nHENCE THE ROOT OF THE EQUATION IS %f",x1);
       }
       x0=x1;
          }
  }
  else
```

```
  printf("NRM IS NOT APPLICABLE");
}
```

Output: -

```
ENTER THE INITIAL VALUE OF A:   0

ENTER THE  MAXIMUM NO. OF ITERATIONS ALLOWED:   5


NRM IS APPLICABLE

            ITERATION NO.        VALUE OF X             VALUE OF F(X)
                 1                1.000000               1.000000
                 2                0.750000               0.171875
                 3                0.686046               0.008941
                 4                0.682340               0.000028
                 5                0.682328               -0.000000


HENCE THE ROOT OF THE EQUATION IS 0.682328
--------------------------------
Process exited after 3.572 seconds with return value 0
Press any key to continue . . .
```

PROGRAM FOR TRANSIDENTAL: -

Determine the root of the equation **e^x * sinx - 1 = 0** by the NRM.

```
#include <stdio.h>

#include <conio.h>

#include <math.h>

#define e 0.001


float F(float x)

{

   return exp(x)*sin(x)-1;

}
```

```
float DF (float x)
{
   return exp(x)*sin(x) + exp(x)*cos(x);
}

int main()
{
  int maxitr,i;
  float x0,x1,h;
  printf("\nENTER THE INITIAL VALUE OF A:\t");
    scanf("%f",&x0);
  printf("\nENTER THE  MAXIMUM NO. OF ITERATIONS ALLOWED:\t");
  scanf("%d",&maxitr);
  if(DF(x0)!=0)
  {
          printf("\n\nNRM IS APPLICABLE\n\n");
          printf("\t    ITERATION NO.    \t VALUE OF X    \t\t VALUE OF F(X)");
          for(i=1;i<=maxitr;i++)
          {
                  h= F(x0)/DF(x0);
                  x1=x0-h;
                  printf("\n\t\t %d    \t\t  %f    \t\t   %f",i,x1,F(x1));
       if (fabs(h) < e)
       {
           printf("\n\n\nHENCE THE ROOT OF THE EQUATION IS %f",x1);
       }
       x0=x1;
```

```
        }
  }
  else
  printf("NRM IS NOT APPLICABLE");
}
```

Output: -

```
ENTER THE INITIAL VALUE OF A:   0

ENTER THE  MAXIMUM NO. OF ITERATIONS ALLOWED:   5


NRM IS APPLICABLE

            ITERATION NO.        VALUE OF X              VALUE OF F(X)
                 1                1.000000                1.287355
                 2                0.657258                0.178822
                 3                0.591183                0.006632
                 4                0.588537                0.000010
                 5                0.588533                0.000000


HENCE THE ROOT OF THE EQUATION IS 0.588533
--------------------------------
Process exited after 1.903 seconds with return value 0
Press any key to continue . . .
```

ITERATION METHOD

ALGORITHM: -

I. Start
II. Define function F(x), G(x) and error e=0.001.
III. Enter the initial guesses as x_0.
IV. Calculate new approximated root as $x_1 = g(x_0)$ and its corresponding value of function.
V. Then change the values as follows: $x_0 = x_1$
VI. Repeat step 4 & 5 till $|F(x_1)| > e$.
VII. Display x_1 as root.
VIII. Stop

PROGRAM FOR ALGEBRIC: -

Find a real root of the equation **f (x) = x^3 + x – 1 = 0** by iteration method.

```
#include<stdio.h>

#include<conio.h>

#include<math.h>

#define   f(x)   x*x*x + x -1

#define   g(x)   1/(1+x*x)

#define e 0.001


int main()

{

        int i,maxitr;

        float x0, x1;


        printf("Enter initial guess: ");

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

```
        printf("\n\tNo. of iterations\t\tx0\t\tx1\t\tf(x1)\n");
        for(i=1;fabs(f(x1)) > e;i++)
        {
         x1 = g(x0);
                printf("\t    %d\t\t\t    %f\t    %f\t     %f\n",i, x0, x1, f(x1));
                x0 = x1;

        }
         printf("\nRoot is %f", x1);
        getch();
        return(0);
}
```

Output: -

```
Enter initial guess: 0

        No. of iterations               x0              x1              f(x1)
                1                   0.000000        1.000000        1.000000
                2                   1.000000        0.500000        -0.375000
                3                   0.500000        0.800000        0.312000
                4                   0.800000        0.609756        -0.163535
                5                   0.609756        0.728968        0.116337
                6                   0.728968        0.653000        -0.068556
                7                   0.653000        0.701061        0.045624
                8                   0.701061        0.670472        -0.028129
                9                   0.670472        0.689878        0.018212
                10                  0.689878        0.677538        -0.011432
                11                  0.677538        0.685374        0.007319
                12                  0.685374        0.680394        -0.004627
                13                  0.680394        0.683557        0.002949
                14                  0.683557        0.681547        -0.001870
                15                  0.681547        0.682824        0.001189
                16                  0.682824        0.682013        -0.000755

Root is 0.682013
```

PROGRAM FOR TRANSIDENTAL: -

Determine the root of the equation **Cos x = 3*x - 1** by iteration method.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define   f(x)   cos(x) +1 -3*x
#define   g(x)   (cos(x)+1)/3
#define e 0.001

int main()
{
        int i,maxitr;
        float x0, x1;

        printf("Enter initial guess: ");
        scanf("%f", &x0);

        printf("\n\tNo. of iterations\t\tx0\t\tx1\t\tf(x1)\n");
        for(i=1;fabs(f(x1)) > e;i++)
        {
         x1 = g(x0);
                printf("\t    %d\t\t\t    %f\t    %f\t     %f\n",i, x0, x1, f(x1));
                x0 = x1;

        }

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

```
    getch();
    return(0);
}
```

Output: -

```
Enter initial guess: 0

        No. of iterations              x0              x1              f(x1)
             1                    0.000000        0.666667        -0.214113
             2                    0.666667        0.595296        0.042095
             3                    0.595296        0.609328        -0.007950
             4                    0.609328        0.606678        0.001514
             5                    0.606678        0.607182        -0.000288

Root is 0.607182
```