Your college name

Design with logo

Lab no.:

Lab title:

Submitted by:

Name:

Roll no.:

Group:

Submitted to:

Your Department name

Numerical Method

Submission date:

/* PROGRAMME TO IMPLEMENT BISECTION METHOD */

Objective: to find the root of the nonlinear equation by bisection method

Theory: briefly describe about the method with figure and formula whichever applicable.

Algorithm:

```
1. Start
2. Define function f(x)
3. Choose initial guesses x0 and x1 such that f(x0)f(x1) < 0
4. Choose pre-specified tolerable error e.
5. Calculate new approximated root as x2 = (x0 + x1)/2
6. Calculate f(x0)f(x2)
    a. if f(x0)f(x2) < 0 then x0 = x0 and x1 = x2
    b. if f(x0)f(x2) > 0 then x0 = x2 and x1 = x1
    c. if f(x0)f(x2) = 0 then goto (8)
7. if |f(x2)| > e then goto (5) otherwise goto (8)
8. Display x2 as root.
9. Stop
```

Programme: C –code for above programme

```c
#include<stdio.h>
#include<conio.h>
#include<math.h>
/*
 Defining equation to be solved.
 Change this equation to solve another problem.
*/
#define f(x) cos(x) - x * exp(x)

void main()
{
     float x0, x1, x2, f0, f1, f2, e;
     int step = 1;
     /* Inputs */
     up:
     printf("\nEnter two initial guesses:\n");
     scanf("%f%f", &x0, &x1);
     printf("Enter tolerable error:\n");
     scanf("%f", &e);
     /* Calculating Functional Value */
     f0 = f(x0);
     f1 = f(x1);
     /* Checking whether given guesses brackets the root or not. */
     if( f0 * f1 > 0.0)
     {
          printf("Incorrect Initial Guesses.\n");
          goto up;
     }
   /* Implementing Bisection Method */
     printf("\nStep\t\tx0\t\tx1\t\tx2\t\tf(x2)\n");
     do
     {
          x2 = (x0 + x1)/2;
          f2 = f(x2);
```

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

        if( f0 * f2 < 0)
        {
            x1 = x2;
            f1 = f2;
        }
        else
        {
            x0 = x2;
            f0 = f2;
        }
        step = step + 1;
    }while(fabs(f2)>e);
    printf("\nRoot is: %f", x2);
    getch();
}
```

Output: show the output
Discussion and conclusion: shortly conclude your report.

Exercise: find the root of $x^3 - 4x - 9$ using function.

/* Programme to implement false position method */

Objective: To find the root of the nonlinear equation by false position method
Theory: briefly describe about the method with figure and formula whichever applicable.
Algorithm:

```
1. Start
2. Take two initial guess ‘a’ and ‘b’ such that f(a)*f(b)<0
3. Define tolerable error (e).
4. Calculate c = a f(b)-b f(a) / f(b)-f(a)
5. If (f(c)*f(a)>0)
   {
     a=c;
   f(a)=f(c);
   }
   Else{
   b=c;
   f(b)=f(c);
   }
6. If |f(c)<e| goto 7
   Else goto 4
7. Stop and print root ‘c’.
```

c- code:

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

double f(double x) {
    // Define the function whose root is to be found
    return pow(x, 3) - x - 1;
```

```c
}

double false_position(double a, double b, double tol, int max_iter) {
    // Implement the false position method to find the root of the function
    double fa = f(a);
    double fb = f(b);
    double c, fc;
    int iter = 0;
    printf("Iteration\t a\t\t b\t\t c\t\t f(c)\n");
    do {
        c= (a*fb-b*fa)/(fb-fa);
        fc = f(c);
        printf("%d\t\t %f\t %f\t %f\t %f\n", iter, a, b, c, fc);
        if (fc*fa>0)
        {
            a=c;
            fa=fc;
        }
        else{
            b=c;
            fb=fc;
        }

        iter++;
    } while (fabs(fc) > tol && iter < max_iter);

    return c;
}

int main() {
    // Example usage
    double x,y;
    printf("enter the initial guess x: \n");
    scanf("%lf",&x);
    printf("enter the initia guess y: \n");
    scanf("%lf",&y);
    if (f(x)*f(y)<0)
    {
        double root = false_position(x, y, 0.0001, 100);
        printf("The root is: %f\n", root);
    }
    else{
        printf("wrong guess try again");
    }
    return 0;
}
```

Output: show the output

Discussion and conclusion: shortly conclude your report.

Exercise: implement same question using secant method.

/* PROGRAMME TO IMPLEMENT FIXED POINT ITERATION */

Objective: To find the root of the nonlinear equation by false position method
Theory: briefly describe about the method with figure and formula whichever applicable.
Algorithm:

```
1. Start
2. Define g(x)
3. Choose initial guess x0;
4. Find x=g(x0);
5. If |x-x0<e| goto 6
   Else
   x0=x and goto 4
6. Stop and print the root as 'x'.
```

c-code:

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

double f(double x) {
    // Define your function here
    return exp(-x) - x;
}

double g(double x) {
    // Define your g(x) function here
    return exp(-x);
}

int main() {
    double x0, x, tolerance;
    int iterations;

    // Read in initial guess, tolerance, and maximum iterations
    printf("Enter initial guess x0: ");
    scanf("%lf", &x0);
    printf("Enter tolerance: ");
    scanf("%lf", &tolerance);
    printf("Enter maximum iterations: ");
    scanf("%d", &iterations);

    // Perform fixed point iteration
    for (int i = 1; i <= iterations; i++) {
        x = g(x0);
        printf("Iteration %d: x = %lf, f(x) = %lf\n", i, x, f(x));
        if (fabs(x - x0) < tolerance) {
            printf("Converged to solution x = %lf after %d iterations.\n", x, i);
            return 0;
        }
        x0 = x;
    }
    printf("Failed to converge within %d iterations.\n", iterations);
    return 0;
}
```

Output: show the output

Discussion and conclusion: shortly conclude your report.

/* PROGRAMME TO IMPLEMENT NEWTON RAPHSON METHOD */

Objective: To find the root of the nonlinear equation by false position method

Theory: briefly describe about the method with figure and formula whichever applicable.

Algorithm:

1. Choose a starting value for x, denoted by x_0.
2. Compute the value of the function f(x_0) and its derivative f'(x_0) at x_0.
3. Compute the next estimate for the root using the formula: x_1 = x_0 - f(x_0) / f'(x_0).
4. Repeat steps 2-3 until the function value is sufficiently close to zero, i.e., until |f(x_n)| < epsilon, where epsilon is a small positive number that determines the desired accuracy of the root.
5. The final value of x is the estimated root of the function.

C code:

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

#define EPSILON 0.000001 //define your error here

double f(double x) {
    // Define your function here
    return x * log10(x) - 1.2;
}

double f_prime(double x) {
    // Define the derivative of your function here
    return 0.43429 + log10(x);
}

double newton_raphson(double x) {
    double x_next = x;
    double fx, fpx;
    int iter = 1;

    printf("iter\t   x\t\t   f(x)\n");

    while (1) {
        fx = f(x_next);
        fpx = f_prime(x_next);

        printf("%d\t%lf\t%lf\n", iter, x_next, fx);

        if (fabs(fx) < EPSILON) {
            break;
        }
        x_next = x_next - fx/fpx;
        iter++;
    } // run the loop until break staement is encountered.

    return x_next;
}

int main() {
    double root = newton_raphson(2); // Starting initial value for x
    printf("The root is: %lf", root);
    return 0;
}
```

Output: show the output

Discussion and conclusion: shortly conclude your report.

Question: in the above question start with initial guess ‘0’.