Nonlinear Solvers

Morgan Monzingo

10.7.14

MATH 3316

There are many times in higher level math, functions are given that have roots, x-intercepts, that are unable to be solved for. When this occurs we need other ways to be able to approximate the roots. Therefore we to use methods like the Newton method and Steffensen's method to be able to solve for the unknown roots.

Part A:

In part a we are able to evaluate the roots of the function by using the Newton method. The Newton method uses an x value, takes the tangent line from that position and then finds the root to the tangent line. If that value of x is within the tolerance provided then that is the approximated root, otherwise we repeat the process with the current x.

To begin we create a new c++ file that tests the function nine different times, either changing the first test value or the tolerance. The instructor gave us the function f(x) ≡.2(x - 5)(x + 2)(x + 3) and from there we calculated the derivative to be f'(x) = .6(x^2) - 3.8. The x values and tolerances were given to be x = {–1, 2, 3}, and ε = {10–3, 10–7, 10–11}.

For each call of the Newton method, you have to pass the function f, f', x, the maximum number of iterations, the tolerance, whether we are showing the iterations and data, any auxiliary data needed. First we show the initial x, tolerance, and f(x) value. Then in a for loop we solve for the root approximation. The x value that needs to be tested is exalted to be (current x) - f/f. Then h, the solution update, is calculated to be the absolute value of f/f'. Since we are showing the iteration for the Newton method, i, the new x, the absolute value of h and the absolute value of f(x) are printed out. Then if the solution update is less than the tolerance we print out the final x and are finished.

The output values show that most guesses get very close to the actual root. if they aren't the root then they are just thousands off from the root. The values do change based off of the x and tolerances passed, the more tolerance, usually the closer to the correct value we get. When we change the initial x guess, the results change to be different roots in the function. Since the function has three roots our test finds the closest root based off of iteration 0 for our guess of x, the guesses don't always find the same root. If we look at the guess x = 2, the next step in the newton method gives us x = -6.57143 which is closest to the root x = -3. When we vary the tolerance of our guesses, our predictions are slightly different. For example when the x guess is 3 for each tolerance we have the root to be 5, 5, and 5.0001. So the answers are close, but there is slight variation in how exact we want our answers to be. When tolerance is small our answer is a little off from what it should be but when the tolerance is larger our answer for the approximated root is closer to the actual value.

Part B:

In part b we are approximating the root in a very similar was as we did in part a. However for Steffensen's method, we do not pass a derivative of f, we approximate the derivative using the equation provided by the instructor, $f'(x) \approx \delta^{-}f(x) = (f(x) - f(x - \alpha)) / \alpha$ where α is approximated to be $f(x)$ to account for error, so the equation becomes $f'(x) \approx (f(x) - f(x - f(x))) / f(x)$. The reason we use this method is if the function is too complicated to easily solve for it's derivative.

The test_steffensen method is built just like the test_newton method, we call Steffensen's method nine times either changing the initial x or the tolerance, the values are provided and are the same as before, $x = \{-1, 2, 3\}$, and $\varepsilon = \{10–3, 10–7, 10–11\}$. We also still use the same function as mentioned above, $f(x) \equiv .2(x - 5)(x + 2)(x + 3)$.

In steffensen.cpp we pass in f, x, the max number of iterations, tolerance, if we are showing iterations and the data pointer. Then I printed out the initial information as done before. I then created a for loop to run through the approximation process. First the derivative is approximated and then we find the next x guess using xnew = x - f(x,data)/the calculated derivative. Next h is calculated as the absolute value of the function f divided by the derivative calculated. Then all of the information is printed out. If the h value is less than the tolerance, we then return the new x value as the root.

The output for Steffensen's method is very similar to that of the Newton method. Steffensen's method has more iterations for each of it's x guesses and also the values are closer to the actual roots. However there is an anomaly with x = 2 , tolerance = 1e-11, the root is calculated to be nan which is not a number. This occurs because we are using float values and the last two iterations of this guess are not recognizable. Other than that one difference, the roots calculated for Steffensen's method and for Newton's method are the same. Newton's method is just more efficient in it's calculations because it only approximates the root, while Steffensen's method approximated the root and the derivative of the function which leaves room for error.

```
//Morgan Monzingo
//Project 2
//Part A

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

using namespace std;

double f(const double x, void *data);
double df(const double x, void *data);
double newton(double (*f)(const double, void *data),
              double (*df)(const double, void *data),
              double x, int maxit, double tol,
              bool show_iterates, void *data);

//tests the newton.cpp function
int main(int argc, char* argv[])
{
    //running Newton's method nine times based on the three x values and three tolerances
    double x = -1;
    double tol = .001;
    int maxit = 15;
    double rt_one = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_one << endl << endl;

    tol = 1e-07;
    rt_one = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_one << endl << endl;

    tol = .00000000001;
    rt_one = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_one << endl << endl;

    x = 2;
    tol = .001;
    double rt_two = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_two << endl << endl;

    tol = 1e-07;
    rt_two = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_two << endl << endl;

    tol = .00000000001;
    rt_two = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_two << endl << endl;

    x = 3;
    tol = .001;
    double rt_three = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_three << endl << endl;

    tol = .0000001;
    rt_three = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_three << endl << endl;

    tol = .00000000001;
    rt_three = newton(f, df, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_three << endl << endl;


}

//calculates the function f
double f(const double x, void *data)
{
    return (.2*(x - 5.0)*(x + 2.0)*(x + 3.0));
}

//calculates the derivative df
double df(const double x, void *data)
{
    return(.6*x*x-3.8);
}
```

```cpp
//Morgan Monzingo
//Project 2
#include <iostream>
#include <math.h>

using namespace std;

double newton(double (*f)(const double, void *data), double (*df)(const double, void *data),
              double x, int maxit, double tol, bool show_iterates, void *data)
{
    double xStep = 0.0;
    double h = 0.0;

        cout << "Xo = " << x << " tol= " << tol << endl;
        cout << "Initial f(x)" << f(x,data) << endl;


    for(int i = 0; i < maxit; i++)
    {
        //just caluculate the new x value
        xStep = x - f(x, data)/df(x, data);
        h = fabs(f(x, data)/df(x, data));

        //show iterations if it's a true
        if(show_iterates)
        {
            cout << "  iter " << i << ", x= " << xStep << ", |h|= " << h << ", |f(x)|= " << fabs(f(xStep, data)) << endl;
        }

        //break out if less than tolerance
        if(h < tol)
        {
            break;
        }

        x = xStep;

    }

    return xStep;
}
```

```
Xo = -1 tol= 0.001
Initial f(x)-2.4
    iter 0, x= -1.75, |h|= 0.75, |f(x)|= 0.421875
    iter 1, x= -1.96497, |h|= 0.214968, |f(x)|= 0.0505087
    iter 2, x= -1.99902, |h|= 0.0340506, |f(x)|= 0.00137486
    iter 3, x= -2, |h|= 0.000980394, |f(x)|= 1.15303e-06
 The approximate root is -2
```

```
Xo = -1 tol= 1e-07
Initial f(x)-2.4
    iter 0, x= -1.75, |h|= 0.75, |f(x)|= 0.421875
    iter 1, x= -1.96497, |h|= 0.214968, |f(x)|= 0.0505087
    iter 2, x= -1.99902, |h|= 0.0340506, |f(x)|= 0.00137486
    iter 3, x= -2, |h|= 0.000980394, |f(x)|= 1.15303e-06
    iter 4, x= -2, |h|= 8.23591e-07, |f(x)|= 8.13838e-13
    iter 5, x= -2, |h|= 5.81313e-13, |f(x)|= 0
 The approximate root is -2
```

```
Xo = -1 tol= 1e-11
```

```
Initial f(x)-2.4
    iter 0, x= -1.75, |h|= 0.75, |f(x)|= 0.421875
    iter 1, x= -1.96497, |h|= 0.214968, |f(x)|= 0.0505087
    iter 2, x= -1.99902, |h|= 0.0340506, |f(x)|= 0.00137486
    iter 3, x= -2, |h|= 0.000980394, |f(x)|= 1.15303e-06
    iter 4, x= -2, |h|= 8.23591e-07, |f(x)|= 8.13838e-13
    iter 5, x= -2, |h|= 5.81313e-13, |f(x)|= 0
 The approximate root is -2

Xo = 2 tol= 0.001
Initial f(x)-12
    iter 0, x= -6.57143, |h|= 8.57143, |f(x)|= 37.7843
    iter 1, x= -4.86252, |h|= 1.70891, |f(x)|= 10.5164
    iter 2, x= -3.85001, |h|= 1.01251, |f(x)|= 2.78338
    iter 3, x= -3.30356, |h|= 0.546452, |f(x)|= 0.657155
    iter 4, x= -3.06443, |h|= 0.239131, |f(x)|= 0.11061
    iter 5, x= -3.00413, |h|= 0.0602968, |f(x)|= 0.00664097
    iter 6, x= -3.00002, |h|= 0.00411235, |f(x)|= 3.04686e-05
    iter 7, x= -3, |h|= 1.90421e-05, |f(x)|= 6.52683e-10
 The approximate root is -3

Xo = 2 tol= 1e-07
Initial f(x)-12
    iter 0, x= -6.57143, |h|= 8.57143, |f(x)|= 37.7843
    iter 1, x= -4.86252, |h|= 1.70891, |f(x)|= 10.5164
    iter 2, x= -3.85001, |h|= 1.01251, |f(x)|= 2.78338
    iter 3, x= -3.30356, |h|= 0.546452, |f(x)|= 0.657155
    iter 4, x= -3.06443, |h|= 0.239131, |f(x)|= 0.11061
    iter 5, x= -3.00413, |h|= 0.0602968, |f(x)|= 0.00664097
    iter 6, x= -3.00002, |h|= 0.00411235, |f(x)|= 3.04686e-05
    iter 7, x= -3, |h|= 1.90421e-05, |f(x)|= 6.52683e-10
    iter 8, x= -3, |h|= 4.07927e-10, |f(x)|= 0
 The approximate root is -3

Xo = 2 tol= 1e-11
Initial f(x)-12
    iter 0, x= -6.57143, |h|= 8.57143, |f(x)|= 37.7843
    iter 1, x= -4.86252, |h|= 1.70891, |f(x)|= 10.5164
    iter 2, x= -3.85001, |h|= 1.01251, |f(x)|= 2.78338
    iter 3, x= -3.30356, |h|= 0.546452, |f(x)|= 0.657155
    iter 4, x= -3.06443, |h|= 0.239131, |f(x)|= 0.11061
    iter 5, x= -3.00413, |h|= 0.0602968, |f(x)|= 0.00664097
    iter 6, x= -3.00002, |h|= 0.00411235, |f(x)|= 3.04686e-05
    iter 7, x= -3, |h|= 1.90421e-05, |f(x)|= 6.52683e-10
    iter 8, x= -3, |h|= 4.07927e-10, |f(x)|= 0
    iter 9, x= -3, |h|= 0, |f(x)|= 0
 The approximate root is -3

Xo = 3 tol= 0.001
```

```
Initial f(x)-12
    iter 0, x= 10.5, |h|= 7.5, |f(x)|= 185.625
    iter 1, x= 7.52285, |h|= 2.97715, |f(x)|= 50.5619
    iter 2, x= 5.84618, |h|= 1.67668, |f(x)|= 11.7464
    iter 3, x= 5.14308, |h|= 0.703098, |f(x)|= 1.66451
    iter 4, x= 5.00519, |h|= 0.137896, |f(x)|= 0.0581538
    iter 5, x= 5.00001, |h|= 0.00517791, |f(x)|= 8.04879e-05
    iter 6, x= 5, |h|= 7.18639e-06, |f(x)|= 1.54934e-10
 The approximate root is 5

Xo = 3 tol= 1e-07
Initial f(x)-12
    iter 0, x= 10.5, |h|= 7.5, |f(x)|= 185.625
    iter 1, x= 7.52285, |h|= 2.97715, |f(x)|= 50.5619
    iter 2, x= 5.84618, |h|= 1.67668, |f(x)|= 11.7464
    iter 3, x= 5.14308, |h|= 0.703098, |f(x)|= 1.66451
    iter 4, x= 5.00519, |h|= 0.137896, |f(x)|= 0.0581538
    iter 5, x= 5.00001, |h|= 0.00517791, |f(x)|= 8.04879e-05
    iter 6, x= 5, |h|= 7.18639e-06, |f(x)|= 1.54934e-10
    iter 7, x= 5, |h|= 1.38334e-11, |f(x)|= 0
 The approximate root is 5

Xo = 3 tol= 1e-11
Initial f(x)-12
    iter 0, x= 10.5, |h|= 7.5, |f(x)|= 185.625
    iter 1, x= 7.52285, |h|= 2.97715, |f(x)|= 50.5619
    iter 2, x= 5.84618, |h|= 1.67668, |f(x)|= 11.7464
    iter 3, x= 5.14308, |h|= 0.703098, |f(x)|= 1.66451

    iter 4, x= 5.00519, |h|= 0.137896, |f(x)|= 0.0581538
    iter 5, x= 5.00001, |h|= 0.00517791, |f(x)|= 8.04879e-05
    iter 6, x= 5, |h|= 7.18639e-06, |f(x)|= 1.54934e-10
    iter 7, x= 5, |h|= 1.38334e-11, |f(x)|= 0
    iter 8, x= 5, |h|= 0, |f(x)|= 0
 The approximate root is 5
```

```
//Morgan Monzingo
//Project 2
//Part B
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <math.h>

using namespace std;

double f(const double x, void *data);
double steffensen(double (*f)(const double, void *data), double x, int maxit, double tol,
                   bool show_iterates, void *data);

//tests the steffensen.cpp function
int main(int argc, char* argv[])
{
    //running newton nine times based on the three x values and three tolerances
    double x = -1;
    double tol = 10e-3;
    int maxit = 20;
    double rt_one = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_one << endl << endl;

    tol = 10e-7;
    rt_one = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_one << endl << endl;

    tol = 10e-11;
    rt_one = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_one << endl << endl;

    x = 2;
    tol = 10e-3;
    double rt_two = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_two << endl << endl;

    tol = 10e-7;
    rt_two = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_two << endl << endl;

    tol = 10e-11;
    rt_two = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_two << endl << endl;

    x = 3;
    tol = 10e-3;
    double rt_three = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_three << endl << endl;

    tol = 10e-7;
    rt_three = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_three << endl << endl;

    tol = 10e-11;
    rt_three = steffensen(f, x, maxit, tol, true, NULL);
    cout << " The approximate root is " << rt_three << endl << endl;


}

//The function f
double f(const double x, void *data)
{
    return (.2*(x - 5.0)*(x + 2.0)*(x + 3.0));
}

```

```cpp
//Morgan Monzingo
//Project 2
#include <iostream>
#include <math.h>

using namespace std;

double steffensen(double (*f)(const double, void *data), double x, int maxit, double tol, bool show_iterates, void *data)
{
    double xStep = 0.0;
    double df = 0.0;
    double h = 0.0;
    //initial print out of information passed
    cout << "Xo = " << x << " tol= " << tol << endl;
    cout << "Initial f(x) = " << f(x,data) << endl;

    //for loop to actually execute Steffensen's method
    for(int i = 0; i < maxit; i++)
    {
        //solving for the derivative
        df = (f(x,data) - f(x-f(x,data),data))/f(x,data);
        xStep = x - f(x, data)/df;
        h = fabs(f(x,data)/df);

        //printing out if necessary
        if(show_iterates)
        {
            cout << "  iter " << i << ", x = " << xStep << ", |h| = " << h << ", |f(x)| = " << fabs(f(xStep, data)) << endl;
        }

        //break out of program if you are less than the tolerance
        if(h < tol)
        {
            break;
        }

        x = xStep;

    }

    return xStep;
}
```

```
Xo = -1 tol= 0.01
Initial f(x) = -2.4
    iter 0, x = -1.68807, |h| = 0.688073, |f(x)| = 0.547385
    iter 1, x = -1.89985, |h| = 0.211776, |f(x)| = 0.152046
    iter 2, x = -1.98418, |h| = 0.0843276, |f(x)| = 0.0224516
    iter 3, x = -1.99951, |h| = 0.0153311, |f(x)| = 0.000688776
    iter 4, x = -2, |h| = 0.000491279, |f(x)| = 6.95489e-07
 The approximate root is -2
```

```
Xo = -1 tol= 1e-06
Initial f(x) = -2.4
    iter 0, x = -1.68807, |h| = 0.688073, |f(x)| = 0.547385
    iter 1, x = -1.89985, |h| = 0.211776, |f(x)| = 0.152046
    iter 2, x = -1.98418, |h| = 0.0843276, |f(x)| = 0.0224516
    iter 3, x = -1.99951, |h| = 0.0153311, |f(x)| = 0.000688776
    iter 4, x = -2, |h| = 0.000491279, |f(x)| = 6.95489e-07
    iter 5, x = -2, |h| = 4.96777e-07, |f(x)| = 7.10632e-13
 The approximate root is -2
```

```
Xo = -1 tol= 1e-10
Initial f(x) = -2.4
    iter 0, x = -1.68807, |h| = 0.688073, |f(x)| = 0.547385
    iter 1, x = -1.89985, |h| = 0.211776, |f(x)| = 0.152046
    iter 2, x = -1.98418, |h| = 0.0843276, |f(x)| = 0.0224516
    iter 3, x = -1.99951, |h| = 0.0153311, |f(x)| = 0.000688776
    iter 4, x = -2, |h| = 0.000491279, |f(x)| = 6.95489e-07
```

```
    iter 5, x = -2, |h| = 4.96777e-07, |f(x)| = 7.10632e-13
    iter 6, x = -2, |h| = 5.07657e-13, |f(x)| = 0
 The approximate root is -2

Xo = 2 tol= 0.01
Initial f(x) = -12
    iter 0, x = 2.28708, |h| = 0.287081, |f(x)| = 12.2983
    iter 1, x = 2.55176, |h| = 0.264682, |f(x)| = 12.3735
    iter 2, x = 2.80087, |h| = 0.249103, |f(x)| = 12.2488
    iter 3, x = 3.03872, |h| = 0.23785, |f(x)| = 11.9353
    iter 4, x = 3.26828, |h| = 0.229563, |f(x)| = 11.4373
    iter 5, x = 3.49167, |h| = 0.223386, |f(x)| = 10.7544
    iter 6, x = 3.71035, |h| = 0.218688, |f(x)| = 9.88346
    iter 7, x = 3.92521, |h| = 0.214861, |f(x)| = 8.82041
    iter 8, x = 4.13634, |h| = 0.211128, |f(x)| = 7.56409
    iter 9, x = 4.34255, |h| = 0.206211, |f(x)| = 6.12352
    iter 10, x = 4.54028, |h| = 0.197729, |f(x)| = 4.53424
    iter 11, x = 4.72142, |h| = 0.181135, |f(x)| = 2.89162
    iter 12, x = 4.87017, |h| = 0.148754, |f(x)| = 1.40394
    iter 13, x = 4.96422, |h| = 0.0940486, |f(x)| = 0.396892
    iter 14, x = 4.99675, |h| = 0.0325328, |f(x)| = 0.0363245
    iter 15, x = 4.99997, |h| = 0.00321749, |f(x)| = 0.000320199
 The approximate root is 4.99997

Xo = 2 tol= 1e-06
Initial f(x) = -12
    iter 0, x = 2.28708, |h| = 0.287081, |f(x)| = 12.2983
    iter 1, x = 2.55176, |h| = 0.264682, |f(x)| = 12.3735
    iter 2, x = 2.80087, |h| = 0.249103, |f(x)| = 12.2488
    iter 3, x = 3.03872, |h| = 0.23785, |f(x)| = 11.9353
    iter 4, x = 3.26828, |h| = 0.229563, |f(x)| = 11.4373
    iter 5, x = 3.49167, |h| = 0.223386, |f(x)| = 10.7544
    iter 6, x = 3.71035, |h| = 0.218688, |f(x)| = 9.88346
    iter 7, x = 3.92521, |h| = 0.214861, |f(x)| = 8.82041
    iter 8, x = 4.13634, |h| = 0.211128, |f(x)| = 7.56409
    iter 9, x = 4.34255, |h| = 0.206211, |f(x)| = 6.12352
    iter 10, x = 4.54028, |h| = 0.197729, |f(x)| = 4.53424
    iter 11, x = 4.72142, |h| = 0.181135, |f(x)| = 2.89162
    iter 12, x = 4.87017, |h| = 0.148754, |f(x)| = 1.40394
    iter 13, x = 4.96422, |h| = 0.0940486, |f(x)| = 0.396892
    iter 14, x = 4.99675, |h| = 0.0325328, |f(x)| = 0.0363245
    iter 15, x = 4.99997, |h| = 0.00321749, |f(x)| = 0.000320199
    iter 16, x = 5, |h| = 2.85872e-05, |f(x)| = 2.50095e-08
    iter 17, x = 5, |h| = 2.23299e-09, |f(x)| = 0
 The approximate root is 5

Xo = 2 tol= 1e-10
Initial f(x) = -12
    iter 0, x = 2.28708, |h| = 0.287081, |f(x)| = 12.2983
```

```
    iter 1, x = 2.55176, |h| = 0.264682, |f(x)| = 12.3735
    iter 2, x = 2.80087, |h| = 0.249103, |f(x)| = 12.2488
    iter 3, x = 3.03872, |h| = 0.23785, |f(x)| = 11.9353
    iter 4, x = 3.26828, |h| = 0.229563, |f(x)| = 11.4373
    iter 5, x = 3.49167, |h| = 0.223386, |f(x)| = 10.7544
    iter 6, x = 3.71035, |h| = 0.218688, |f(x)| = 9.88346
    iter 7, x = 3.92521, |h| = 0.214861, |f(x)| = 8.82041
    iter 8, x = 4.13634, |h| = 0.211128, |f(x)| = 7.56409
    iter 9, x = 4.34255, |h| = 0.206211, |f(x)| = 6.12352
    iter 10, x = 4.54028, |h| = 0.197729, |f(x)| = 4.53424
    iter 11, x = 4.72142, |h| = 0.181135, |f(x)| = 2.89162
    iter 12, x = 4.87017, |h| = 0.148754, |f(x)| = 1.40394
    iter 13, x = 4.96422, |h| = 0.0940486, |f(x)| = 0.396892
    iter 14, x = 4.99675, |h| = 0.0325328, |f(x)| = 0.0363245
    iter 15, x = 4.99997, |h| = 0.00321749, |f(x)| = 0.000320199
    iter 16, x = 5, |h| = 2.85872e-05, |f(x)| = 2.50095e-08
    iter 17, x = 5, |h| = 2.23299e-09, |f(x)| = 0
    iter 18, x = nan, |h| = nan, |f(x)| = nan
    iter 19, x = nan, |h| = nan, |f(x)| = nan
 The approximate root is nan

Xo = 3 tol= 0.01
Initial f(x) = -12
    iter 0, x = 3.23077, |h| = 0.230769, |f(x)| = 11.5325
    iter 1, x = 3.45506, |h| = 0.224291, |f(x)| = 10.8803
    iter 2, x = 3.67445, |h| = 0.219388, |f(x)| = 10.0407
    iter 3, x = 3.88991, |h| = 0.215463, |f(x)| = 9.0097
    iter 4, x = 4.10169, |h| = 0.211784, |f(x)| = 7.78514
    iter 5, x = 4.30888, |h| = 0.207187, |f(x)| = 6.37361
    iter 6, x = 4.50842, |h| = 0.199535, |f(x)| = 4.80453
    iter 7, x = 4.69314, |h| = 0.184728, |f(x)| = 3.16008
    iter 8, x = 4.84873, |h| = 0.155585, |f(x)| = 1.62628
    iter 9, x = 4.9532, |h| = 0.104474, |f(x)| = 0.517571
    iter 10, x = 4.99457, |h| = 0.041365, |f(x)| = 0.0607438
    iter 11, x = 4.99992, |h| = 0.00535178, |f(x)| = 0.000892291
 The approximate root is 4.99992

Xo = 3 tol= 1e-06
Initial f(x) = -12
    iter 0, x = 3.23077, |h| = 0.230769, |f(x)| = 11.5325
    iter 1, x = 3.45506, |h| = 0.224291, |f(x)| = 10.8803
    iter 2, x = 3.67445, |h| = 0.219388, |f(x)| = 10.0407
    iter 3, x = 3.88991, |h| = 0.215463, |f(x)| = 9.0097
    iter 4, x = 4.10169, |h| = 0.211784, |f(x)| = 7.78514
    iter 5, x = 4.30888, |h| = 0.207187, |f(x)| = 6.37361
    iter 6, x = 4.50842, |h| = 0.199535, |f(x)| = 4.80453
    iter 7, x = 4.69314, |h| = 0.184728, |f(x)| = 3.16008
    iter 8, x = 4.84873, |h| = 0.155585, |f(x)| = 1.62628
    iter 9, x = 4.9532, |h| = 0.104474, |f(x)| = 0.517571
```

```
    iter 10, x = 4.99457, |h| = 0.041365, |f(x)| = 0.0607438
    iter 11, x = 4.99992, |h| = 0.00535178, |f(x)| = 0.000892291
    iter 12, x = 5, |h| = 7.96532e-05, |f(x)| = 1.94197e-07
    iter 13, x = 5, |h| = 1.7339e-08, |f(x)| = 9.9476e-15
 The approximate root is 5
```

```
Xo = 3 tol= 1e-10
Initial f(x) = -12
    iter 0, x = 3.23077, |h| = 0.230769, |f(x)| = 11.5325
    iter 1, x = 3.45506, |h| = 0.224291, |f(x)| = 10.8803
    iter 2, x = 3.67445, |h| = 0.219388, |f(x)| = 10.0407
    iter 3, x = 3.88991, |h| = 0.215463, |f(x)| = 9.0097
    iter 4, x = 4.10169, |h| = 0.211784, |f(x)| = 7.78514
    iter 5, x = 4.30888, |h| = 0.207187, |f(x)| = 6.37361
    iter 6, x = 4.50842, |h| = 0.199535, |f(x)| = 4.80453
    iter 7, x = 4.69314, |h| = 0.184728, |f(x)| = 3.16008
    iter 8, x = 4.84873, |h| = 0.155585, |f(x)| = 1.62628
    iter 9, x = 4.9532, |h| = 0.104474, |f(x)| = 0.517571
    iter 10, x = 4.99457, |h| = 0.041365, |f(x)| = 0.0607438
    iter 11, x = 4.99992, |h| = 0.00535178, |f(x)| = 0.000892291
    iter 12, x = 5, |h| = 7.96532e-05, |f(x)| = 1.94197e-07
    iter 13, x = 5, |h| = 1.7339e-08, |f(x)| = 9.9476e-15
    iter 14, x = 5, |h| = 9.04327e-16, |f(x)| = 0
 The approximate root is 5
```

```
#Morgan Monzingo
#makefile

total: partA partB

partA:
		g++ newton.cpp test_newton.cpp -o partA.exe

partB:
		g++ steffensen.cpp test_steffensen.cpp -o partB.exe

clean:
		rm *.exe
```