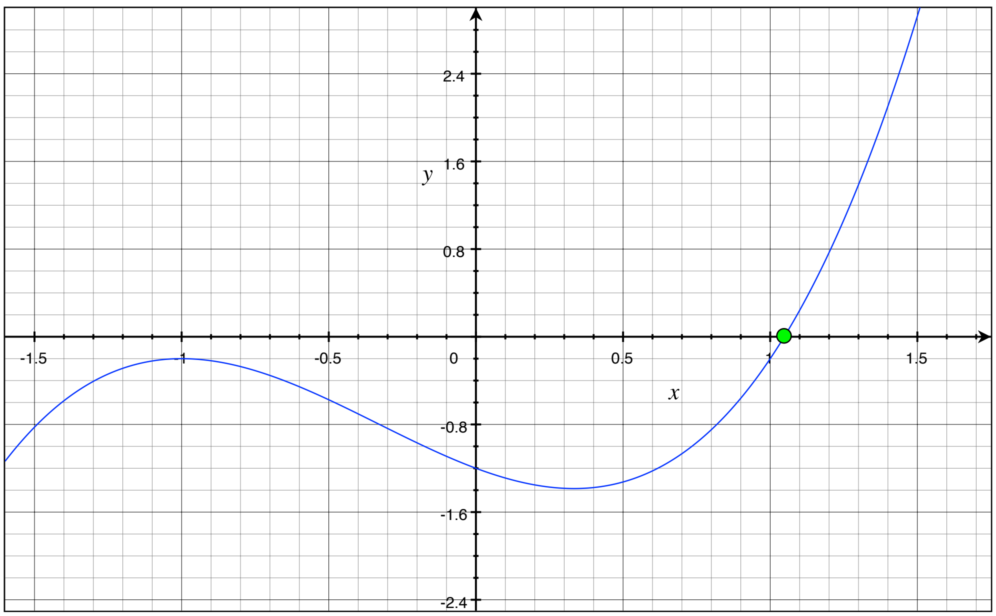
CSC 150 **Program #1** Fall 2014

Due: Oct. 10, 2014

1 Assignment Overview:

Your first program will find a root of a function in some prescribed interval. For example, below we have the graph of $y = x^3 + x^2 - x - 1.2$. The root is approximately at 1.0477. This program will request from the user an interval and then attempts to find the root of the function in that interval.



2 Background:

Recall that a root (or a zero) of a function is a value for which the function evaluates to zero. For example, $x = -2$ is a root or zero for $y = x^2 - 4$. In Algebra courses you learn methods to find these roots. The most common method is factoring. However, factoring methods only work on polynomials, and only on a subset of those (try factoring the example equation in the graph above). In engineering, there are many types of functions that arise which are not polynomials such as equations containing exponentials and trigonometric functions. So, how do we find roots for these more complicated functions?

In Calculus, a solution to this problem is introduced: Newton's Method. Newtons's method uses the function and it's derivative to locate roots. Newton's idea is rather simple. If a function is a straight line, we can solve for the root algebraically. Starting from an initial guess, one computes the function value and the slope. A straight line approximation to the function can be constructed. Using the approximation, one can find a root. This is an approximation of the real root and so you just repeat the process. Mathematically, this is stated as:

Set the initial guess to be x_0
while $|f(x_n)| > err$ **do**
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \text{ for } n = 0, 1, 2, 3..$$
end while

3 Assignment

Write a program to implement Newton's Method.

3.1 Overview

Your program will compute the root of $f(x) = x^3 + x - 0.5\cos(x)$ on the interval $a \le x \le b$.

3.2 Functional Requirements

1. Input
   (a) Your program will display a brief, descriptive message when it first starts, indicating the purpose of the program and the user-required input and the output that will be provided.
   (b) The program will prompt the user for the left and right endpoints: a and b.
   (c) The program will prompt for the error tolerance and the maximum number of iterations.
2. Check Inputs
   (a) If $b \le a$ the program will exit will error message `b should be greater than a`.
   (b) If the user enters a zero or negative error tolerance or iterations, the program will exit with message `positive error and iterations required`.
   (c) The program will check for a root by checking for a sign change: $f(a) * f(b) < 0$.
   (d) If there is not a sign change the program will print: `No sign change, root may not exist.`
3. Algorithm
   (a) The program will implement Newton's Method using $(a + b)/2$ as the initial guess.
   (b) The program will print each iteration as it proceeds.
4. Output: the program will terminate with the root (to the tolerance provided) or with `maximum iterations`.

4 Program Notes

- The end points and error tolerance will be decimal numbers.
- Results should be correct to four decimal places.
- Before you start solving the problem, experiment with pow, sqrt, and tan functions to get a feel for how they works. That is, understand the tools before you begin trying to solve the problem.
- Begin problem solving by using paper and pencil (and calculator) to solve the problem. You need to figure out how to work out a solution by hand before trying to develop a C++ program.
- Put in output statements to print out the input values. Run the program and test it.
- Continue in this manner to add functionality to your program, running and testing it at each step. Run the program and test it.
- $f'(x) = 3x^2 + 1 + 0.5\sin(x)$

5 Comments

- Do not delay. Start writing the program early. If you wait until the night before the due date, you will have a miserable night.
- Do not try and write the entire program all at one time. Work on the program in small sections, as the timeline indicates.

- Debugging Hint: Make sure your programs output matches the sample run. Then make sure values in between the demo values give results that make sense.
- You must use DoxyGen to document your code and format it according to the CSC150 coding standard. Be sure your code file is readable and neat. Do not allow lines to extend past 80 characters, use appropriate white space and make sure to use a consistent and attractive indentation scheme. The detailed coding standard is described in http://www.mcs.sdsmt.edu/csc150/Course/Coding_Style.pdf
- Name your code file prog1.cpp. Points will be deducted if this naming convention is not followed.
- Your program must correctly compile in Visual C++ 2012.

6 Time line

- 9/19/14 - Program Assigned
- 9/22/14 - Work out Newton's Method by hand for a simple example.
- 9/24/14 - Write a program that performs the input operation.
- 9/26/14 - Extend your program to verify inputs.
- 9/29/14 - Add a single Newton's Method step (figure out derivative).
- 10/1/14 - Add the loop to repeat the Newton Step.
- 10/3/14 - Verify correct loop termination.
- 10/6/14 - Test program completely and document.
- 10/8/14 - Submit (not putting this off until the last day).
- 10/10/14 - Program due.

7 Program Submission:

Submit your program code (the **prog1.cpp** file only) at `http://www.mcs.sdsmt.edu/submit` before midnight of the due date. (Your file gets time stamped, so late submissions will be noted and may be given a late penalty!) Be sure to submit to the correct lecture section!

DO YOUR OWN WORK.

8 Samples

Normal run:

```
This program computes the roots of f(x) = x^3 + x - 0.5cos(x).

Please enter the interval to search for the root.
 a  b:  0 1
Please enter the error tolerance: .001
Please enter the maximum number of iterations for the search: 100
n = 1   x = 0.406414
n = 2   x = 0.397986
n = 3   x = 0.397925
```

Error example:

```
This program computes the roots of f(x) = x^3 + x - 0.5cos(x).

Please enter the interval to search for the root.
 a  b:  1 0
b should be greater than a
```

Error example:

```
This program computes the roots of f(x) = x^3 + x - 0.5cos(x).

Please enter the interval to search for the root.
 a  b:  0 1
Please enter the error tolerance: -1
Please enter the maximum number of iterations for the search: 100
Positive error and iterations required
```

Example:

```
This program computes the roots of f(x) = x^3 +x - 0.5cos(x).

Please enter the interval to search for the root.
 a  b:  10 20
Please enter the error tolerance: 0.00001
Please enter the maximum number of iterations for the search: 250
No sign change, root may not exist.

n = 1   x = 9.98706
n = 2   x = 6.63147
n = 3   x = 4.39413
n = 4   x = 2.86473
n = 5   x = 1.82206
n = 6   x = 1.1234
n = 7   x = 0.679464
n = 8   x = 0.455621
n = 9   x = 0.400699
n = 10   x = 0.397931
n = 11   x = 0.397925
n = 12   x = 0.397925
```

Need more test samples? Use the demo program on the course website to verify your solution.

1. Download http://www.mcs.sdsmt.edu/csc150/Assignments/Programs/Program1/prog1.exe to your computer.
2. Go to the parent directory of the directory that contains the downloaded prog1.exe file.
3. Hold Shift key and right click the icon of the directory with prog1.exe, then select Open command window here from the pop-up menu.
4. Type prog1.exe in the console and hit Enter key to execute the demo program.