Tufts University
Department of Mathematics
Math 50 Homework 2

Due: Thursday, September 29, at 1:30 p.m. (in class).

1. In class, we studied a few methods for finding the zeros of $f(x) = F'(x)$, in order to find the extrema of $F(x)$. In this exercise, you will implement both Newton's Method and the Secant Method, as well as three other approaches (described below), and compare their performance on solving the example used in class. For parts (a) through (e), submit a print-out of your code.

   (a) (12 points) Implement Newton's Method. Your function should take two function handles as inputs, to specify $f(x)$ and $f'(x)$, as well as an initial guess, x_0. As outputs, it should give the computed zero, x_k, as well as the iterate number, k.

   To implement this, you want to use a while-loop. The stopping condition for the while loop is a little tricky, since the round-off error associated with evaluating $f(x)$ may prevent your function from being exactly *equal* to zero, even though it may be *essentially* zero. Matlab provides the `eps` constant to deal with this; instead of using the condition `f(x) == 0` in your while statement, use `abs(f(x)) <= 100*eps`. You may also want to set a maximum number of iterates to generate, to guard against an infinite loop.

   (b) (12 points) Implement the Secant Method. Your function should now take a single function handle as input, to specify $f(x)$, as well as two initial guesses, x_0 and x_1. As outputs, it should give the computed zero, x_k, as well as the iterate number, k.

   (c) (12 points) The *Bisection Method* is another approach for finding a zero of $f(x)$ without computing its derivative.

   As inputs, the Bisection Method takes two values, x_L and x_R with $x_L < x_R$, such that the signs of $f(x_L)$ and $f(x_R)$ are different (and so that, for a continuous function $f(x)$, there must be a zero of $f(x)$ between x_L and x_R).

   At each step of the Bisection Method, the value of $f\left(\frac{x_L+x_R}{2}\right)$ is checked. If it is zero, then the iteration stops. If it has the same sign as $f(x_L)$, then x_L is replaced by $\frac{x_L+x_R}{2}$; otherwise, it has the same sign as $f(x_R)$, and x_R is replaced by $\frac{x_L+x_R}{2}$.

   In this way, each step of the Bisection Method replaces one of the endpoints, x_L or x_R, of the interval (x_L, x_R) that contains the zero of $f(x)$. When $f\left(\frac{x_L+x_R}{2}\right)$ is sufficiently small, the value $\frac{x_L+x_R}{2}$ is returned.

   Implement the Bisection Method. Your function should now take a single function handle as input, to specify $f(x)$, as well as the endpoints of the initial interval, x_L and x_R. As outputs, it should give the computed zero, x_k, as well as the iterate number, k.

   (d) (12 points) The *Method of False Position* is an approach that combines aspects of the Secant and Bisection Methods. It again begins with two values, x_L and x_R with $x_L < x_R$, such that the signs of $f(x_L)$ and $f(x_R)$ are different.

   At each step of the Method of False Position, the zero, x_S, of the secant line between x_L and x_R is computed and the value of $f(x_S)$ is found. If it is zero, then the iteration stops. If it has the same sign as $f(x_L)$, then x_L is replaced by x_S; otherwise, it has the same sign as $f(x_R)$, and x_R is replaced by x_S.

   An important modification to the Method of False Position is to adjust the slope of the secant line if the method starts to update only on one side. To do this, keep track of

which side was last updated. If the method tries to update the same side twice in a row, replace the function value on the other side by half its value. This changes the slope (and, thus, the intercept) of the secant line and forces the method to make an update the other endpoint. This is called the Illinois Algorithm.

Implement the modified Method of False Position. The inputs and outputs should match those of the Bisection Method.

(e) (12 points) *Halley's Method* (named after Edmond Halley, after whom is also named Halley's Comet), is an extension of Newton's method. It is derived by applying Newton's Method to the function $g(x) = f(x)/\sqrt{f'(x)}$, and gives the update formula

$$x_{k+1} = x_k - \frac{2f(x_k)f'(x_k)}{2\left(f'(x_k)\right)^2 - f(x_k)f''(x_k)}.$$

Further extensions are possible, giving better convergence but requiring more derivatives (but you don't need to worry about these here).

Implement Halley's Method. Your function should take three function handles as inputs, to specify $f(x)$, $f'(x)$, and $f''(x)$, as well as an initial guess, x_0. As outputs, it should give the computed zero, x_k, as well as the iterate number, k.

(f) (40 points) Apply all of the above methods to finding the extrema of the function considered in class,

$$F(x) = 1 - q^x + 1/x,$$

for values of q including $0.95, 0.99, 0.999$, and 0.9999. Investigate how the convergence of each varies as you vary the initial guess(es). Can you come up with a good heuristic for choosing the initial guess(es) as a function of q? Can you find initial guesses so that the unmodified Method of False Position fails to converge within many steps, but the modified version (the Illinois Algorithm) converges quickly?