Math 4500/6500 Homework #3

This homework assignment covers our notes on solving equations. You are welcome to look at the code from the *Mathematica* notebooks, but when the problems say "write a piece of code to" they mean "write your own code from scratch", not "modify the code in the notebook" or "find a piece of code on the web".

1. Suppose that $c_0, c_1, \ldots, c_n$ are the approximations produced by the bisection method to the solution r of the equation $f(x) = 0$ on an interval $[a_0, b_0]$. Prove that

$$|r - c_n| \leq \frac{b_0 - a_0}{2^{n+1}}.$$

2. Use *Mathematica*'s built-in `FindRoot` function to find a *nonzero* solution to the equation

$$6(e^x - x) = 6 + 3x^2 + 2x^3$$

3. Write your own code to implement the bisection method in *Mathematica*. Use it to find a numerical solution of the equation

$$6(e^x - x) = 6 + 3x^2 + 2x^3.$$

   with $x \in [2.0, 3.0]$ with absolute error less than 10^{-8}. Compare your result to the result of the last question.

4. (Challenge) Devise a modified bisection algorithm which guarantees that the result has *relative* error less than ϵ. Implement your method in *Mathematica*.

5. Solve the polynomial equation

$$x^8 - 36x^7 + 546x^6 - 4\,536x^5 + 22\,449x^4 - \\ 67\,284x^3 + 118\,124x^2 - 109\,584x + 40\,320 = 0.$$

   using *Mathematica*'s `NSolve` procedure. Plot the resulting zeros in the complex plane. Now use *Manipulate* to study the effect of changing the coefficient -36 by ± 0.1. Explain why the results show that the roots of a polynomial are numerically *unstable* functions of the coefficients.

6. Write your own code to implement the false position method in *Mathematica*. Make a table comparing the number of correct digits of approximate solutions to $\tan x + \tanh x = 0$ obtained by your implementations of the Bisection Method and the False Position method from several starting points. Which method performs better?

7. Write your own code to implement Newton's method. (Challenge) Use *Mathematica* itself to take the derivative of an input function $f(x)$. This will require some noodling around

with documentation in order to get it to work as you expect it to. Test your code against False Position and Bisection at finding solutions of the equation $x^2 - 22x + 3 = 0$.

8. The **Steffensen method** is sometimes used in place of Newton's method when the derivative of $f(x)$ is known to exist, but it cannot be easily calculated. The Steffensen method has the iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)} \quad \text{where } g(x) = \frac{f(x + f(x)) - f(x)}{f(x)}.$$

   (1) Implement the Steffensen method in *Mathematica*.
   (2) Find solutions of the equation $f(x) = (x-1)^8$ using both Newton's method and the Steffensen method.
   (3) (Challenge) Prove that when $f(x)$ is small, $g(x)$ is close to $f'(x)$ using Taylor series.
   (4) (Challenge) Prove that the Steffensen method is quadratically convergent.

9. (Challenge) Prove that for any $R > 0$ there is an interval of x_0 values around $1/R$ so that the limit of the iteration

$$x_{n+1} = x_n(2 - x_n R)$$

is $1/R$. This iteration was actually used to implement the division operation on some old computers which had circuitry for multiplication but not division!