MATH-UA 252 - Spring '23 - Numerical Analysis - HW #1

Problem 1. A fixed point iteration is an iteration of the form:

$$x_{n+1} = f(x_n), \tag{1}$$

where $f : \mathbb{R} \to \mathbb{R}$. A function f is said to be Lipschitz continuous with constant $L \geq 0$ on the interval $[a, b] \subseteq \mathbb{R}$ if, for every $x, y \in [a, b]$, we have:

$$|f(x) - f(y)| \leq L|x - y|. \tag{2}$$

Assume that f has the fixed point $x^* \in [a, b]$.

1. Prove that if $x_0 \in [a, b]$, if $f([a, b]) \subseteq [a, b]$, and f is Lipschitz with $L < 1$ on $[a, b]$ then the fixed point iteration (1) converges to x.
2. Prove that the relative error at the nth iteration satisfies:

   $$|x_n - x^*| \leq L^n |x_0 - x^*|. \tag{3}$$

   What is the order and rate of convergence of the sequence $\{x_n - x^*\}$?
3. Show that the number of iterations needed for a relative error smaller than $\epsilon > 0$ is at least $\log_L(\epsilon)$.
4. The *number of digits of relative precision* p for a relative error tolerance $\epsilon > 0$ is:

   $$p(\epsilon) = \log_{10}(1/\epsilon). \tag{4}$$

   Show that $\log_L(\epsilon) \propto p(\epsilon)$ (this means that there is some constant C such that $\log_L(\epsilon) = Cp(\epsilon)$). What is the constant of proportionality C?
5. What a 10×10 table with rows corresponding to values of L and columns corresponding to values of $p(\epsilon)$, and with table values given by the number of iterations required to achieve the given relative precision for that particular Lipschitz constant L. Try to choose values of p and L which are the most interesting and informative.
6. Summarize your results and write any observations about the fixed point iteration that you have made based on the previous parts of this exercise.
7. The Babylonian algorithm is a fixed point iteration. How does its order of convergence relate to what you have found in this problem? Are there any discrepancies? How do you account for them?

Problem 2. Another algorithm for solving rootfinding problems is called *bisection*. Assume that f has a single root $x^* \in [a, b]$, and that $f \in C^0([a, b])$ (that is, f is continuous on $[a, b]$). Bisection works by initially setting $x_0 = a$ and $x_1 = b$. Let $a_n = \min(x_n, x_{n+1})$ and $b_n = \max(x_n, x_{n+1})$. For each $n \geq 0$, we want to maintain the invariant:

$$a_n \leq x^* \leq b_n \tag{5}$$

Note that this implies that if $x^* \neq x_n$ and $x^* \neq x_{n+1}$, that either $f(x_n) < 0$ and $f(x_{n+1}) > 0$, or $f(x_n) > 0$ and $f(x_{n+1}) < 0$.

1. Complete this idea by writing down an algorithm which repeatedly computes the midpoint of x_n and x_{n+1}, giving the *bisection* iteration. (It may help to draw a picture.)
2. What is the rate and order of convergence of the sequence $\{x_n - x^*\}$?

Problem 3. Yet another algorithm for solving rootfinding problems is the Secant method, where after choosing x_0 and x_1, we iterate:

$$x_{n+2} = x_{n+1} - \frac{x_{n+1} - x_n}{f(x_{n+1} - x_n)} f(x_{n+1}), \qquad n \geq 0. \tag{6}$$

1. This algorithm can be derived by repeatedly finding where the line passing through $(x_n, f(x_n))$ and $(x_{n+1}, f(x_{n+1})$ passes through the x-axis. Prove this.
2. Draw a picture illustrating this iteration applied to the problem of computing $x = \sqrt{y}$ for $y = 2$.

Problem 4. So far we've seen four different methods which can be used to solve 1D nonlinear equations: the fixed point iteration, the secant method, Newton's method, and bisection. The problem of computing $x = \sqrt{y}$ is equivalent to solving a 1D nonlinear equation. We saw in class that applying Newton's method to its solution results in the Babylonian algorithm, which is a fixed point iteration.

1. Derive the secant method iteration for computing $x = \sqrt{y}$.
2. Write Python code to compute $x = \sqrt{y}$ using the Babylonian algorithm, the secant method, and bisection.
3. Compare each of your three iterations by computing the square root of five different numbers of your choice. Collect the following data:
   (a) The number of floating-point operations used to compute the square root by the entire iteration until convergence.
   (b) The relative error $|x_n - x^*|/|x_0 - x^*|$.

   You will have to decide the best way to judge when the iteration has "converged".
4. Compare and contrast the different methods. Do they always work? Do they work better or worse for different regions?