Numerical Methods for Digital Computers (Fall 2020) Homework: 2

ICSI 401 – Numerical Methods

Instructor: Abram Magner Date: Fall 2020

Instructions: Please answer the following questions in complete sentences, showing all work (including code and output of programs, when applicable). I prefer that you type your solutions (e.g., using LaTeX, with Overleaf, TeXworks, etc., or Word), but will accept handwritten notes. If the grader cannot read your handwriting, then they cannot award you points.

Certain problems require you to write Matlab code. You should include in your submission a printout of all Matlab code that you write, in addition to the output that you get when you run it.

Due date: Thursday, 10/1/2020, 11:59 p.m., on Blackboard.

2.1 Continuity, differentiability review

- Give an example of a function $f : \mathbb{R} \to \mathbb{R}$ that is continuous everywhere on its domain but has at least one point at which it is not differentiable.
- Give an example of a function $f : \mathbb{R} \to \mathbb{R}$ that is not continuous at 0.

Hint for both of these: You can define piecewise functions (i.e., you can think of a function that has some form on one interval and then a different form on another interval). The purpose of this problem is for you to recall intuitively/graphically what continuity and differentiability mean.

2.2 Floating point and related topics

1. Write down the binary expansion of 50.5.
2. Consider the gaps between representable floating point numbers. Which of the following has the larger gap between it and the next larger representable floating point number?
   - 2
   - 201

   Just a definition and not a problem that you have to solve: we say that x is an accurate approximation to y to k decimal places if $|x - y| < 5 \times 10^{-k-1}$. For example, if we want 4 decimal places of accuracy, then $|x - y| < 5 \times 10^{-5}$. You should use this definition in the problems below.

2.3 Bisection search

1. In Matlab, implement a function that performs a bisection search. It should take the following parameters:

   - `F`: A function (assumed to be continuous) whose roots you want to find,
   - `a`: A floating point number giving the left endpoint of the initial interval in which you want to search for a root of F.
   - `b`: A floating point number giving the right endpoint of the initial interval.
   - `delta`: A non-negative floating point number giving the acceptable proximity of the output to a root of F.

   Your function should first check a and b to determine whether or not they satisfy the condition given by the Intermediate Value Theorem that allows us to conclude that $[a, b]$ contains a root of F. If this condition is not satisfied, return `NaN` (Matlab for "Not a number"). If the condition is satisfied, your function should perform a bisection search until it finds a number z that it can guarantee satisfies $|x - x_*| <$ `delta`, for *some* real-valued root x_* of F. It should return z.

2. Use the Matlab function that you wrote to find a real-valued root of the function $F(x) = x^5 + x + 1$, with accuracy to 4 decimal places (this last requirement will determine your choice of `delta`).

   A fun fact: You know about the quadratic formula, which gives the roots of a quadratic function in terms of its coefficients. It turns out that there exist similar formulas for polynomial functions of degree 3 and 4, but *not* for 5 or more (formally, such equations are not *solvable by radicals*). This is a result by Niels Henrik Abel, and the proof uses what is called *Galois theory* (sometimes covered in classes on abstract algebra). The fact that no such simple formula exists for the roots is a pretty good motivation for numerical methods.

3. Suppose that you use bisection search to find a root of $F(x) = \sin x$, with $a = -\pi/2, b = 5\pi/2$. To which root will the bisection search converge?

2.4 Newton's method

1. Write a Matlab function that implements Newton's method. It should take the following parameters:

   - F: A function whose roots you want to find,
   - `Fprime`: The derivative of F,
   - w_0: An initial point at which F is differentiable,
   - k: A positive integer giving the number of iterations to execute.

   Your function should execute k iterations of Newton's method using the parameter functions F and `Fprime` and output the resulting number, w_k.

2. Suppose that you want to find the points at which the graphs of two functions $f(x)$ and $g(x)$ intersect. Write down a function $H(x)$ such that $f(x)$ and $g(x)$ intersect at a point z if and only if $H(z) = 0$.

3. Find the point of intersection of $f(x) = x$ and $g(x) = x \log x$:

   - Write down a function $H(x)$ as above, such that the roots of $H(x)$ are exactly the points of intersection of x and $x \log x$.
   - Write down the Newton iteration equation (i.e., w_{k+1} in terms of w_k, $H(x)$, and $H'(x)$) that you would use to find roots of $H(x)$.
   - Use your Matlab implementation of Newton's method with $k = 50$ and some appropriate starting point w_0 to find an approximation w_k to the root x_* of $H(x)$. In order to choose w_0, you can plot $H(x)$ and identify a point visually close to x_*.