GEB 6895: Business Intelligence
Department of Economics
College of Business Administration
University of Central Florida
Fall 2019

Assignment 7

Due Tuesday, November 5, 2019 at 11:59 PM
in *your* private mirror of the GEB6895F19 GitHub repo.

Instructions:

Complete this assignment within the space on your private mirror of the GEB6895F19 GitHub repo in the folder `assignment_07`. Create a folder called `my_answers` that will contain all of your work for this assignment, which can be in the form of a `docx` file, a `README.md` file or simply comments within in the relevant scripts.

Question 1:

In Assignment 4, we used the `uniroot()` function to find the root x^* of the function $f(x)$ such that $f(x^*) = 0$. One popular algorithm for doing so is called Newton's method or the Newton-Raphson method. This algorithm proceeds by making successive changes to a candidate root x_i by solving for the first-order Taylor polynomial to $f(x)$ at the point x_i. Specifically, the iterations take the form

$$x_{i+1} = x_i - \frac{f(x_i)}{f\prime(x_i)} \tag{1}$$

and continue recursively until $f(x_i) < \epsilon$, for some small tolerance value ϵ.

a) Create a function that solves for the root of $f(x) = \log(x) - e^{-x}$ using Newton's method. Your function should take the form `newton_a(x_0, tol_f)` and return the root of the function x^* such that $f(x^*) < \epsilon$, where ϵ is represented by the argument `tol_f`.

b) Create a function that solves for a root of $f(x) = x^3 - 7x^2 + 8x - 3$ using Newton's method. Your function should take the form `newton_b(x_0, tol_f)` and return the root of the function x^* such that $f(x^*) < \epsilon$, where ϵ is represented by the argument `tol_f`.

c) Test your functions with a few different starting values `x_0` and tolerance levels `tol_f`.

Question 2:

In this exercise, you will solve a function $f(x)$ for the root x^* such that $f(x^*) = 0$ using an algorithm called the bissection method. Create two functions `bisect_a(x_l, x_u, tol_f)` and `bisect_b(x_l, x_u, tol_f)` that calculate roots of the function $x^2 - 2 = 0$ within an interval $[x\_l, x\_u]$, which is specified by the user through the arguments of the function, up to a degree of accuracy specified by `tol_f`. The function will recursively split the interval in half and choose the new endpoints according to the application of the intermediate value theorem. Specifically, the functions will replace one of the endpoints depending on the sign of $(\frac{x\_u - x\_l}{2})^2 - 2$ and conintue by repeating this iteration until $f(x^*) < \epsilon$, where ϵ is represented by the argument `tol_f`.

a) The function `bisect_a(x_l, x_u, tol_f)` should use a `while` loop to continue iterating until $x^2 - 2 = 0$ is less than `tol_f`.

b) The function `bisect_b(x_l, x_u, tol_f)` should progress according to the algorithmic strategy of recursion, such as was used for the factorial function in Paarsch & Golyaev on pages 165-166.

c) Test and compare your functions with a few different starting intarvals $[x\_l, x\_u]$ and tolerance levels `tol_f`.