Worksheet 2: Root finding

If you are using Julia or Python, we recommend using a jupyter notebook. In WeLearn, you need to submit this file. Please clearly indicate in the markup cells, the number of the question for which you are writing the program. Also, please remember to add documentation through comments in your program.

You may also use scripts and use REPL to evaluate them. In that case, please keep all your files for a particular worksheet in a folder and you may upload the compressed archive of that folder.

Please feel free to ask for help!

1. (3 points) Consider a function $f_1(x) = (x+3)(x-1)^2$. This function has roots at -3 and $+1$. You need to find the root at -3 numerically with an accuracy of 10^{-6} using bisection algorithm. You may choose a bracketing interval $[-4, -2.5]$. Your program, should print the iteration number, the computed value of the guess (c) and the difference between the actual root and the computed root at every step. Write the program with the usual safety checks, such as,

   1. if the bracketing interval doesn't contain the root, then the program should report that and stop,
   2. if the number of iterations exceed a certain pre-set value (say, `maxiter = 100`) then it should stop and mention that it couldn't converge to the solution in `maxiter` steps.

   The program must define a function (or subroutine) `bisection` which returns the final number of iterations and the final computed root. So the main body of the program should define the tolerance (say, `tol`), the bracketing interval (`l` and `r`), the functions (f_1 and *the bisection*), the maximum number of iteration (`maxiter`) as parameters and then call the function bisection (with arguments f_1, `l, r, tol`) to arrive at the root. Let `maxiter` be a global parameter (no need to pass it to *bisection* as an argument. Finally, your program should report the numerical root,

   To check whether your safety checks are working try finding the other root of the function f_1 using *bisection*.

2. (3 points) Modify the above program to find the root of a function $f_2(x) = x - \frac{1}{3}\tan x$ within a bracketing interval $[1.2, 1.4]$. Other parameters (tolerance etc) remain unchanged. There is no need to print the difference this time.

3. (14 points) Consider a particle moving in an asymmetric one-dimensional double-well potential $V(x) = x^4 + \left(\frac{2x}{3}\right)^3 - x^2$ eV with a total energy -0.125 eV. You need to find the turning points for the particle when it is in either potential well. Your result should be accurate (in terms of position, not the value of the energy) upto 5th decimal place. [Hint: plot and choose appropriate initial conditions.] Use the following methods.

   (a) (3 points) Bisection

(b) (3 points) Regula Falsi

(c) (4 points) Secant

(d) (4 points) Newton Raphson

Page 2