# ACM40290: NUMERICAL ALGORITHMS

## **Assignment 3:** Root finding

Although Matlab has built-in functions for solving $n$-dimensional non-linear equations, it is important that you program some of the methods discussed in class so that you appreciate how difficult it is to write (and use) good software. This appreciation will, it is hoped, make you a critical user of other peoples software.

## EXERCISE

Write Matlab functions for finding the roots of a general function using each of the following methods:

(1) Bisection Method: standard bisection method
(2) Newton Method: second order Newton method
(3) Secant Method: standard (two-point) secant method
(4) ModFzero: modified version of Matlab's `fzero`. This is a function that is given the arguments specified below and calls Matlab's `fzero` with the appropriate arguments.

Each of the methods should be written as an independent m-file function with file names: `Bisect.m`, `Newton.m`, `Secant.m`, `ModFzero.m`. The functions should have the following calling conventions:

(1) `[root flag iters] = Bisect(X1, X2, Xtol, Maxits)`
(2) `[root flag iters] = Newton(X1, Xtol, Maxits)`
(3) `[root flag iters] = Secant(X1, X2, Xtol, Maxits)`
(4) `[root flag iters] = ModFzero(X1, X2, Xtol, Maxits)`

Here, `X1` and X2 are initial guesses, `Xtol` is the desired tolerance in the result, and `Maxits` is the maximum number of iterations the routine should try before returning. The functions should return three items:

(1) `root`: The value of the root that was found
(2) `flag`: A flag indicating the result of attempting to find the root. Possible flags are:
    (a) `flag=0`: $x$-sequence convergence satisfied.
    (b) `flag=1`: $f$-value is zero.
    (c) `flag=-1`: `Maxits` reached without $x$ or $f$ convergence.
    (d) `flag=-2`: Cannot proceed.
(3) `iters`: The number of iterations.

Your MATLAB programs for these methods should be able to run without error when called by the MATLAB test program `DrEqn.m`. This tests your root finding implemenations with the following functions:

$$\begin{aligned}
f_1(x) &= x - \cos(x) \\
f_2(x) &= e^{x+1.00202} - e^1 \\
f_3(x) &= \frac{\log_2 x}{x-1} - \frac{1}{\ln 2} \\
f_4(x) &= x^20 - 1.0 \\
f_5(x) &= -1.4 + x^2 + \frac{\ln(|1 + 3(1-x)|)}{80} \\
f_6(x) &= (y-4)(y+2)(y+41) \\
&\qquad \text{where} \quad y = (X - 1.01 \times 10^{-9}) \times 10^8 \\
f_7(x) &= (x^2 - 2.2x + 1.21) \times \frac{x - 1.1}{|x - 1.1|} \\
f_8(x) &= |x - 9.1|^{4.5} \\
f_9(x) &= |x - 8.4317|^0.4 \\
f_{10}(x) &= \begin{cases} 0, & |x| < 3.8 \times 10^{-4} \\ xe^{-1/x^2}, & |x| \geq 3.8 \times 10^{-4} \end{cases} \\
f_{11}(x) &= \begin{cases} e^x, & x > -1 \times 10^6 \\ e^{-1\times 10^6} - (x + 1 + 10^6)^2, & x \leq -1 \times 10^6 \end{cases} \\
f_{12}(x) &= \tan^{-1} x.
\end{aligned}$$

These functions and their derivatives are defined in the files `FUN.m` and `DFUN.m`, which can be download from the assignment page on Blackboard. The definitions for the derivatives of functions 3, 5, and 12 are missing. **NB:** You must derive the derivative functions for these yourself and add the definitions inside `DFUN.m`.

All answers should be submitted via Blackboard and should include MATLAB code for each of the four functions.

**Due: Midnight on Monday, October 30th 2017**