COMPUTATIONAL ALGORITHMS FOR FINDING ROOTS OF AN EQUATION

I. Bisection Method

1. Define the function $F(X)$

2. Find 2 values of x that bisect the root i.e., Guess x_l and x_u such that $F(x_l) * F(x_u) < 0$

3. $x_r = \frac{1}{2}(x_l + x_u)$

4. Determine whether root lies between x_l and x_r or x_r and x_u
   i.e., if $(F(x_l) * F(x_r) < 0)$
   then $x_u = x_r$ else if $(F(x_r) * F(x_u) < 0)$
   then $x_l = x_r$

5. Check for convergence: if ($|f(x_r)|$ < tolerance) converged
   else return to 3.

II. False-Position Method

Same as Bisection Method except for Step 3.

3. $$x_r = x_i - \frac{F(x_l)(x_u - x_l)}{F(x_u) - F(x_l)}$$

III. One-Point Iteration

1. From the function $F(x) = 0$, find the iteration equation $x = G(x)$, such that the derivative of $G(x), |G(x_o)| < 1$ for the 1st guess x_o. Then define the function $G(x)$ at the 1st guess x_o.

2. Calculate a better guess $x_i = G(x_{i-1})$

3. Check of convergence: if ($|F(x_i)|$ < tolerance) converged
   else, increment *i* and return to 2.

IV. Newton-Raphson Iteration

1. Define the function $F(x)$ and its derivative $F'(x)$, and make the first guess of the root x_o.

2. Calculate a better guess: $x_i = x_{i-1} - \frac{F(x_{i-1})}{F(x_{i-1})}$

3. Check the convergence: If ($|F(x_i)|$ < tolerance) converged
   else, increment *i* and return to 2.

V. Secant Method

Same as Newton Raphson method, except that in:

1. No need to define derivative $F1(x)$, but 2 guesses of the root are required, x_{-1} and x_o.

2. Calculate a better guess: $x_i = x_{i-1} - \frac{F(x_{i-1})(x_{i-1} - x_{i-2})}{F(x_{i-1}) - F(x_{i-2})}$