INSTRUCTOR: RUIWEN SHU

Problem 1: Write codes for bisection method, Newton's method, Secant method, Steffensen's method for solving one nonlinear equation. Apply them to find the unique root of

$$f(x) = e^{-x} - x$$

with properly chosen initial values and error tolerance 10^{-10}. For each method, plot points (n, e_n) to illustrate its convergence rate (the error e_n can be taken as $f(x_n)$; do the plots for all methods in one picture to compare them).

Problem 2: Write codes for 2-dimensional Newton's method. Use it to find a solution of

$$\begin{cases} 2x + y + 0.3\cos x = 0 \\ x - y + 0.1\sin(2y) = 0 \end{cases}$$

with error tolerance 10^{-10}.

Problem 3: Write codes for Newton's method for complex polynomials. Consider

$$p(z) = z^3 - z - 1$$

For every initial point $z_0 = (0.02j, 0.02k)$, $j, k = -100, \ldots, 100$, apply the Newton's method and determine which zero the iteration converges to, or it diverges. Plot these initial points in the complex plane with different colors according to their convergence results.

Problem 4: Repeat Problem 3 with Laguerre's method.

Problem 5: Write codes for Horner's algorithm. Write codes to find all complex roots of a given polynomial $p(z)$ in the following way: each time you try a random initial value to run Laguerre's method until you find a zero (up to certain tolerance); then factor it out by Horner's algorithm and repeat it for the quotient polynomial. For the roots $r_1, \ldots, r_n$ you find, calculate $p(r_1), \ldots, p(r_n)$ to see whether they are below the prescribed tolerance level.
Test your codes for the following two examples:
1. A degree-20 polynomial with coefficients being randomly ± 1;
2. the same as the previous example with the constant coefficient multiplied by 10^7.