Homework 2 Introduction to Computational Finance Spring 2023

Solutions due Friday, 2/24/23

Answers to the homework problems should be submitted using the class canvas page.

You should submit pdf files. **Do not sent Word files or any other text processing tool's input file**.

As with all homework assignments you are allowed and encouraged to consult the relevant literature. You are also expected **to cite all literature that is used to generate your solutions and your solutions must make clear your understanding of the work cited.**

Programming Assignment

Problem 2.1

Programming tasks

Your main programming task is to implement a nonlinear solver class with appropriate methods. The class should be able to solve using single and double precision variables.

The first method uses the regula falsi method to solve a nonlinear equation. It should have an interface that looks something like

RegFal(f,a,b,absTol,relTol)

where f is the function whose root is to be found, a and b are the left and right endpoints respectively of the interval containing a root, and $absTol$ and $relTol$ are the desired absolute and relative errors in the result.

The second method uses Newton's method to solve a nonlinear equation. It should have an interface that looks something like

Newton(f,fp,startValue,absTol,relTol)

where f and fp are the function and its derivative, $startValue$ is the initial guess at the root.

The third method uses the secant method to solve a nonlinear equation. It should have an interface that looks something like

Secant(f,startValue,absTol,relTol)

where f is the function, $startValue$ is the initial guess at the root.

RegFal, Secant and Newton are to be stand-alone and reusable and should not need to be modified when changing problems to be solved. They should include no I/O and should not assume that the functions f and fp have particular names only that $f = f(x)$ and

$fp = f'(x)$. You should return an error flag to indicate to the user if the iteration has completed successfully or not.

These routines should be tested individually and your report must contain convincing evidence and arguments that the routines work correctly. You should base the test design and analysis on the theory that describes how the methods should behave. **It should include a discussion of the convergence rates and behavior expected by the theory and observed in your experiments. You should use carefully chosen functions to make your points in this discussion. Include examples of the influence of the choice of initial condition on the behavior of the methods (especially for locally convergent Newton and Secant methods). The effect the multiplicity of the root on convergence behavior should also be investigated.**

You should also include in your examples to be analyzed elements from the set of Chebyshev polynomials of the first kind that are defined using the recurrence:

$$T_0(x) = 1, \quad T_1(x) = x, \quad \text{and} \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

where $T_i(x)$ is clearly a polynomial of degree i. For example, we have for the first three polynomials,

$$T_0 = 1, \;\; T_1 = x, \;\; T_2 = 2x^2 - 1, \;\; T_3 = 4x^3 - 3x.$$

It is known that all of the roots of these polynomials are contained in the closed interval $[-1, 1]$. The roots of the degree $m + 1$ Chebyshev polynomial $T_{m+1}(x)$ are known to be

$$x_i = \cos\left(\frac{(2i+1)}{(m+1)}\frac{\pi}{2}\right), \quad 0 \le i \le m.$$

You should apply all the three methods above to several of the Chebyshev polynomials and include the analysis of convergence, accuracy etc. mentioned above. In particular, explore how the choice of initial conditions for Secant and Newton affect leaving the interval with the iterates and what happens. For example, does it return and converge to a particular root? Does it blow up and fail with, say, overflow? For these problems **add damping to Secant and Newton so that the iterates are not allowed to leave the interval** $[-1, 1]$ **(as described in class and the notes).** How well do the damped versions work? Support your conclusions with empirical behavior observations.

Notes on the empirical evaluation of the methods

Your solutions must provide evidence of correct execution using specifically designed experiments that include evidence of the rates of convergence expected from Regula Falsi (linear), Secant (superlinear), and Newton (quadratic).

To estimate the convergence rate observed for problems where the solution is known, i.e., one of your experiments to demonstrate correctness, one can exploit various observations. For example, the simplest is to monitor

$$\frac{|e_k|}{|e_{k-1}|}.$$

There should be a series of k values for which this ratio is close to constant with a constant less than 1. This value should be consistent with the theoretical values deduced from analyzing $|\phi'(x_*)|$ as discussed in the notes. Note, however, that before convergence sets and after convergence has reached roundoff levels, in you may not see a near constant ratio.

Linear versus super and sublinear can also be observed graphically by noting that the assumption $|e_k| = C|e_{k-1}|$ with $C < 1$ yields

$$\begin{aligned} |e_k| &= C|e_{k-1}| = \cdots = C^k|e_0| \\ \rho_k &= \log_\gamma \left\{ \frac{|e_k|}{|e_0|} \right\} = k \log_\gamma C \end{aligned}$$

So ρ_k is a linear function of k with slope $\log_\gamma C < 0$ since $C < 1$. For superlinear convergence, ρ_k will go to $-\infty$ faster than linearly, i.e., the slope will get steeper and steeper negatively. In the case of quadratic convergence we know

$$|e_k| = \tilde{C}|e_{k-1}||e_{k-1}|$$

whose slope goes to ∞ very quickly for a convergent algorithm. A sublinear algorithm like Bisection that may not even be monotonically convergent the curve can look very nonuniform but never consistently faster than linear except for serendipitously favorable situations.

For quadratically convergent methods, there will be an interval once convergence sets in where each iteration should roughly double the number of digits known in the solution. Note that this happens very quickly. Consider for example a case where the initial guess has 4 digits accuracy. The next step will have about 8 and the next 16, i.e., double precision in three steps. Single precision starting out with 2 digits would have a similar profile.

You should look carefully at the shape of the functions and sample the various intervals that the roots and critical points define. Global convergence vs local convergence should be examined. The polynomial $f(x) = x^3 - 4x$ has roots at $x_\pm = \pm 2$ and $x_* = 0$ and we have

$$f(x) = \begin{cases} \leq 0 & x \leq -2 \\ \geq 0 & -2 \geq x \leq 0 \\ \leq 0 & 0 \geq x \leq 2 \\ \geq 0 & x \geq 2 \end{cases}$$

and

$$f'(x) = \begin{cases} = 0 & x = \pm\sqrt{\frac{4}{3}} \\ > 0 & x < -\sqrt{\frac{4}{3}} \\ < 0 & -\sqrt{\frac{4}{3}} < x < \sqrt{\frac{4}{3}} \\ > 0 & x > \sqrt{\frac{4}{3}}. \end{cases}$$

The function is antisymmetric around 0. So you can construct initial intervals of Regula Falsi or that contain x_0 (and x_{-1}) for Newton (and Secant) based on, e.g. $\alpha > 0$ and $\beta > 0$, with various values and sign patterns. For example, $a = -\alpha$ and $b = \alpha$ or $a = -\alpha$ and $b = -\beta$ or $a = \alpha$ and $b = \beta$ with the specific values chosen based on the shape of the function. The effects of their values and the location of the initial guesses should be related to convergence behavior and rates that you observe.

You should also apply your methods to the test problems from the notes and other simple functions for which you can use the shape to predict where the choice of the initial guesses for the roots must be in order to observe convergence to a particular root, divergence from all roots (if possible), very slow convergence etc. Essentially, these experiments should be part of the overall plan you use to show that the codes work as expected and that you have developed some characteristics of the shapes expected to influence the methods behaviors.

Written Exercises

Problem 2.2

Suppose to solve for a root of $f(x)$, i.e., $f(x^*) = 0$, we use the iteration

$$x_{k+1} = \phi(x_k)$$

where $\phi(x)$ and $f(x)$ given functions with as many continuous derivatives as you require.

Show that if $|\phi'(x^*)| < 1$ then for $|x_0 - x^*|$ sufficiently small the iteration $x_{k+1} = \phi(x_k)$ produces a sequence that converges to x^*.

Note that the iteration $\phi(x)$ is called a contraction mapping in the neighborhood of x^* when $|\phi'(x^*)| < 1$.

Problem 2.3

Let $f(x) = x^3 - 3x + 1$. This polynomial has three distinct roots. Consider using the iteration function

$$\phi(x) = \frac{1}{3}(x^3 + 1)$$

Which, if any, of the three roots can you compute with $\phi(x)$ and how would you choose $x^{(0)}$ for each computable root?

Problem 2.4

2.4.a

Suppose you have the Lagrange form of the unique interpolating polynomial of degree n through points (x_i, f_i)

$$p_n(x) = \sum_{i=0}^{n} f_i \ell_i(x)$$

Is it true or false that

$$\sum_{i=0}^{n} \ell_i'(x) = 1$$

Justify your answer.

2.4.b

Given the data points

$$(x_0, f(x_0)) = (1, 7), \quad (x_1, f(x_1)) = (3, 59)$$
$$(x_2, f(x_2)) = (4, 124), \quad (x_3, f(x_3)) = (6, 402)$$

find the unique cubic interpolating polynomial, $p_3(x)$ and evaluate it at $x = 0$ and $x = 10$.

Problem 2.5

Use this divided difference table for this problem.

i	0		1		2		3		4		5
x_i	-1		0		2		4		5		6
f_i	13		2		-14		18		67		91
$f[-,-]$		-11		-8		16		49		24	
$f[-,-,-]$			1		6		11		$-25/2$		
$f[-,-,-,-]$				1		1		$-47/8$			
$f[-,-,-,-,-]$					0		$-55/48$				
$f[-,-,-,-,-,-]$						$-55/336$					

2.5.a

Use the divided difference information about the unknown function $f(x)$ and consider the unique polynomial, denoted $p_{1,5}(x)$, that interpolates the data given by pairs (x_1, f_1), (x_2, f_2), (x_3, f_3), (x_4, f_4) , and (x_5, f_5). Use two different sets of divided differences to express $p_{1,5}(x)$ in two distinct forms.

2.5.b

What is the significance of the value of 0 for $f[x_0, x_1, x_2, x_3, x_4]$?

2.5.c

Denote by $p_{0,4}(x)$, the unique polynomial, that interpolates the data given by pairs (x_0, f_0), (x_1, f_1), (x_2, f_2), (x_3, f_3), and (x_4, f_4) and recall the definition of $p_{1,5}(x)$ from part (a). Use the divided difference information about the unknown function $f(x)$ to derive error estimates for $f(x) - p_{1,5}(x)$ and $f(x) - p_{0,4}(x)$ for any $x_0 \leq x \leq x_5$.