Responsible TA:
Jiwoo Oh
jiwoo.oh@chem.ethz.ch

Statistical and Numerical Methods
Exercise 3, HS 2024

08/10/2024

1 Fixed point iterations (Core)

Consider the following nonlinear equation

$$f(x) = x\text{exp}(x) - 1 = 0 \tag{1}$$

Like in the lecture, we consider the three fixed point equations

$$\begin{aligned} x &= \phi_1(x) \quad \text{with } \phi_1(x) = \text{exp}(-x) \\ x &= \phi_2(x) \quad \text{with } \phi_2(x) = \frac{x^2\text{exp}(x) + 1}{\text{exp}(x)(1 + x)} \\ x &= \phi_3(x) \quad \text{with } \phi_3(x) = x + 1 - x\text{exp}(x) \end{aligned} \tag{2}$$

1. Show analytically that the three fix point equations are consistent with (1).
2. For each of the iterative formulas in(2) try to find a fixed point using an iteration of the form $x^{k+1} = \phi_i(x^k)$ with $i = 1, 2, 3$ and k denoting the k-th iteration (Start with pseudo-codes[1] on paper):
   - Use a starting guess x_0 between 0 and 1
   - Loop `while abs(x`$_k$` - x`$_{k-1}$`) > 1e-8` calculate the next `x` value
   - Store all values that you calculate in a vector `xvec`
   - Also terminate the while-loop if 10^5 iterations are exceeded
3. For each formula, say if the fixed point iteration converges or not. Provide the answers using an `if` block in your code.
4. Compare your results with those, which can be obtained by using fsolve for each of the cases in (1) and (2). Could you improve your results by using a different maximal error (tolerance level)?
5. Estimate the convergence orders p and the rates of convergence C for the formulas which have a fixed point (keep in mind that the following formulas cannot be applied to the first and last elements of your x^k vectors).
   - Define `xstar` based on the last iteration value or the solution from `fsolve`
   - Calculate the vector `eps = abs(xvec - xstar)`
   - Use the results in to determine p and C according to (3) and (4), plot the results (p and C vs k) using `subplot` and interpret them.

$$p = \frac{\log(\epsilon^{(k+1)}) - \log(\epsilon^{(k)})}{\log(\epsilon^{(k)}) - \log(\epsilon^{(k-1)})} \tag{3}$$

$$C = \frac{\epsilon^{(k+1)}}{(\epsilon^{(k)})^p} \tag{4}$$

[1]Please write pseudo-codes in a note and submit separately.

2 Solving Nonlinear Equation (Core)

The steady state heat flux Q of a CSTR for a first order, irreversible reaction is given by

$$\begin{aligned} Q &= \frac{\eta\kappa(\theta)}{1+\kappa(\theta)} + 1 - \theta + K^C(\theta^C - \theta) = 0 \\ \kappa(\theta) &= \kappa_0 \text{exp}\Big(-\frac{\alpha}{\theta}\Big) \end{aligned} \tag{5}$$

1. Plot the total heat flow Q from and to the reactor (5) vs. the dimensionless reactor temperature θ , for θ between 0.9 and 1.25
   - Use α=49.46; κ_0=2.17$\times 10^{20}$; K^C=0.83; η=0.33; θ^C=0.9;
2. Implement and use the secant method in a function to find the three steady state temperatures of the CSTR (Start with pseudo-codes on paper).
   - Your function file header should read something like `function [x, xvec] = secantRoot(f, x0)` where `f` is a function handle to the function that is to be solved, and `x0` is an initial guess.
   - Store and return all x-values calculated in a vector `xvec`.
   - The calculation steps of the secant method read

   $$\begin{aligned} x_{k+1} &= x_k - \frac{f(x_k)}{f'(x_k)} \\ f'(x_k) &\approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \end{aligned}$$

   - The secant method requires two starting points, use $x_1 = (1+\epsilon)x_0$ as a second point. Suggest a value for ϵ (not too small, why?).
   - Loop `while abs(x`$_k$` - x`$_{k-1}$`) > 1e-8` **and** `abs(f(x`$_k$`)) > 1e-6` **and** `n < 1e5`
   - You will have to work with three x-values at any given iteration, that is `x`$_{k+1}$, `x`$_k$ and `x`$_{k-1}$
   - In what range of `x0` can you converge to the intermediate solution? What feature of the function determines which solution is found?
3. Repeat the task of 2. by using the Newton method, i.e. by using the analytical value of $f'(x_k)$ instead of the approximation (Start with pseudo-codes on paper).
4. Use the resulting `xvec` to estimate the convergence order and rate of convergence of the two methods. What do you observe regarding the algorithmic performance of each method? (**Hint:** you can use the built-in keywords `tic` and `toc`)