Project 1, MAT 362
Prof. Swift, Fall 2019
Due Wednesday, October 9, at 11:59 pm.

Version 2: I added the italics phrase in the first bullet, changed p_n to p_{n+1} in the third bullet, and put an added note at the end.

This project concerns the root-finding methods we have studied in this class. We also investigate the order of convergence of the various root finding methods.

- Approximate the solution to $\cos(x) = x^3$, or equivalently $f(x) = \cos(x) - x^3 = 0$. Start by graphing both $y = \cos(x)$ and $y = x^3$ on the same axes, and convince the reader that there is a unique solution. Then approximate the zero of f in five ways: Use bisection method, fixed point method with $g(x) = x - f(x)/C$ for $C = -3$ and $C = -2$, Newton's method, and secant method.

  In the fixed point iteration, show the graph of f and argue that $C = -3$ is the better choice, *since $f'(p)$ is close to -3.* You may also want to graph g for these two choices of C. Your programs should have a tolerance tol and an N_{max}. The loop should stop if $|p_{n+1} - p_n| \leq tol$, or if $n > N_{max}$. For each method, see how many iterations it takes to converge with $tol = 10^{-6}$ and reasonable initial guesses.
- See if any of the methods give a solution with $tol = 0$ in fewer than $N_{max} = 10$ iterations. (That is, stop if the floating point approximation to p_{n+1} and p_n are *the same.*) When $p_{n+1} = p_n$, is this always the same number for different initial conditions and different methods? (Print 17 digits to see all the double precision approximation.) Comment on the results.
- For each of these methods, make a table of n, p_{n+1} and the error $p_{n+1} - p$ (using scientific notation) up to $n = 10$. It is very important that you use the most accurate floating point estimate for the solution p. (You hopefully found this p in the previous bullet.)
- Now consider the *order of convergence* for each of these methods. Test for linear convergence ($\alpha = 1$) and quadratic convergence ($\alpha = 2$), as well as "golden ratio convergence" ($\alpha = \phi := (\sqrt{5} + 1)/2$). The number $\phi \approx 1.618$ is often called the golden ratio. That is, try to determine numerically if this limit λ exists (and is positive) for those three values of α:

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lambda$$

  This is a difficult calculation to do because of roundoff error that occurs when you subtract nearby numbers.

  To this end, define $R(\alpha, n) = |p_{n+1} - p|/|p_n - p|^\alpha$. For each of the five methods, make a table of n, $p_{n+1} - p$, $R(1, n)$, $R(\phi, n)$ and $R(2, n)$ for n up to $n = 10$ or until $p_{n+1} - p = 0$. Comment on the results. You should be able to conclude that some limits do not exist, or that the limit is zero. In some cases you should be able to estimate the asymptotic error constant λ. Quote theorems in the book that predict the values of λ when $\alpha = 1$ for the fixed point iteration method, and the value of λ when $\alpha = 2$ in Newton's method. Get specific predictions of the number λ for the function f you are using here. It is a theorem (but not in the book) that the secant method converges with order ϕ, but do not worry about λ in this case. Are your results consistent with these theorems? Identify those cases where roundoff error makes it difficult to determine if the limit exists.

Groundrules

Work in groups of your own choosing. Two or three people in a group is preferred, but if you want you may work alone or have a group of 4. Share the coding and writing responsibilities. Each group should email me a **pdf** of the report (no MS word documents, please), along with samples of code. Please attach 2 files: a pdf of the report and a zipped file with code and perhaps a readme.txt file. Send me email with subject "Project 1", and cc a copy to the other members of the group .

Your report should have a title, alphabetical list of authors, a short introduction of the topic and contents of the report, the numerical results, and a discussion of the results. Your report will need graphs, tables of numbers, and snippets of code. The report should be self-contained, and make sense without reading this assignment. Reports should be professional in appearance and format, but need not be exceedingly long. Include some code in the report to indicate how you did the calculations. Avoid the inclusion of tedious and/or redundant code and data. The emphasis should be on developing and testing simple, correct code, and presenting the results clearly.

The programs can be written in any language: MATLAB, Mathematica, python, C++, etc. It is allowed, and perhaps advantageous, if more than one language is used within your group. For example, you can make observations comparing the performance of different languages. The code should contain at least some comments. The parts of the code included in the pdf should have liberal comments. You do not need to upload every program you used. I will mainly look at the pdf, but I may also read and test the code you upload.

Added note in version 2: I suggest you use MS Word, with equation editor. If you know LaTeX that would be a great choice. With MS word, you can copy and paste figures in from Desmos Graphing calculator, and you can also copy and paste in tables of output made with print statements like the one in the sample programs I send you via email.