PHYS 331 – Introduction to Numerical Techniques in Physics

Homework 3: Root-Finding and Convergence

Due Friday, Feb. 2, 2018, at 11:59pm.

Problem 1 – Comparison of the Bisection and Newton-Raphson Methods (10 points)

Note: This problem is paper only. No Python files should be submitted.

You are stranded on a desert island with a solar calculator that only has the four basic arithmetic operators (+,–,×,÷). You also have a ruler on hand, and want to construct a cubical box that has a volume of exactly 750 cm^3 (it's a long story), within a tolerance of 0.1 cm^3 (*i.e.*, acceptable values of $V = 750 \pm 0.1$ cm^3). To accomplish this, you need to compute the desired length of the cube, L, in centimeters, fairly accurately. Remembering the good old days when you took PHYS 331, you recall that the Newton-Raphson method provides a way that you can use your simple calculator to iteratively solve for L.

(a) First, define a function, $f(x)$, such that at least one of the roots is the desired cube length, L.

(b) Using Newton-Raphson, write an iteration function (*i.e.*, x_{n+1} = only a function of x_n) using only the operators that are available on your solar calculator. **Simplify the function** as much as possible so that you can perform each iteration with the minimum number of operations on your calculator. What is the minimum number of operations needed to compute your iteration function?

(c) Starting with x_0 = the integer value closest to the true answer, perform the Newton-Raphson iteration, making a list of x_0, x_1, x_2, *etc* as well as their associated volumes, $V_0 = x_0^3$, $V_1 = x_1^3$, *etc*, until you reach the desired tolerance in the volume. How many iterations were required to get the stated accuracy?

(d) If you had used the bisection method with a starting interval of 1 (*i.e.*, the two adjacent integers that bracket the solution), how many iterations would have been required to achieve the needed tolerance? **Do not actually perform the bisection;** this problem is looking for a theoretical answer only. (Warning: consider the fact that the tolerance in this problem is related to the final volume, not the cube length.)

(e) Consider how you would perform Newton-Raphson or bisection to find the root with your calculator in practice, including all of the steps needed. Write out each of the steps you would use for each method, and the number of operations in each step. Add up the total number of arithmetic operations needed for each method to achieve the desired tolerance, using the number of iterations from your answers in parts (c) and (d). There may be different answers depending on your approach and assumptions; make sure to explain. Which method requires fewer operations?

Problem 2 – Newton-Raphson Root-Finding and Basin of Convergence (15 points)

Consider the real roots of the following 5th order polynomial, which, being of order >4, may not have an analytical solution:

$$f(x) = x^5 - 3x^3 + 15x^2 + 29x + 9$$

(a) Write a Python function of this function, `func(x)`, and explore plotting it over different ranges to find all **real** roots. Provide a final, nice plot of the function over a limited range of x where all of the real roots can be viewed easily. How many real roots are there and what are their approximate locations?

(b) Write the Newton-Raphson iteration equation for solving the roots for this function. Write out the entire expression as a function of x_n so that it can be readily implemented in Python in the next step.

(c) Now, implement the root-finding of the above as a Python function `Newt(xstart,tol)`, where `xstart` is the starting value, and `tol` is the desired tolerance. The function should return the root found. In this case, we define the error as the difference between the computed root in successive iterations (*i.e.*, $| x - x_{old} |$). Measure each root to a tolerance of 10^{-20}.

(d) Now let's examine the basin of convergence for these roots, recalling that the convergence criterion is given by $| g'(x) | < 1$, where g(x) is the function used in the fixed-point method x = g(x). First, rewrite your iteration expression as a fixed-point method, and calculate $| g'(x) |$, (it's messy – don't worry about multiplying it out, just keep it factorized for simplicity.) Now, write this as a Python function `dg(x)`. Make an overlay plot that includes: $| g'(x) |$, f(x), the line y=1, and the line y=0. (Scale the plot as needed to observe the desired features). Is the convergence criterion satisfied in the region of some or all of the roots? Is this what you would expect for a Newton-Raphson technique?

(e) Using your `Newt` function from part (c), what happens when you select a value of `xstart` for which $| g'(x) | > 1$? Try several different values in several different regions, report your observations, and explain what is happening.

Problem 3 – A Computational Physics Problem (12 points). Do problem 15 of problem set 4.1 in the textbook (natural frequencies of a uniform cantilever beam). Use the supplied *HW3p3template.py*, which is the "safe" Newton-Raphson function from the textbook. Do not modify this function. Note that the moment of inertia for a rectangular cross-section is $I=bh^3/12$, where b is width and h is height. Add to the supplied .py file any plots, functions, and comments needed to show your thinking processes, and make sure all results (including plots and answers) display automatically when executing your file. (Hint: pay attention to units.)

Problem 4 – Order of Convergence (13 points).

(a) Similar to Homework 2 Problem 3, modify the Newton-Raphson function provided in the new template file, *HW3p4template.py*, with a diagnostic tool to track the estimated error with each iteration. The function `newtonRaphsonMOD` has already been partially modified to remove unwanted code that will help you with this task. Estimate the error **by the absolute value of the difference between successive iterations**, and have the function **only** output a numpy array of floats corresponding to the error at each iteration.

(b) Use this modified function to find the real root of

$$f(x) = (x-10)(x+25)(x^2+45)$$

that exists in the range of x=(0,15), and plot the error versus number of iterations for 10 iterations. After how many iterations is the error effectively zero (to within machine precision?)

(c) Determine whether the order of convergence from the above data is more consistent with m=1, m=1.5, or m=2. Use the fact that:

$$\varepsilon_{n+1} = c\varepsilon_n{}^m$$

and thus when the correct value of m is chosen,

$$\frac{\varepsilon_{n+1}}{\varepsilon_n{}^m} \approx const \quad in \quad n \tag{1}$$

The best way to do this is by trial and error to see which value of m is most consistent with the error convergence model. Is the value of m what you expected?