PHYS 250 Homework 3

Due: 17 February 2025

Problem 1. (10 points) We might think that all this work to make sophisticated algorithms to compute derivatives is not necessary, why not just make h extremely small? Here we will see that this approach does not work. As an example, we will consider $f(x) = \mathrm{e}^{-x/4}$ and evaluate the derivative at $x = 0.65$ using center differencing.

(i) Calculate the fractional error in the numerical derivative for $h = 10^{-4}$, 10^{-5}, 10^{-6}, and 10^{-7}. You should find that the error starts growing at some point as h gets smaller! The fractional error can be calculated in at least of couple of ways that are analytically equal but numerically they are not (for reasons we understand, …, numerical precision). For the fractional error use

$$f_{\text{error}} = \left| 1 - \frac{\text{approximate}}{\text{true}} \right|,$$

where "approximate" is the approximate value you calculate using some algorithm and "true" is the actual value from an analytic calculation (though, of course, evaluated numerically). [*Note*: Since it is not hard you really should calculate the error for a large number of h values and make a log-log plot. You will see that the error is not a smooth function but does have a general trend of growing as h gets smaller. You do not need to show this work, doing it yourself will make things more clear.]

(ii) How can the error grow as h gets smaller? Typically when we talk about the error in an algorithm we write it as $\mathscr{O}(h^n)$. This is true while the algorithm itself dominates the error, but there is another source of error: round off error due to the limit precision for storing numbers. As h gets very small, the precision error begins to dominate. For the center differencing algorithm it can be shown that the error $e(h)$ is given by

$$e(h) = \frac{\epsilon}{h} + \frac{h^2}{6} M.$$

Here the second term comes from the center differencing algorithm with M being a bound on the error from the higher order derivative (see the next part) and the first term is due to numerical precision with ϵ being the precision at which we are doing our calculation. Calculate the value of h in terms of ϵ and M for which $e(h)$ is a minimum.

(iii) Typically we cannot calculate the numerical value for the optimal h but in the example we have considered here we can estimate it. For double precision calculations $\epsilon \sim 10^{-16}$. The bound, M, comes from the neglected term in our algorithm,

$$M \le \left| \frac{\mathrm{d}^3 f(x)}{\mathrm{d}x^3} \right|$$

evaluated at some value of x, we will not be too precise about this. For our $f(x)$ what is a "reasonable" estimate for M? In this case we can calculate the third derivative. Use this to estimate the value of h for which $e(h)$ is a minimum. [*Hint*: Our $f(x)$ is a simple function. What is its third derivative and what is the maximum magnitude it can have for all $x \ge 0$? This is a rough estimate of the bound on M. We could do better by using a narrower region around the point at which we are evaluating the derivative, but this is a reasonable estimate.]

Problem 2. (20 points) We have seen that center differencing is more accurate than forward differencing. Due to this we may jump to the conclusion that using center differencing is always better. Here we will explore this idea. For our purposes we will use the function

$$f(x) = 3^x \sin(x),$$

evaluated at $z = 1.15$ and a step size $h = 0.4$.

(i) Analytically calculate the derivative, $f'(x)$.

(ii) Estimate the derivative, $f'(z)$, using Richardson extrapolation of both the forward and center differencing algorithms up to $n = 8$. What is the magnitude of the fractional error in each case? (See the previous problem for comments about calculating the fractional error.)

(iii) Determine the number of iterations, n, required for each algorithm to compute $f'(z)$ to an accuracy of 10^{-12}.

(iv) You will have found that center differencing requires fewer iterations. Does this mean it is better? More sophisticated algorithms come at a cost. As we see from the two algorithms, once we calculate $F_1(h)$ for the various values of h we need the iterations are just algebraic manipulations and, in fact, are essentially the same in both cases. Thus, all the cost in the algorithms come from evaluating $f(x)$ at various points. If calculating $f(x)$ were expensive, for example it is the solution to some complicated equation that takes hours to compute for each value of x, then we want to minimize the number of function values we need. To evaluate $F_1(h)$ for both algorithms requires 2 function evaluations. How many **new** function evaluations are required to calculate $F_1(h/2)$ in each case? Use this to find expressions for the number of function evaluations needed to perform n iterations of Richardson extrapolation for both algorithms.

(v) Using your expressions from the previous part how many function evaluations are required for the results from part (iii)?

(vi) Find the minimum error and the number of iterations that give this minimum error for estimating $f'(z)$ for both algorithms.

(vii) Again we see there is a minimum error. Why do the algorithms eventually fail? Mathematically we can take the limit $h \to 0$, describe what happens numerically to prevent us from doing this?