PHYS 512: Problem Set 1

Brandon Ruffolo

September 20, 2019

1. Problem 1.(a):

$$\frac{f(a+h)-f(a-h)}{2h} = f'(a) + \frac{f'''(a)h^2}{3!} + \frac{f^{(5)}(a)h^4}{5!} + \mathcal{O}(h^6)$$
$$\frac{f(a+2h)-f(a-2h)}{4h} = f'(a) + \frac{4f'''(a)h^2}{3!} + \frac{16f^{(5)}(a)h^4}{5!} + \mathcal{O}(h^6)$$

We add these two so as to cancel the h^2 order:

$$\frac{4}{3}\left(\frac{f(a+h)-f(a-h)}{2h}\right) - \frac{1}{3}\left(\frac{f(a+2h)-f(a-2h)}{4h}\right) = f'(a) + \mathcal{O}(h^4)$$

And so the optimal four point numerical derivative is

$$f'_{4h}(a) = \frac{8(f(a+2h)-f(a-2h)) - f(a+2h) + f(a-2h)}{12h}.$$

The error for a given h is

$$\text{error}(h) = |f'(a) - \overline{f'_{4h}}(a)|,$$

where $\overline{f'_{4h}}(a)$ is the floating point representation of $f'_{4h}(a)$. We can place upper bound on the error, using the method shown in class, of

$$|f'(a) - \overline{f'_{4h}}(a)| \leq \frac{|f^{(5)}(\xi_1)|h^4}{18} + \frac{3|f(\xi_2)|\varepsilon}{2h}.$$

where $f^{(5)}(\xi_1) = \max(f^{(5)})$ and $f(\xi_2) = \max(f)$ on $[a, a+2h]$. We will make the approximation $\xi_1, \xi_2 \approx a$. Finally we differentiate the error with respect to h and set the result to zero to find the optimal value of h,

$$\frac{\partial}{\partial t}\text{error} = \frac{2|f^{(5)}(a)|h^3}{9} - \frac{3|f(a)|\varepsilon}{2h^2} = 0.$$

This gives,

$$h = \sqrt[5]{\frac{27}{4}\left|\frac{f(a)}{f^{(5)}(a)}\right|\varepsilon}$$

1.(b):

Using the above formula for the optimal h, we find:

For $f_1(x) = e^x$ the optimal h is about $x = 0$ is $\approx 10^{-3}$
For $f_2(x) = e^{0.01x}$ the optimal h about $x = 0$ is $\approx 10^{-1}$

Which is in very good agreement with the plot produced by Q1.py, which calculates the relative error between the numerical and analytical derivatives of f_1 and f_2 for various magnitudes of h, and plots the result.

2. Problem 2: Two different methods were attempted.

   The script Q2_spline.py uses a spline interpolation. After running the script the user is asked to input a temperature, and the diode voltage at that temperature will be interpolated and returned to the user. The code will also plot the full spline curve, as well as the derivative of the spline, both of which can be compared with the overlaid measurement data.

   The script Q2_hermite.py uses the cubic Hermite interpolation. The datasheet of these diodes includes not only a set of measured diode voltages V at a given temperatures T, but also the first derivative at same temperatures. The cubic Hermite interpolation allows incorporation of the derivative information at a point, along with the value of the function at that point. Rather than fitting the four degrees of freedom of the cubic to four diode voltages (as was done in the spline), we fit two degrees of freedom to a set of diode voltages and the other two to the respective derivatives at those voltages. As with the spline, the code will ask the user to input a temperature, and the diode voltage at that temperature will be interpolated and returned to the user. The code will also plot the interpolation curve and derivative.

3. Problem 3:

   The "Q3.py" file contains two integrator functions, "optimized_integrator" and "simple_integrator". The "optimized_integrator" does not call $f(x)$ multiple times for the same x, as compared to the "simple_integrator", which is a direct copy of the class code that does. A separate executable python file, "integrator_tester.py" will compute and display the results of several integrals performed with the two integrators, as well as the number of function calls required for each. The following table shows some examples:

Integral	Number of function calls for old integrator	Number of function calls for new integrator
$\int_{-10}^{10} \frac{1}{1+x^2} dx$	2175	873
$\int_{-1}^{1} 1 + e^{-0.5(x/0.1)^2} dx$	1015	409
$\int_{-1}^{1} \cos(x) dx$	295	121

   In general, the optimized integrator uses fewer than half the number of function calls that the simple integrator does.

4. Problem 4:

   The integral for the electric field of a charged spherical shell is:

$$|\vec{E}\,(z)| = E(z) = \frac{\sigma R^2}{2\epsilon_0} \int_0^\pi \frac{(R\cos\theta - z)\sin\theta}{(R^2 + z^2 - 2rz\cos\theta)^{3/2}}$$

   Getting rid of the constants, normalizing the integral and putting it into appropriate form for our computation:

$$E(z/R) = \frac{1}{2} \int_0^\pi \frac{(\cos\theta - z/R)\sin\theta}{(1 + (z/R)^2 - 2(z/R)\cos\theta)^{3/2}}$$

   Where we can have z/R varying from 0 (centre of the sphere) to $+\infty$ (infinitly far from the edge of the sphere).

   The script Q4.py will produce a plot of the value of this integral as a function of z/R going from 0 to 20, as evaluated by both `scipy` quad and the integrator written in problem 3. The analytic solution

to the integral is also plotted for reference. The analytic solution is actually hard to see on the plot, as it almost exactly overlaps the two numerical integration results.

The value of the integral at $z/R = 1$ is not plotted. The `scipy` quad integrator has no problem at $z/R = 1$, but our variable step size integrator crashes with this input. This occurs because the integrad has a singularity at $\theta = 0$. The variable step size integrator, using Simpsons's integration rule, tries to evaluate the integrad at endpoints of the integration interval $(0, \pi)$, but the denominator of the integral is 0 at $\theta = 0$, causing it to fail.