PHY 688: Numerical Methods for (Astro)Physics
Homework #1

Due: 2013-02-20

Programs can be written in any language, and should be accompanied by a short description of how to compile and run them (you can put this in the comments at the head of the source code). Assignments will be submitted using *git*. All registered students should have received a document listing their username and password for `bender.astro.sunysb.edu`, along with some instructions on how to access their git repository.

1. (*Numerical derivatives*) Here we explore two different approaches to deriving numerical derivatives.

   (**a**) Derive a second-order accurate *one-sided* derivative at $x = x_i$ using the function value at the three points x_i, x_{i+1}, and x_{i+2}, assuming that the spacing, Δx, is constant.

   (**b**) In class (when deriving Simpson's rule), we showed that a quadratic function passing through $f(x)$ at points x_0, x_1, and x_2 is

   $$f(x) = \frac{f_0 - 2f_1 + f_2}{2\Delta x^2}(x - x_0)^2 + \frac{-f_2 + 4f_1 - 3f_0}{2\Delta x}(x - x_0) + f_0 \tag{1}$$

   Show that (**i**) the derivative of $f(x)$ at x_1 recovers the centered difference formula and (**ii**) the derivative of $f(x)$ at x_0 gives an analogous expression to that from part (**a**), but right-sided instead of left-sided.

2. (*Derivative error estimates*) Starting with the centered difference:

   $$\Delta_1(h) = \frac{f(x+h) - f(x-h)}{2h} \tag{2}$$

   write a program to compute the numerical derivative of $f(x) = \sin(x)$ at $x = 1$. By comparing $\Delta_1(h)$ and $\Delta_1(h/2)$, control h until you reach a *relative* error of $\epsilon = 10^{-7}$.

   Since you have both $\Delta_1(h)$ and $\Delta_1(h/2)$, return the Richardson extrapolated value of f' that is $\mathcal{O}(h^4)$ accurate.

3. (*Simpson's rule*) In class we derived the compound version of Simpson's rule, noting that we integrate over pairs of slabs/intervals.

   (**a**) Imagine that you want to integrate $f(x)$ over $[a, b]$, and have divided the domain into an odd number, N, slabs/intervals, with the function specified at the points $x_0, \ldots, x_N$.

   In this case, you would integrate all the pairs of slabs up until the last slab. For the remaining odd slab, $[x_{N-1}, x_N]$, show that a Simpson's rule for this slab is

   $$\int_{x_{N-1}}^{x_N} f(x)dx \approx \frac{\Delta x}{12}(-f_{N-2} + 8f_{N-1} + 5f_N) \tag{3}$$

   (Hint: fit a parabola to the last three points and integrate over the last slab). This is Pang, Eq. 3.26.

   (**b**) Integrate $f(x) = \sin(\pi x)$ over $[0, 1]$ using N = 3, 7, 15, & 31 slabs/intervals, and plot the absolute error vs. $\delta = (b - a)/N$ on a log-log plot

4. (*Gaussian quadrature*) The "magic" of the Gauss-Legendre quadrature method is that with n quadrature points, the method will give the exact integral of a polynomial up to degree $d = 2n - 1$. Here we show this numerically.

   Consider a 5-point quadrature. For the Gauss-Legendre method, the roots and weights are:

roots	weights
0	$128/225$
$\pm\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$	$\frac{322+13\sqrt{70}}{900}$
$\pm\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$	$\frac{322-13\sqrt{70}}{900}$

(Wikipedia).

Consider a 9th degree polynomial:

$$p(x) = \sum_{k=0}^{9} x^k \tag{4}$$

Compute

$$I = \int_0^1 p(x)dx \tag{5}$$

using the compound version of Simpson's rule and Gauss-Legendre quadrature, in both cases using 5 points. For Simpson's rule, this would mean 2 pairs of intervals. Compute the error against the exact integral.

You should observe that the 5-point Gauss-Legendre quadrature gets the integral exact. You can also see that it gets an 8th, 7th, . . . degree polynomial exact, but not at 10th degree.