Radiative Gas Dynamics
Problem Set 1
Due Thursday, Jan. 18

The goal of this problem set is to write a program to do numerical integrals of user specified functions, and to compare the performance of some simple algorithms for doing so. You may find chapter 4 of Numerical Recipes to be useful reading, especially the early sections. However, while their description of algorithms and their properties is good, you should write your own code rather than borrow the NR subroutines.

You will modify your integration program and use it for a hydrodynamics (well, hydrostatics) problem in a future problem set.

1. Write a program in a language of your choice to compute

$$I = \int_a^b f(x)dx,$$

where a and b are finite limits of integration and we will use various choices for $f(x)$ below. Use two different approximations to calculate the integral as a discrete sum:

(1) The "Euler method":

$$I = \sum_{i=1}^{N} f(x_i)h_N,$$

where N is the number of (equal-sized) integration steps and

$$h_N = \frac{b-a}{N}, \qquad x_i = a + (i-1)h_N.$$

(2) The "Trapezoidal Rule":

$$I = \sum_{i=1}^{N} \left[\frac{1}{2}f(x_i) + \frac{1}{2}f(x_{i+1})\right] h_N.$$

(This seems at first sight to require twice as many function evaluations as the Euler method, but there is an obvious way to avoid this, which you should implement in your program.)

Numerically compute the integral

$$\int_1^5 \frac{1}{x^{3/2}}dx$$

with both methods, and plot the error in the numerical integral against the step size h_N for both methods. (Include this plot as part of your solution to the problem set.)

How many steps (approximately) are required to get an answer with a fractional error $|(I - I_{\rm exact})/I_{\rm exact}| < 10^{-3}$ for the Euler method and for the trapezoidal rule? What about 10^{-5}?

2. Sketch a graphical interpretation of the Euler method and the trapezoidal rule (i.e., show how they approximate the area under a curve as a sum of discrete areas). Why does the trapezoidal rule converge so much faster?

3. If you didn't write your program this way in the first place, revise it so that it automatically doubles the number of steps until it converges to some desired fractional tolerance. Specifically, you should be able to specify the desired tolerance and an initially small number of steps to try (e.g., $N = 4$). The program should calculate the integral with this number of steps, then double the number of steps and compare the answers. If the fractional difference is larger than the specified tolerance, it should double the number of steps again, and so forth, until the fractional difference between two successive evaluations is less than the tolerance. You should have some safeguard with a maximum number of steps, to prevent your program from running forever if it doesn't converge.

With this step doubling, you can implement a neat trick, described in Numerical Recipes. Given estimates IT_N and $IT_{N/2}$ from the trapezoidal rule using N and $N/2$ steps, make the new estimate $IS = (4 \times IT_N - IT_{N/2})/3$. This approximation, Simpson's rule, should converge faster than the trapezoidal rule.

Implement this Simpson's rule trick in your program. Add a plot of error vs. step size for Simpson's rule to your plot from part 1.

4. For each of the following, give the value of the integral I and the number of steps needed to get convergence to a fractional tolerance of 10^{-5}, for each of the three numerical integration methods.

$$\int_1^2 \frac{1}{x^{3/2}(1+x^{3/2})} dx$$
$$\int_1^{100} \frac{\sin x}{x} dx$$
$$\int_1^{1000} \frac{\sin^2 x}{x^{1/2}} dx.$$
$$\int_1^{1000} \left(x + \frac{1}{x}\right)^{-1} dx.$$
$$\int_0^{\ln 1000} \left(1 + e^{-2x}\right)^{-1} dx.$$

5. So that you finish this problem set with something of practical value in hand, use your Simpson's rule routine to write a standalone program that computes the integral

$$D_c = \int_0^z dz' \, [\Omega_m (1+z')^3 + (1. - \Omega_m - \Omega_\Lambda)(1+z')^2 + \Omega_\Lambda]^{-1/2},$$

given input values of Ω_m, Ω_Λ, and z. If you look up Hogg (1999, astro-ph/9905116), equation (15), you will see that this integral, multiplied by $cH_0^{-1} = 3000\, h^{-1}$ Mpc, gives the "comoving distance" to an object at redshift z in a universe with matter density parameter Ω_m and cosmological constant Ω_Λ. This can in turn be used to calculate (with no additional integrals) other cosmologically useful distances like the luminosity or angular diameter distance. What is the comoving distance to $z = 2$ in a universe with $\Omega_m = 0.3$ and $\Omega_\Lambda = 0.7$?

Escape Clause

You should spend at least 2-3 hours writing your integration program yourself, from scratch. At that point, if things aren't working, you can take a look at my program, which is available as `http://www.astronomy.ohio-state.edu/~dhw/A825/integrate.c`. How useful this will be depends in part on how well you know `C`.

If you have spent 5 hours on this problem set and are still not finished, you can use my program to do the integrals instead of yours, but note in your solution that you have used my program. To use my program, you edit the function subroutine `integrand()` to correspond to the function being integrated, compile the program with `cc -o integrate integrate.c -lm`, and run it with `integrate a b`, where `a` and `b` are the lower and upper limits of the integral.