Computational methods in Astrophysics

Homework #2

Programs can be written in c, c++, Matlab, Python or Python notebook (preferably the last). In addition to the program, you should have a writeup that contains plots requested, answers to any analytical or explanation questions, and a short description of your code and how to run it. This can be done in, e.g., LATEX, WORD, etc. If your program requires a compilation to run, add the executable file as well. Code and writeup should be submitted to the Moodle site.

1) *Comparing methods of integration*: (based on Newman exercise 5.7). Consider the function:

$$I = \int_0^1 \sin^2\left(\sqrt{100x}\right) dx \tag{1}$$

a) Plot the integrand over the range of the integral.

b) Write a program that uses the adaptive trapezoid rule to calculate the integral to an approximate accuracy of $\varepsilon = 10^{-6}$, using the following procedure: Start with the trapezoid rule using a single subinterval. Double the number of subintervals and recalculate the integral. Continue to double the number of subintervals until the error is less than 10^{-6}. Recall that the error is given by $\varepsilon_i = \frac{1}{3}\left(I_i - I_{i-1}\right)$, where the number of subintervals N_i used to calculate I_i is twice that used to calculate I_{i-1}. To make your implementation more efficient, use the fact that

$$I_i = \frac{1}{2} I_{i-1} + \mathrm{h}_i \sum_k f(a + k h_i) \tag{2}$$

where f is the function we integrate, h_i is the width of the subinterval in the i'th iteration, and k runs over odd numbers from 1 to $N_i - 1$ (see Newman sec. 5.3).

c) Write a separate program that uses Romberg integration to solve the integral, also to an accuracy of 10^{-6} using the following procedure. First calculate the integral with the trapezoid rule for 1 subinterval (as you did in part b); we will refer to this as step $i = 1$, and to the result as $I_1 \equiv R_{1,1}$. Then, calculate $I_2 \equiv R_{2,1}$ using 2 subintervals (make use of Eq. 2). Using these two results, we can construct an improved estimate of the integral as: $R_{2,2} = R_{2,1} + \frac{1}{3}(R_{2,1} - R_{1,1})$. The general for the integral in the $m + 1$ iteration is (Newman sec. 5.4):

$$R_{i,m+1} = R_{i,m} + \frac{1}{4^m - 1}\left(R_{i,m} - R_{i-1,m}\right) \tag{3}$$

The iteration method is as follow: (i) Calculate the first two integral estimates using the regular trapezoidal rule, $I_1 \equiv R_{1,1}$ and $I_2 \equiv R_{2,1}$ and from these calculate $R_{2,2}$ using Eq. 3. (ii) Now, calculate the next trapezoidal rule estimate, $I_3 \equiv R_{3,1}$ and from this calculate $R_{3,2}$

and $R_{3,3}$ using Eq. 3. (iii) At each successive stage compute one more trapezoidal rule estimate $I_i \equiv R_{i,1}$, and from it calculate $R_{i,2}$ ... $R_{i,i}$.
For each iteration i (where we double the number of subintervals), we can obtain improved approximations up to $m = i - 1$ with very minor extra work. For each step $m + 1$ in the iteration we can evaluate the leading term of the error of the previous step (m) as:

$$\varepsilon_{i,m} = \frac{1}{4^m - 1}\left(R_{i,m} - R_{i-1,m}\right) \tag{4}$$

Use Eqs. 3 and 4, to iterate until the error in $R_{i,i}$ is less than 10^{-6}. How significant is the improvement with respect to number of subintervals necessary compared to the approach of part b?

2) *Cubic spline of a noisy function.* Consider the function

$$f(x) = \frac{\sin(x - x^2)}{x} \tag{5}$$

a) Write a program that adds random noise to $f(x)$ so that $f_{noisy}(x) = f(x) + \lambda\delta(x)$, where $\delta(x)$ is a random number in the range $[-1, 1]$, and λ is a scaling factor. Plot $f(x)$ and $f_{noisy}(x)$ in the range $[0.5, 10]$. Take as your data set for the next parts (b, c) 100 equally-spaced points of $f_{noisy}(x)$ in that range.

b) Calculate the numerical derivative of this dataset across the range. By plotting $f'(x) - f'_{noisy}(x)$. Comment on how the noise in the derivative changes with λ, i.e., compared to the level of noise in the data.

c) Write a program that performs a cubic spline interpolation of the noisy data set (feel free to base it on the program discussed in class, but please do not just use a library). Use the spline to (analytically) take the first derivative of the data. Comment (using plots) on the noise in the numerical derivative in (b) versus what you obtain from the first derivative of the spline, and how it changes with λ.

3) *Multiple roots of functions.* Write a program that finds all of the real roots in a given interval of a given function using the Bisection, Newton-Raphson, and Secant methods. Test your program on the function from the previous problem:

$$f(x) = \frac{\sin(x - x^2)}{x} \tag{6}$$

Comment on the relative speed of convergence of the Bisection method versus the Newton-Raphson method.