Math 245 Project 1: Numerical Calculus (Approximating Definite Integrals, and Derivatives)

Brenton LeMesurier

October 9, 2015

Reference: Chapter 5 of *Numerical Analysis* by Timothy Sauer; in particular Sections 1 to 3.

This project will combine and explore methods for approximation of derivatives and integrals, along with methods for estimating errors and *using these error estimates to improve accuracy*; the strategy already seen with Newton's method,and soon to be seen with *Richardson extrapolation*.

We will also compare the performance of various methods: this is often important in making the choice of algorithms for a larger task.

In part, this is an exercise in *good written presentation of mathematical and computational results*, and as preparation for the second individual project that each of you will do at the end of the semester. The "introduction" and "methods" sections specified below can be quite brief in this project (as many details are specified below) but will be more significant in your final project.

Report format, requirements, and goals

The submission for this project will be centered on a written report, preferably as an IPython notebook or LaTeX document submitted via OAKS. However, other formats including hand-written reports combined with *selective* printouts of numerical results are also acceptable.

In either case, all programs and related files will be submitted electronically via OAKS, not on paper. This will include all relevant program files: the core functions implementing the numerical methods, and any scripts or IPython notebooks that organize the calculations needed to produce data for the report.

The main report will have three sections: *introduction*, *methods* and *results and conclusions*. The introduction will briefly state the mathematical problems to be solved; this will be followed by a description of the various methods, and then conclusions, such as comparisons of cost/speed between methods. The conclusions will be supported by including appropriate selections and summaries of computed output. Be selective, cutting and pasting from raw output files if needed, so as to make you points in a readable and reasonably concise fashion.

As with all programming work and almost any substantial project, there will be drafts, feedback and revisions: in particular, we will check that your programs are working before committing to writing the results section of the report.

Project tasks

1. Write a function to approximate derivatives using the centered difference formula. The input variables should include the node spacing h; think about and discuss what all the input and output variables should be. Also, remember that such a function should not do any interactive input, or any output to the screen or files.

2. For the above function and every function below, write a Python "script" file or IPython notebook to run test cases: this will handle any interactive input and any screen or file output.

3. Write a function to apply Richardson extrapolation using the above centered differences function, and a file to test it and handle any interactive input and screen or file output, as noted above. As with almost any practical implementation of a numerical approximation algorithm, the input should include an **error tolerance**, and the output should include an **error estimate**.

4. Write a function that can approximate any definite integral $I = \int_a^b f(x)\,dx$ using the (composite) trapezoid rule with n sub-intervals of equal width, T_n.
   The input should specify the function f to integrate, the interval $[a, b]$, and the number n of sub-intervals to use.
   The output will be just the approximation value of the integral.

5. Write a function that uses the recursive trapezoid rule, $R_m = T_{2^m}$ to estimate a definite integral with a specified absolute error tolerance.

   The input should specify the function f and interval $[a, b]$, and also the error tolerance.

   The output should include the approximate answer, estimated error, and also the number of evaluations of $f(x)$, as a measure of cost.

6. Write a function based on the *modified trapezoid method*

$$T_n' = T_n - \frac{f'(b) - f'(a)}{12} h^2,$$

   to estimate a definite integral with a specified absolute error tolerance. Input and output should be as above for the recursive trapezoid rule.

   As the input includes only the function f, not its derivative f', this must use appropriate derivative approximation methods from Section 5.1: we will discuss this in class and during some lab time.

7. Write a function to implement the Romberg method.

   Again, the input should include an error tolerance, and the output should include estimates of absolute error and of cost.

Your final version should be arranged with all the integrating functions in a single module, and then define and run all test cases from one or more scripts or notebooks. However, development might be done with a single file for each function (`trapezoid()`, etc.), including code for testing below the code for the main function.

For initial testing, evaluate some integrals of polynomials of degree from one to four (for which some methods should give exact results), and then $\int_1^3 \frac{dx}{x^2}$. Then choose some more challenging test cases.