MATH 3340 - Scientific Computing Homework 7

Due: Monday, 04/27/2020, 11:59 PM

The deadline will be strictly enforced. If you do not submit in time there will be a 20% penalty for each day you're late. If you do not submit in time there will be a 20% penalty upfront plus another 20% for each day you're late. Remember that you are allowed to work in teams of two on this assignment. You are encouraged to prepare your work in LaTeX; a template will be provided to help you put it all together. If you choose to submit a hard copy, you may submit only one copy for a team, indicating the names of both contributors. Online submission is encouraged, however, in that case both members of a team should submit the PDF file containing their work and showing both their names.
All plots generated in this homework should have a title, legend, and labeled x and y-axes.

Instruction

1. Go to `https://www.overleaf.com` and sign in (required).
2. Click *Menu* (up left corner), then *Copy Project*.
3. Go to `LaTeX/meta.tex` (the file `meta.tex` under the folder `LaTeX`) to change the section and your name, e.g.,
   - change author to `\author{Albert Einstein \& Carl F. Gauss}`
4. For Problem 1 and 2, you are encouraged to type solutions in LaTeX. But if you want to write it on the printout, make sure your scanned work is *clear* enough, and compile all solutions *in order*, i.e., 1, 2, 3, in a single PDF (failure to do so will lead to points deduction).
5. For Problem 3, you need to write function/script files, store results to output files, and save graphs to figure files. Here are suggested names for function files, script files, output files, and figure files:

Problem	Function File	Script File	Output File	Figure File
3	`richardson.m`	`hw7_p3.m`	`hw7_p3.txt`	

   Once finished, you need to upload these files to the folder `src` on Overleaf. If you have different filenames, please update the filenames in `\lstinputlisting{../src/your_script_name.m}` accordingly. You can code in the provided files in hw7.zip, and use the MATLAB script `save_results.m` to generate the output files and store the graphs to `.pdf` files automatically (the script filenames should be exactly same as listed above).
6. Recompile, download and upload the generated PDF to WyoCourses.
7. You may find LaTeX.Mathematical.Symbols.pdf and the second part of Lab 01 Slides and Lab 02 Slides helpful.

Problem 1. Use Richardson extrapolation to approximate the derivative of the function

$$f(x) = \frac{e^x}{1 + 2x^2}$$

at the points $x = 0$ and $x = 1$. For the first point, use three mesh sizes defined by $h = h_1 = 0.4$, $h_2 = 0.2 = h_1/2$. Start by calculating by hand, rounding off to four decimal places, the approximate values denoted in the class notes by $E_{1,1}$, $E_{2,1}$ and $E_{3,1}$ (see the March 30 version, page 97). Then use these values to compute the extrapolated approximation $E_{3,3}$. To that end, use the last equation on the same page in the notes that defines $E_{n,m+1}$ in terms of $E_{n,m}$ and $E_{n,m-1}$. Repeat the calculation for $x = 1$, this time using only the first two step sizes to obtain $E_{2,2}$. Please note that in the hand written notes from April 15 the quantity $E_{n,m}$ was denoted by $D_{n,m}$.

Problem 2. Compute, again by hand, the value of

$$\int_0^{2\pi} x\sin^2(x)\,dx$$

using both the trapezoidal rule and Simpson's rule. For a fair comparison, keep the same number of function evaluation, in this case at five equi-spaced $\{x_0, x_1, x_2, x_3, x_4\}$. Use the same round-off strategy as in the first problem, keeping a minimum of four decimal places in all your calculations.

Problem 3. Write a MATLAB code that implements the calculation for $x = 0$ in the first problem above, but for the more general case of an arbitrary number of divisions of the step size. That amounts to allowing the second index m for the approximation $E_{n,m}$ to take values in some range $1, 2, \ldots, M$, where M is provided upon input. The range for n necessarily follows. As usual, organize you code as a MATLAB script that calls a MATLAB function. The latter should accept M as input and compute the entries $E_{n,m}$ that can be stored in a matrix. You can either print these values in the function itself or return them as output and print them in the script that calls the function.