Assignment #8

Due Thursday, 21 November 2019, at the start of class

This Assignment is based on Chapter 10 material only. Please read all of sections 10.1–10.5. You do not need to worry about sections 10.6 and 10.7. Reading section 9.2 about Richardson extrapolation would also be a good idea, but I will go over this idea in class; it relates only to Romberg integration in section 10.5.

Remember that when you turn in homework problems involving MATLAB/OCTAVE, the following two expectations always apply:

1. The commands/code that you ran are shown, along with the results.
2. Minimal paper is used, while still fully answering the question.

Do the following exercises:

CHAPTER 10

- Exercise 2 on page 248
- Exercise 3 on page 249
- Exercise 4 on page 249
- Exercise 6 on page 249
- Exercise 7 on page 250 (*In this problem the built-in integrator* `quad` *is only used to get the "exact" value of the integral. It should be called with an anonymous function like this:*
  ```
  q = quad(@(x) cos(x.^2),0,1,[1e-12 1e-12])
  ```
- Exercise 8 on page 250 (*The way book implements Romberg integration is merely o.k. It would be better to implement it yourself, for instance as shown in class, because that way you will really understand it. In grading this problem I will not care which implementation of Romberg you use.*)
- Exercise A.
  (i) By using a substitution, compute the exact value of the integral
  $$\int_{-1}^{1} x^2 \sin(-5x^3+1)\,dx.$$
  Plot the integrand $f(x) = x^2\sin(-5x^3+1)$ on the interval $[-1,1]$.
  (ii) In Exercise 7 above you wrote a composite Simpson's rule code. Use it, perhaps making it into a convenient function like
  ```
  z = mysimpsons(f,a,b,n)
  ```
  first, to compute the error when approximating the above integral for $n = 10, 20, 40, 80, 160$ points.

(iii) I posted a very short code called `clenshawcurtis.m`. It is the best possible implementation of the method described in section 10.4. Use it to compute the error when approximating the above integral, again for $n = 10, 20, 40, 80, 160$ points.

(iv) Use `semilogy` to make a plot of the computed errors, for the two above methods in parts (ii) and (iii), versus n. (*That is, put n on the x axis and the errors on the y axis, but with logarithmic scaling of the errors.*) Briefly describe what you see, and why it is this way.