MATH 353: Engineering Mathematics III – Section 012

Spring 2014 (F.–J. Sayas) Lab # 7 March 28

Open Matlab and *move to the Desktop or to a folder where you can find your work* at the end of the session. Type these lines

```
>> diary myworkMarch28
>> format long
>> format compact
```

Download the function `midpointrule.m` and the script `scriptMarch28.m` from my website.

At the end of today's lab, you should have produced two new functions `trapezoidrule.m` and `simpsonrule.m`. You will have to give them in the next HW assignment, so try to do a good job today in keeping what you did.

---

1. In the first exercise, I'm giving you almost everything done and I'm asking you to wrap it up. The function `midpointrule` computes the composite midpoint rule approximation to an integral

$$\int_a^b f(x)\mathrm{d}x \approx M_h(f) = h\sum_{i=1}^m f\left(\frac{x_{i-1}+x_i}{2}\right) \qquad h=\frac{b-a}{m}, \qquad x_i = a + i\,h.$$

Have a look at the code. There's a new function `sum` you might not know. Can you figure out what it does? The script `scriptMarch28.m` chooses a particular function and a particular interval (and gives the exact value of the integral) and tests with m in the following list of numbers

$$1 \qquad 2 \qquad 4 \qquad 8 \qquad 16 \qquad 32 \qquad 64.$$

Each experiment doubles the number of integrals. We then compute the error. What we know from class is that

$$E_h = \left|\int_a^b f(x)\mathrm{d}x - M_h(f)\right| \le \frac{b-a}{24}\,h^2 \max_{a\le x\le b}|f''(x)|$$

which we can shorten (ignoring constants) by $E_h = \mathcal{O}(h^2)$. You are asked to do three things:

- Have a look at the errors and explain how they confirm the theoretical result.
- Do a loglog plot (using circular markers joined by lines) of these errors (h, E_h) , side by side with a loglog plot (using a red line) of the predicted behavior (h, h^2) to visually confirm these experiments. (Remember: `loglog`, `hold on`, `'o-'`)
- Repeat the experiment with a bigger list of points $[2, 4, 6, 8, 10, \ldots, 100]$

2. We now want to test the composite trapezoid rule. Build up a function

   ```
   trapezoidrule(f,interval,m)
   ```

   to compute the approximation

   $$T_h(f) = \frac{h}{2}\Big(f(x_0) + 2\sum_{i=1}^{m-1} f(x_i) + f(x_m)\Big)$$

   It might be a good idea to first evaluate `f` at all the needed points, keep this in a vector, and then use the vector to compute the approximation. Once you have constructed it, test it like you did with the midpoint rule:

   - First doubling the number of intervals, starting with a single interval and checking that the errors are divided by .... ?
   - Second, doing a loglog plot of (h, E_h) side by side (h, h^2)
   - Third by performing a large experiment with many possible subdivisions.

3. Finally, we want to test the composite Simpson rule

   $$S_h(f) = \frac{h}{6}\Big(f(x_0) + 2\sum_{i=1}^{m-1} f(x_i) + f(x_m) + 4\sum_{i=1}^{m} f\left(\frac{x_{i-1}+x_i}{2}\right)\Big)$$

   and check that

   $$E_h = \left|\int_a^b f(x)\mathrm{d}x - S_h(f)\right| = \mathcal{O}(h^4).$$

   Write a function

   ```
   simpsonrule(f,interval,m)
   ```

   and repeat the same test that you have done for the previous numerical integration rules.

4. We know that Simpson's rule (the simple and therefore the composite) integrate exactly polynomials up to degree three. Confirm this by integrating

   $$\int_0^1 (x^3 + 2x - 1)\mathrm{d}x$$

   using the simple Simpson rule (take `m=1` in the code).