# Math 4500/6500 Exam #1

Fall 2010

This take-home exam covers the material from Chapter 1 through 5 of Cheney/Kincaid, from floating point numbers through numerical integration. You will need Mathematica to complete the exam. Please don't be afraid to read over the notes and these sections in the book as you work on the problems– there is more in the notes than we were able to cover in the lectures, and some of those extra facts might be helpful to you as you work on the exam problems.

| You **are** permitted to use | You **are not** permitted to use |
|---|---|
| Mathematica | The internet (except for Mathematica help) |
| Our book | Other books |
| Your notes | Other people's notes |
| Your brain | Other people's brains |
| Class notes posted on the website | |
| Mathematica code posted on the website | |

We have covered in class various methods for finding a polynomial $p_n(x)$ of degree $n$ which agrees with a function $f(x)$ at $n+1$ points $x_0, \ldots, x_n$. Another type of polynomial interpolation finds a degree $n$ polynomial $P_n(x)$ so that $n$ derivatives of $P(x)$ agree with the corresponding derivatives of $f(x)$ at $x_0$, or

$$P(x_0) = f(x_0), P'(x_0) = f'(x_0), \ldots, P^{(n)}(x_0) = f^{(n)}(x_0).$$

The first several questions of the test will explore various alternate types of polynomial interpolation:

1. Find an explicit formula for $P_n(x)$ in terms of $x_0$ and $f(x_0), f'(x_0), \ldots, f^{(n)}(x_0)$. (Yes, I realize that this is very easy once you think of it correctly!) Give an error bound for the difference between $P_n(x)$ and $f(x)$ in terms of $(x - x_0)^{n+1}$.
2. Suppose we have two points $x_0$ and $x_1$ and we wish to find a polynomial $Q_3$ of degree 3 which agrees with $f$ and $f'$ at $x_0$ and $x_1$. Solve explicitly for $Q_3$ in terms of $f(x_0), f'(x_0), f(x_1), f'(x_1)$ as well as $x_0$ and $x_1$.
3. Now use the construction of the Newton polynomials to write down a general algorithm for finding a polynomial $Q_{2n+1}$ which agrees with the first $n$ derivatives of $f$ at $x_0$ and $x_1$. Hint: Find the derivatives of $(x - x_0)^k (x - x_1)^j$ at $x_0$ and $x_1$.
4. (Challenge, extra credit) Can you find an error bound for $Q_{2n+1}$ on $[x_0, x_1]$?

The second part of the exam concerns an inventive (if somewhat odd) algorithm for numerical integration. We know that the trapezoid rule works particularly well on periodic functions. Consider the following algorithm for numerical integration on $[0, 1]$, assuming that $n$ derivatives of the function $f$ are available at 0 and 1:

(1) Find a degree $2n+1$ polynomial $R(x)$ so that the function

$$g(x) = \begin{cases} x \in [0,1] & f(x) \\ x \in [1,2] & R(x) \end{cases}$$

is $C^n$ on $[0,2]$ and has $g(0) = g(2), \ldots, g^{(n)}(0) = g^{(n)}(2)$.

(2) Integrate $g(x)$ on $[0,2]$ using the trapezoid rule.

(3) Integrate $R(x)$ on $[1,2]$ symbolically.

(4) Subtract these integrals to arrive at an approximation to the integral for $f$ on $[0,1]$.

1. Code this method in Mathematica for a function $f$. Your code should be called with something like:

   ```
   MyIntegrator[f,Nderivatives,Ntrapezoidsteps]
   ```

   Where would you get started? Here are some hints for a successful solution:
   - The first step is to evaluate the function and its derivatives at $x_0$ and $x_1$. You should know how to take a derivative with Mathematica at this point. The `Table` command should help you make a list of derivatives.
   - Now you need to find an interpolating polynomial which matches your data at $0$ and $1$. Read the documentation for `InterpolatingPolynomial`.
   - Mathematica can integrate the polynomial for you over $[1,2]$ using the `Integrate` command.
   - You'll have to code the trapezoid rule yourself. But there is a `Sum` command in Mathematica, and some examples on the course webpage.
2. Explore the behavior of this integrator on various integrands. Does it perform better or worse than Romberg integration with the same number of steps in the trapezoid rule? Write a mini lab report on your experiments with the integration method.
3. Use the advanced error estimate for the trapezoid rule to give an error bound for this method in terms of the number of derivatives $n$ we take and the stepsize $h$ of the trapezoid rule integration. This error estimate is in the notes, but it's also in the book (in slightly different form) on page 208 as the "Euler-Maclaurin Formula".
4. (For graduate students only) The Logarithmic Derivative of a function $\mathrm{LD}(f(x))$ is defined to be the derivative of $\ln(f)$. Find an estimator for $\mathrm{LD}\, f$ at $x$ using only evaluations of $f$ and the Taylor series expansion of $\ln(f)$. Use one step of Richardson extrapolation to improve your formula. Test your formula in some specific numerical examples.