18.330 Course Overview

Spring 2019, MIT. Instructor: Henrik Ronellenfitsch

February 4, 2019

Class Mechanics

Lectures: Tuesdays and Thursdays, 1pm–2:30pm in Room 2-135
Class website: `http://math.mit.edu/18.330`
Homework and Exams: Weekly PSets, one in-class exam, one final project, no final exam
Grading: 33% PSets, 33% Midterm exam, 34% Final project

About the Class

This course is an exploration of the art and science of extracting numbers from mathematical expressions. The material we will cover may be broadly divided into two units. Unit 1 is all about **basic numerical calculus**. We will discuss classical methods for obtaining accurate numerical estimates of integrals, derivatives, and infinite sums. This unit will include discussions of extrapolation, interpolation, root-finding, optimization, and evaluation of special functions. By the end of Unit 1, you will be in a position to implement and understand the properties of the most basic numerical methods for all of these tasks. However, in several places during the course of the elementary treatment of Unit 1, we will encounter phenomena that seem to be hinting at a deeper set of ideas.

This will set the stage for Unit 2, **Fourier analysis and spectral methods**. The overarching theme here is that we can often revolutionize the speed and accuracy of a calculation by approximating a function as an expansion in some convenient set of expansion functions, often a set of orthogonal functions. Our discussion of orthogonal-function expansions will begin, as must any, with the Fourier series and its immediate descendants (the Fourier transform, Parseval's and related theorems, the Fast Fourier Transform, etc...). Then we will broaden the setting to consider more general classes of functions and more general spectral methods: Gaussian quadrature, Chebyshev polynomials and Trefethen's chebfun, approxfun, the Nyström solution of integral equations, and more. Throughout Units 1 and 2 we will discuss examples drawn from engineering and the sciences, including binding energies of solids, coding and modulation schemes for efficient use of the wireless communications spectrum, spherical Bessel functions for electromagnetic scattering and thermal engineering, and Ewald summation.

This class draws on similar classes that have been given at MIT during previous semesters, in particular those by Homer Reid and Laurent Demanet. A useful textbook companion is *Endre Süli and David Mayers: An Introduction to Numerical Analysis*. However, the course will not follow any particular textbook.

On Computer Programming

Both the class and the homework will involve computer programming. The supported programming language for 18.330 is **Julia 1.1.x** (`http://www.julialang.org`), a new language developed at MIT and specifically optimized for numerical computations. All homework programming code should be submitted in Julia. You can try out Julia online at `http://juliabox.org`.

On Friday, Feb 8, MIT Professor and co-creator of Julia, Steven Johnson, will provide an introductory lecture on the language in Room 32-141 at 5pm–7pm.

Homework collaboration policy

For the homework PSets, collaboration between students is explicitly encouraged. However, each student *must* submit a separate PSet which clearly cites *all* sources used. Sources include: other students through collaboration, websites, books, hearsay, etc...

In addition, each student must separately type in and execute any program code written for the PSet, so that everyone becomes familiar with programming.

Preliminary Class Schedule

This is a preliminary class schedule. While it may change, it reflects the general outline planned for the course.

date	topic
Feb 5	Invitation; Infinite Sums
Feb 7	Numerical Integration, Newton-Cotes
Feb 12	Improper Integrals, ODEs, Euler's method
Feb 15	Beyond Euler: stiff ODEs, implicit methods
Feb 19	**President's day: Monday schedule — no class**
Feb 21	Numerical differentiation, finite difference stencils
Feb 26	Guest lecture: *Christopher Rackauckas* on advanced ODE solvers
Feb 28	Boundary value problems, numerical linear algebra
Mar 5	Fixed and floating point arithmetic, rounding errors
Mar 7	Accumulation of rounding errors, precision loss
Mar 12	Extrapolation and root finding in 1D
Mar 14	Root finding in n-D, Newton's method, Optimization
Mar 19	Monte-Carlo integration
Mar 21	Fourier analysis, Fourier transforms
Mar 23	Parseval, Plancherel, Poisson formulæ, n-D Fourier analysis
Mar 26	**Spring break — no class**
Mar 28	**Spring break — no class**
Apr 2	Fourier series, Paley-Wiener theorem, convolutions
Apr 4	Applications of Fourier analysis 1: Ewald summation
Apr 9	Applications of Fourier analysis 2: Rigorous analysis of Newton-Cotes quadrature
Apr 11	Review session
Apr 16	**Patriots' day — no class**
Apr 18	**In-class midterm exam**
Apr 23	Clenshaw-Curtis quadrature, Discrete Fourier Transform
Apr 25	Applications of Fourier analysis 3: FFT
Apr 30	**Final project proposal due.** Guest lecture: *Steven G. Johnson* on FFTW
May 7	Gauss-Legendre quadrature, Integral equations, Nyström's method
May 9	Numerical linear algebra
May 14	Numerical nonlinear algebra
May 16	Machine learning, stochastic gradient descent

Final Project

For the final project you will research, implement, and test one numerical algorithm of your own choice. You have considerable leeway in what particular algorithm you choose, but in general it should either be one that was not covered in class, or an extended/improved version of one that was covered.

Your report (roughly 15–20 pages of LaTeX) should consist of the following parts:

- Introduction: Some discussion of the origins and history of the algorithm you discuss. Who invented (or discovered) it, when, and what was the original motivation for the invention or discovery?
- A technical mathematical discussion of the theoretical underpinnings of the algorithm. The level of rigor here should be roughly commensurate with the style of the 18.330 lecture notes: You do not need to provide fully rigorous proofs of every statement, but the presentation should follow a solid logical sequence in which one idea flows from the next.
- An implementation in Julia of some version of the algorithm you discuss. This does not have to be a sophisticated code suitable for use in cutting-edge research, and it does not have to be optimized for efficiency but it must be capable of generating at least some results relevant for the problem at hand.
- Results (including plots) showing the performance of your implementation, and (where possible) comparing it against other implementations.

Project proposal

Before getting started on your final project, you must submit a brief proposal to let me know what you are planning to do and how you envision structuring your final report. This proposal is due by April 30, 2019 (you may email it to me directly or submit it in class). If you have trouble finding a good project, contact me early enough and I will try to help.

Below is a rough sample of what a reasonable proposal might look like:

Practical Evaluation of Lattice Sums: Is Ewald Summation Always Best?

Many problems in computational science require efficient and accurate evaluation of lattice sums—that is, sums that accumulate contributions to some physical quantity from sites in a one-, two-, or three-dimensional lattice. Brute-force evaluation of such sums is generally prohibitively expensive, and this fact has spurred the development of more sophisticated and efficient methods. In class we covered one such method—Ewald summation—but we discussed only the electrostatic (zero-frequency) case, and then only for the cases of one- and two-dimensional lattices. My final project will extend this discussion to a broader comparison of lattice-summation methods, including consideration of three-dimensional lattices and nonzero-frequency problems.

My paper will begin with a brief introduction (2–4 pages) to the problem. I will then discuss (3–5 pages) the history of lattice-summation methods, touching in particular on the original contributions of Paul Ewald himself and the physical problems that motivated his development of the Ewald-summation technique, but considering also the evolution of techniques spurred by the computer-aided-design (CAD) revolution of the late 20th century. Then I will explain (5-8 pages) the mathematical underpinnings of several sophisticated lattice-summation methods, including (1) Ewald summation, (2) Kummer decomposition, and (3) integral transforms. Finally, for the specific problem of a 2D lattice sum at nonzero frequency I will implement Ewald summation and Kummer decomposition and compare (3–5 pages) the accuracy and efficiency of these two methods.

Final project grading

The final project is intended to be more like a research paper, and less like a literature review. Hence, a good implementation of the algorithm and presentation of your own computational results are emphasized.

If your report contains	**Your grade can be**
• A code listing of your implementation • A thorough presentation (plots, data tables, etc) of results produced by your code • A thorough discussion of background, theory, and implementation technicalities	100%
• Code listing • Thorough presentation of results • Sketchy discussion of background theory	80%
• Code listing • Sketchy presentation of results • Thorough discussion of background theory	50%
• Little or no presentation of results produced by your code • Lengthy discussion of background theory augmented with lots of pictures copied from textbooks or websites	25%