Math 162a: Numerical Methods for Scientific Computing Spring 2020

Instructor: Thomas Fai (tfai@brandeis.edu)

Time: Tuesday and Thursday 5–6:20pm

Location: TBA

Office Hours: Tuesday 1:30-3:30pm, Goldsmith 208

Introduction

Scientific computing has become an indispensable tool in many branches of research, and is vitally important for studying a wide range of physical and social phenomena. In this course we will examine the mathematical foundations of well-established numerical algorithms and explore their use through practical examples drawn from a range of scientific and engineering disciplines.

Many of the problems that arise in practice cannot be solved analytically and require numerical methods for approximating their solutions. These numerical results must be interpreted with discernment: how can one tell if the solution output by a software package is correct? In this class we will learn how to verify numerical results—by checking convergence and building intuition through limiting cases—and how to avoid the most common pitfalls of scientific computing. We will draw on classic asymptotic approximation methods to motivate our discussion of methods and to build intuition and techniques for verifying computational results.

There will be an emphasis on mathematical theory and numerical analysis to ensure that students understand the concepts underpinning each algorithm that we consider. There will also be a significant programming component in the course. Students will be expected to implement a range of numerical methods in homework assignments to get hands-on experience with modern scientific computing. In-class demos will be performed with Python, and the homework assignments can be completed in any programming language of the student's choice.

Prerequisites

Math 37a and Math 122a or permission of the instructor

Objectives

1. Apply standard techniques to analyze key properties of numerical algorithms, such as stability and convergence

2. Implement methods efficiently in a modern scientific computing programming language

3. Understand and analyze common pitfalls in numerical computing such as ill-conditioning and instability
4. Use asymptotics to build intuition and verify computational results

Course Work and Grading

- Homework: **50**% of total grade.
  - Homework sets are due biweekly.
- Take-home midterm: **15**% of total grade.
- Final project: **35**% of total grade.

Success in this 4 credit hour course is based on the expectation that students will spend a minimum of 9 hours of study time per week in preparation for class (readings, papers, discussion sections, preparations for exams, etc.)

Academic Integrity

Students are encouraged to work together on the final project and homeworks. However, homeworks, including relevant code, will be graded individually and are expected to reflect students' own thoughts and work.

You are expected to be honest in all of your academic work. Please consult Brandeis University Rights and Responsibilities for all policies and procedures related to academic integrity. Students may be required to submit work to verify originality. Allegations of alleged academic dishonesty will be forwarded to the Director of Academic Integrity. Sanctions for academic dishonesty can include failing grades and/or suspension from the university. Citation and research assistance can be found at LTS - Library guides.

Accommodations for students with disabilities

If you are a student with a documented disability on record at Brandeis University and wish to have a reasonable accommodation made for you in this class, please see me immediately.

Course topics

1. Numerical linear algebra and data fitting
   (a) Floating point arithmetic and polynomial interpolation
   (b) Linear systems: LU factorization and rootfinding
   (c) Least squares: the QR factorization
   (d) Eigenvalue problems: Google PageRank and iterative methods
2. Asymptotics
   (a) Convergence of series solutions and symbolic computation

   (b) Padé approximants
   (c) Quadrature and Laplace's method

3. Numerical methods for differential equations
   (a) Stability and accuracy for ODE
   (b) Euler and Runge-Kutta methods, Lax Equivalence Theorem
   (c) Finite difference methods for PDE

Course materials

Lectures will usually follow one of the following references:

1. *Scientific Computing: An Introductory Survey*, M.T. Heath (see also the associated and modules at http://web.engr.illinois.edu/ heath/scicomp/notes/)
2. *Numerical Linear Algebra*, L.N. Trefethen and D. Bau
3. *Numerical Recipes: The Art of Scientific Computing* W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery