NYU | COURANT INSTITUTE OF MATHEMATICAL SCIENCES

Honors Numerical Analysis
MATH-UA 396
Spring 2022

Course Syllabus

Lecture	Monday, Wednesday 2:00PM - 3:15PM, CIWW 512
Recitation	Friday 2:00PM - 3:15PM, CIWW 512
Instructor	Mike O'Neil CIWW 1119 and 2 MTC 854 cims.nyu.edu/~oneil oneil@cims.nyu.edu 212-998-3125
Teaching assistant	Tristan Goodwill tg1644@nyu.edu
Office hours	By appointment, CIWW 1119
Course website	cims.nyu.edu/~oneil/hna22

Prerequisites

- Grade of A- in MATH 123 Calculus III (or equivalent) or grade of B+ in MATH 129 Honors Calculus III.
- Grade of A- in MATH 140 Linear Algebra (or equivalent) or grade of B+ in MATH 148 Honors Linear Algebra.
- Programming experience strongly recommended (e.g. julia, Matlab, or numpy), but not required (there is a programming component to this course).

Description

"Numerical analysis is the study of algorithms for the problems of continuous mathematics" - L. N. Trefethen, 1992. This course will cover the analysis of numerical algorithms which are ubiquitously used to solve problems throughout mathematics, physics, engineering, finance, and the life sciences. In particular, we will analyze algorithms for solving nonlinear equations; optimization; finding eigenvalues/eigenvectors of matrices; computing matrix factorizations and performing linear regressions; function interpolation, approximation, and integration; basic signal processing using the Fast Fourier Transform; Monte Carlo simulation. Students are expected to have strong foundations in multivariate calculus and linear algebra, as well as be able to write numerical codes. An introduction to programming will be provided as it is an integral part of numerical analysis, but students should feel quite comfortable programming on their own (or be exceptionally willing to learn along the way).

Note: This is an honors version of MATH 252 Numerical Analysis. There will be more required numerical programming than in non-honors sections, and more advanced theoretical topics will be covered.

Learning Objectives

At the end of this course, students will be able to properly implement numerical algorithms in the julia, Matlab, or python programming languages for solving many problems of continuous mathematics, including but not limited to: optimization, nonlinear root finding, eigenvalue/eigenvector calculation, linear system solves, integration, function approximation and interpolation, basic signal processing with the Fast Fourier Transform, and random variable generation. In each of these cases, students will be able to derive the computational complexity of the underlying numerical algorithm and its stability properties (i.e. how accurate it is, and how sensitive to errors it is). Furthermore, students will gain the judgment to know what type of algorithm is most appropriate in different situations.

Materials

The following texts are recommended references for the course:

- Burden, Faires, and Burden, *Numerical Analysis*, Cengage, 2015
- Greenbaum and Chartier, *Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms*, Princeton, 2012
- Suli and Mayers, *An Introduction to Numerical Analysis*, Cambridge, 2003
- Driscoll and Braun, *Fundamentals of Numerical Computing*, SIAM, 2017

Material from the above texts will be supplemented with lecture notes and other materials, as necessary.

Assignments

There will be a mix of homework assignments, a midterm exam, and a final exam. The homework assignments during the semester will consist of both written and computational (computer programming) work.

Grading

The overall course grade will be determined from a final numerical weighted average. The following breakdown will be used to compute an overall numerical grade:

30%	Homework (6 assignments, 5% each)
30%	Midterm
40%	Final exam (cumulative)

Homework sets will be distributed roughly every other week, and will consist of problems requiring written work, as well as those requiring a fair amount of computer programming. The midterm and final exam will be written exams and will not require any programming.

Weekly schedule

Each week, there will be two lectures plus a recitation section. The lectures will detail various algorithms and computational techniques for solving continuous mathematics problems, and may include brief programming demonstrations. The recitation section will further enforce the concepts presented in the lecture by working through example problems, as well as spending more time on specific computational implementations of the algorithms. Below is a week-by-week schedule:

1. Intro to numerical computing, floating point arithmetic
2. Nonlinear root finding
3. Solving linear systems (Gaussian elimination, LU, Cholesky)
4. Matrix/vector norms; multivariate Newton's method
5. Optimization
6. Least squares, regression, singular value decomposition
7. *Midterm*
8. Finding eigenvalues and eigenvectors
9. Interpolation

10. Function approximation, orthogonal polynomials
11. Numerical integration
12. The Fast Fourier Transform (FFT)
13. Applications of the FFT: filtering, integration, etc.
14. Monte Carlo methods
15. Optional topics (e.g. randomized linear algebra, ODEs)

Disability Disclosure Statement
Academic accommodations are available for students with disabilities. The Moses Center website is www.nyu.edu/csd. Please contact the Moses Center for Student Accessibility (212-998-4980 or mosescsd@nyu.edu) for further information. Students who are requesting academic accommodations are advised to reach out to the Moses Center as early as possible in the semester for assistance.

Academic Integrity, Plagiarism, and Cheating
Academic integrity means that the work you submit is original. Obviously, bringing answers into an examination or copying all or part of a paper straight from a book, the Internet, or a fellow student is a violation of this principle. But there are other forms of cheating or plagiarizing which are just as serious — for example, presenting an oral report drawn without attribution from other sources (oral or written); writing a sentence or paragraph which, despite being in different words, expresses someone else's idea(s) without a reference to the source of the idea(s); or submitting essentially the same paper in two different courses (unless both instructors have given their permission in advance). Receiving or giving help on a take-home paper, examination, or quiz is also cheating, unless expressly permitted by the instructor (as in collaborative projects).
(Above is adapted from the website of the College of Arts & Science: https://cas.nyu.edu/content/nyu-as/cas/academic-integrity.html)