NYU | COURANT INSTITUTE OF MATHEMATICAL SCIENCES

Honors Numerical Analysis
MATH-UA 396
4 Credits
Spring 2024

Course Syllabus

Lecture	Monday, Wednesday 2:00PM - 3:15PM, CIWW 201
Recitation	Friday 11:00AM - 12:15PM, 194 Mercer 307
Instructor	Mike O'Neil CIWW 1119 and 2 MTC 854 cims.nyu.edu/~oneil oneil@cims.nyu.edu 212-998-3125
Teaching assistant	Alexandre Milewski am10315@nyu.edu
Office hours	Monday, Thursday 11:00AM - 12:00PM, CIWW 1119
Course website	cims.nyu.edu/~oneil/hna23

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), at the level of MATH 144, CSCI 002, or CSCI 003, but not required. Note: There is a substantial 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 digital signal processing using the Fast Fourier Transform; random sampling and 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. Programming assignments will be performed in julia, python, or Matlab (at the discretion of the course instructor). 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. In addition to a slightly different list of topics, there will be more required numerical programming than in non-honors sections. Material will be presented in a more mathematically mature format, and the course will progress more quickly than the non-honors version.

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:

- Driscoll and Braun, *Fundamentals of Numerical Computation*, SIAM, 2017
- Suli and Mayers, *An Introduction to Numerical Analysis*, Cambridge, 2003
- Trefethen and Bau, *Numerical Linear Algebra*, SIAM, 1997
- Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*, SIAM, 2019
- Briggs and Henson, *The DFT: An Owner's Manual for the Discrete Fourier Transform*, SIAM, 1995

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

Students will also require access to a numerical programming environment in julia.

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, will generally cover the previous two weeks of material, and are assigned roughly every other week to provide flexibility in their completion and to accommodate their longer format. More details regarding homework and exams will be provided in-class and/or via Brightspace.

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 several problems requiring written work, as well as several problems requiring a fair amount of computer programming. The midterm and final exam will be written exams and will not require any programming. There is no extra credit.

The overall numerical grade will be converted to a letter grade with equivalencies:

100% - 95%	A
95% - 90%	A-
90% - 86%	B+
86% - 83%	B
83% - 80%	B-
80% - 75%	C+
75% - 70%	C
70% - 60%	D
60% - 0%	F

The final letter grade may be curved depending on the overall distribution of grades in the course, but only so as to increase the corresponding letter grade.

Policies

- There will not be makeup homework assignments or exams without prior approval of the instructor. Late homework is not accepted without prior approval of the instructor. Exceptions will be only made in extraordinary circumstances (e.g. illness, emergencies, etc.). Official documentation may be required for some absences.
- While attendance/participation does not factor into the overall grade in the course, attendance and class discussions will likely be crucial to the success of students in the course.
- Collaboration and discussion is strongly encouraged on homework assignments, but each student must write-up and turn in their own work, including programming components.

Weekly schedule
Each week, there will be two lectures plus a recitation section. The lectures will detail various algorithms and computational techniques, 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. More details of specific references for each topic (e.g. sections from a textbook) will be posted to the course webpage or to Brightspace.

Below is a week-by-week schedule assuming two meetings per week, with relevant sections from the reference texts listed in italics:

1. Intro to numerical computing, floating point arithmetic *(D&B 1)*
    a. Binary arithmetic
    b. Relative/absolute precision
    c. Condition numbers
2. Nonlinear root finding *(D&B 4.1-4.4, S&M 1.4-1.7)*
    a. Bisection and secant methods
    b. Newton's method
3. Solving linear systems (Gaussian elimination, LU, Cholesky) *(D&B 2.1-2.5)*
    a. LU factorization
    b. Cholesky factorization
    c. Computational complexity
4. Matrix/vector norms; multivariate Newton's method *(D&B 2.7, 4.5)*
    a. Induced matrix norms
    b. Multivariate series expansions
    c. Solving systems of nonlinear equations
5. Optimization *(D&B 4.5-4.7)*
    a. Newton methods, Hessians
    b. Quasi-Newton methods, e.g. BFGS
6. Least squares, regression, singular value decomposition *(D&B 3.1-3.4, T&B 6-11)*
    a. Gram-Schmidt orthogonalization, modified Gram-Schmidt
    b. Household reflections
    c. Golub-Kahan for computing the SVD
7. *Midterm*
8. Finding eigenvalues and eigenvectors *(S&M 5.1-5.8)*
    a. Power method, Inverse power method with shifts
    b. Jacobi's Method, QR Algorithm
9. Interpolation *(S&M 6.1-6.3, T 1-5)*
    a. Lagrange interpolation
    b. Barycentric formulas
    c. Chebyshev interpolation, Runge's phenomenon
10. Function approximation, orthogonal polynomials *(T 6-10, 13, 17, D&B 9.1-9.4)*
    a. L1 and L2 function approximation
    b. Interpolation vs. approximation
    c. Polynomial recurrences
11. Numerical integration *(S&M 7.1-7.6)*
    a. Trapezoidal rule, Newton-Cotes formulas
    b. Euler-Maclaurin summation
    c. Gaussian quadrature
    d. Adaptive schemes
12. The Fast Fourier Transform (FFT) *(B&H 1, 2.1-2.6, 10.1-10.2)*
    a. The discrete Fourier transform (DFT)

    b. Discrete orthogonality
    c. The FFT algorithm, complexity
13. Applications of the FFT: filtering, integration, etc. *(B&H 3.1-3.3, T 19)*
    a. Convolutions via FFT
    b. Integration, Clenshaw-Curtis quadrature
    c. Digital filters
14. Monte Carlo methods
    a. Probability densities
    b. Uniform sampling, pseudo-random number generators
    c. Inverse CDF sampling
    d. Transformations
15. Optional optics, time permitting
    a. Ex: Basic numerical differential equations
    b. Ex: Modern randomized linear algebra
    c. Ex: TBD

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)

Student Wellness
In a large, complex community like NYU, it is vital to reach out to others, particularly those who are isolated or engaged in self-destructive activities. Student wellness is the responsibility of all of us.
https://cas.nyu.edu/content/nyu-as/cas/academic-programs/student-wellness.html
The NYU Wellness Exchange is the constellation of NYU's programs and services designed to address the overall health and mental health needs of its students. Students can access this service 24 hours a day, seven days a week by emailing wellness.exchange@nyu.edu or calling (212) 443-9999. Students can call the Wellness Exchange hotline (212-443-9999) or the NYU Counseling Service (212-998-4780) to make an appointment for Single Session, Short-term, or Group counseling sessions.
https://www.nyu.edu/students/health-and-wellness/wellness-exchange.html