Applied Mathematics 205

Advanced Scientific Computing: Numerical Methods

Course logistics

Instructor: Chris H. Rycroft (`chr@seas.harvard.edu`)

Aim of the course

- Scientific computation has become an indispensable tool in many areas of research and in industry. This course examines the mathematics behind well-established algorithms and explore their use through examples from many disciplines.
- Provides a broad introduction to many different numerical algorithms (*e.g.* optimization, data fitting, numerical solution of differential equations)
- After taking the course, you should be able to design and analyze numerical algorithms and more effectively use scientific software

Logistics

Course website:
http://courses.seas.harvard.edu/courses/am205/

CHR office hours: Tuesday 2:30 PM–3:30 PM ET (Zoom)
Monday 1 PM–2 PM ET
(SEC, IACS conference room)

TFs: Jovana Andrejevic (jandrejevic@g.harvard.edu)
Xiaoxiao Ding (xiaoxiaoding@g.harvard.edu)
Michael Emanuel (mse999@g.harvard.edu)
Danyun He (danyunhe@g.harvard.edu)
Yue Sun (yuesun@g.harvard.edu)

Course instruction

Lectures: Monday/Wednesday 11:15am–12:30pm in SEC LL2.224

There is also a sequence of YouTube videos of the course material:
https://www.youtube.com/playlist?list=PL43IQ71lgJytIqhiJ6v5lNswFKeQ9952K

We will use Ed Discussion[1] for questions and discussion, which is integrated into the Canvas site:
https://canvas.harvard.edu/courses/92929

[1] This is new for 2021. Previous years used Piazza, but Harvard does not have a site license. Ed Discussion has similar functionality and is integrated into Canvas.

Group activities

Throughout the semester, we will offer a variety of workshops, where students have to work together in groups of 1–3 on a short assignment.

Types of group activities include:

- Programming workshops
- Open-ended problems
- Discussions of papers from scientific computing literature
- Final project presentations

Group activities: examples[2]

- Introduction to Python
- Linux terminal wizardry
- Using data fitting to image comet NEOWISE
- Differential–algebraic integrators
- Introduction to Kalman filters
- The fast Fourier transform
- Further optimization methods
- Implement a one-dimensional multigrid algorithm

[2] The complete list will depend on total enrollment.

Prerequisites

- Calculus
- Linear algebra
- Course will touch on PDEs, but no detailed knowledge required (*i.e.* you don't need to have taken a PDE course)
- Some programming experience

Programming languages

Python will be used for the in-class demonstrations. Why Python?

- Freely available, widely used, and versatile
- Interpreted language, good for small tasks without the need for compilation
- Good linear algebra support via NumPy and SciPy extensions[3]
- Good visualization support via Matplotlib[4]

[3] http://www.numpy.org and http://www.scipy.org/

[4] http://www.matplotlib.org

Programming languages

There are many other languages that are widely used for scientific computing:

Interpreted languages: MATLAB, Julia, Perl, GNU Octave

Compiled languages: Fortran, C/C++

You can complete the assignments in any language of your choice, as long as it is easy for the teaching staff to run your code—for languages not listed here, please check with the teaching staff.

Note that MATLAB shares many similarities with Python, and many numerical functions have identical names (*e.g.* `linspace`, `eig`), making it easy to follow the in-class examples.

Programming languages: assignment 0

Assignment 0 is posted on the website.

Assignment 0 provides some problems to indicate the expected level of programming familiarity for the outset of the course.

Asssignment 0 is not assessed, but it should either:

- confirm that you are already sufficiently familiar with Python (or MATLAB, C++, *etc.*)
- indicate that you need to get some programming assistance

Also, contact me or TFs regarding programming questions (Ed Discussion is useful for these types of questions).

Syllabus (part 1)

0. **Overview of Scientific Computing**
1. **Data Fitting**
   - 1.1 Polynomial interpolation
   - 1.2 Linear least squares fitting
   - 1.3 Nonlinear least squares
2. **Numerical Linear Algebra**
   - 2.1 LU and Cholesky factorizations
   - 2.2 QR factorization, singular value decomposition
3. **Numerical Calculus and Differential Equations**
   - 3.1 Numerical differentiation, numerical integration
   - 3.2 ODEs, forward/backward Euler, Runge–Kutta schemes
   - 3.3 Lax equivalence theorem, stability regions for ODE solvers
   - 3.4 Boundary value problems, PDEs, finite difference method

Syllabus

4. **Nonlinear Equations and Optimization**
    - 4.1 Root finding, univariate and multivariate cases
    - 4.2 Necessary conditions for optimality
    - 4.3 Survey of optimization algorithms
5. **Eigenvalue problems and iterative methods**
    - 5.1 Power method, inverse iteration
    - 5.2 QR algorithm
    - 5.3 Multigrid method
    - 5.4 Lanczos algorithm, Arnoldi algorithm

(Similar to previous years by Chris Rycroft, David Knezevic, and Efthimios Kaxiras, with minor adjustments. Notes from 2013–2020 still online.)

Assessment

- 62% – Five homework assignments with equal weighting
- 6% – Group activities
- 32% – Final project

Assessment: homework

The focus of the homework assignments will be on the mathematical theory, but will involve significant programming.

Homework will be due on Tuesdays and Thursdays – submit a written report and source code via the Harvard Canvas page (linked from main website). The written report should be individually submitted as either a PDF or Word document.[5]

Late homework will be evaluated on a case-by-case basis.

Deadlines are Thu Sep 23, Tue Oct 12, Thu Oct 28, Thu Nov 11, Thu Dec 2

[5] If you write your homework in a Jupyter Notebook, please upload a PDF conversion of it to Canvas.

Writing style and LaTeX

Assignments will be written in LaTeX, which is an excellent platform for writing scientific documents and equations. While completely optional, we encourage you to try and use LaTeX for your assignments. The teaching staff are all happy to talk further.

LaTeX is free and available on all major computing platforms. Using LaTeX involves writing a file in a simple markup language, which is then compiled into a PDF or PostScript file. See the excellent *Not So Short Introduction to LaTeX* 2_ϵ[6] for more information.

The original LaTeX files for the homework assignments will be posted to the website for reference.

[6] https://tobi.oetiker.ch/lshort/lshort.pdf

Assessment: code for homework

Code should be written clearly and commented thoroughly.

In-class examples will try to adhere to this standard.

The TFs should be able to easily run your code and reproduce your figures.

Academic integrity policy

Discussion and the exchange of ideas are essential to doing academic work. For assignments in this course, you are encouraged to consult with your classmates as you work on problem sets. However, after discussions with peers, make sure that you can work through the problem yourself and ensure that any answers you submit for evaluation are the result of your own efforts.

In addition, you must cite any books, articles, websites, lectures, *etc.* that have helped you with your work using appropriate citation practices. Similarly, you must list the names of students with whom you have collaborated on problem sets.

Using homework solutions from previous years is forbidden.

Assessment: group activities

- Worth 6% of overall grade
- Each worth 1.5% and graded on participation only. Can attend any number, but at least four for full credit.
- Submit 1–2 page report in groups of 1–3 students
- Three tries to submit; TFs can reject and provide feedback if incorrect or insufficient.
- If you do $n > 4$ activities, you receive an $(n-4)\%$ credit, and the final project grade percentage is reduced by $(n-4)\%$.

Assessment: final project

The goal of this course is to get you to be a responsible, productive user of numerical algorithms for real-world applications

The best way to demonstrate this is in your final project, worth 32%, to be completed in a group of two or three students[7]

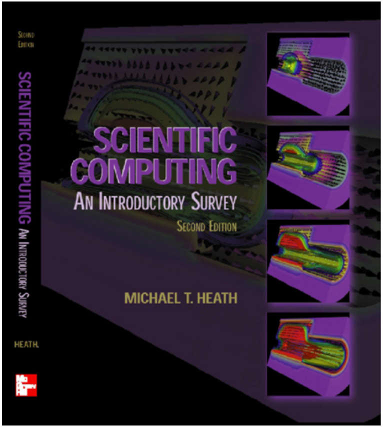
- Use concepts/methods related to the course to solve a problem of interest to your group
- Project proposal in the form of an oral meeting with the teaching staff due by 5 PM ET on Wednesday November 17th
- Project due during final exam period[8]
- Submit a report and associated code
- Option to give a project presentation (counts as two group activities)

[7] Single person or $n \geq 4$ projects may be allowed with instructor permission.

[8] Exact date TBA—depends on final exam schedule.

Textbooks

Most relevant textbook is *Scientific Computing: An Introductory Survey* by Michael T. Heath



Textbooks

- A. Greenbaum and T. P. Chartier. *Numerical Methods: Design, Analysis and Computer Implementation of Algorithms.* Princeton University Press, 2012.
- C. Moler. Numerical Computing with MATLAB. SIAM, 2004.
- L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.
- W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 2007.
- L. R. Scott. *Numerical Analysis.* Princeton University Press, 2011.
- E. Suli, D. F. Mayers. *An Introduction to Numerical Analysis.* Cambridge University Press, 2003.
- J. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.