MENG 23100 / MENG 33100
Applied Numerical Methods in Molecular Engineering
Winter 2019

Instructor:	Prof. A.L. Ferguson 389 ERC andrewferguson@uchicago.edu	**TA:**	Yutao Ma 118/218 ACC yma3@uchicago.edu

CRN: MENG 23100 / MENG 33100 (100 units)

Lecture: Kersten Phys Teach Center 101 ◆ 3:00–4:20 pm ◆ Mon, Wed

Discussion: Kersten Phys Teach Center 101 ◆ 8:30–9:20 am ◆ Wed

Dates: 01/07/2019 – 03/23/2019

Course Summary

This course will provide training in fundamental numerical methods to solve problems in molecular engineering, including: linear algebra, interpolation, differentiation, quadrature, integration, optimization, spectral methods, regression, classification, and deep learning. The approaches illustrated in applications to multi-scale engineering problems spanning the atomistic to continuum. Fundamental concepts of statistical thermodynamics, transport phenomena, quantum mechanics, and applied mathematics will be revisited. Course material will be delivered through a combination of formal lectures to expose the underlying mathematics and principles underlying each technique to confer an understanding of their applicability and limitations, and hands-on immersive praxis through project-based learning to give students familiarity and confidence with the tools. The course will provide training in the Python 3 programming language and its implementation in numerical solution techniques.

Prerequisites

MATH 20000-20100 or PHYS 22000-22100
CHEM 11300/12300 or PHYS 13300/14300
Prior experience with Python is useful but not required

Laptop Computer

A laptop computer with a working Anaconda install of Python 3 and attendant libraries is required for this course. Laptops are available for loan by contacting the course instructor.

Required Text

None.

Recommended Texts

J. Kiusalaas *Numerical Methods in Engineering with Python* (Cambridge University Press, 2005)
→ https://catalog.lib.uchicago.edu/vufind/Record/8209396
M. Newman *Computational Physics* (2012)
→ https://www.amazon.com/Computational-Physics-Mark-Newman/dp/1480145513

Attendance

Lecture: Lectures will be split between (i) traditional instruction covering the mathematical underpinnings of the numerical tools, (ii) hands-on introduction to the methods, and (iii) immersive project-based learning. **Attendance to the lectures is expected.**

Discussion: The discussion section serves as instructor office hours and an opportunity for additional time to work on hands-on exercises and projects. This time may also be used to provide make-up lectures. **Attendance to the discussion is optional**.

Assessment

Quizzes: Short, online multiple-choice quizzes will be issued to gauge understanding and mastery of the course material. These tests are designed to provide the instructor and students with feedback on basic understanding of the mathematical and algorithmic principles underlying the numerical techniques. Quizzes will be made available online for a specified time period, and solutions will be immediately posted after the quiz closes. **Accordingly, no extensions can be granted.**

Projects: Projects associated with each topic are the primary means of instruction and assessment for the course. Students will be provided with a project brief comprising a list of learning objectives, problem statement, solution approach, and set of deliverables. Students will write Python 3 code implementing a numerical solution to the problem and produce a short report detailing their findings. The code and report will be submitted as a single Jupyter notebook that must execute under the course virtual environment. A Jupyter notebook is the only acceptable submission format. **Late submissions will not be accepted, but students with legitimate excuses should contact Prof. Ferguson in advance of the due date.**

Exams: None.

Grading

Breakdown:

Quizzes	10%
Project 1 – Intro to Python **(ungraded)**	0%
Project 2 – Linear Algebra	15%
Project 3 – Nonlinear Equations	15%
Project 4 – Quadrature and Monte Carlo Integration	15%
Project 5 – Numerical Integration of ODEs	15%
Project 6 – Numerical Integration of PDEs	15%
Project 7 – Machine Learning	15%

Letter Grades: Letter grades will be based on final aggregate student scores, with numerical cutoffs specified by the instructor. However, students with aggregate scores >95% are guaranteed *at least* an A, >85% *at least* a B, and >75% *at least* a C (i.e. cutoffs for these letter grades will not be higher than these values).

Canvas

Course announcements, materials, grades, quizzes, and projects will be posted via Canvas (https://canvas.uchicago.edu). Online quizzes and projects submitted will be submitted via this portal. It is students' responsibility to check Canvas for announcements and updates.

Software

It is a primary learning objective of the course to provide training in Python 3 as a popular and ubiquitous language of numerical computing. **All numerical assignments must be completed and submitted as Python 3 Jupyter notebooks.** It is highly recommended that regardless of any current Python installations students install the Anaconda Python 3 distribution and setup the class virtual environment following the instructions provided.

What is Python? Python is a powerful, versatile, and user-friendly programming language that has become a popular and ubiquitous standard in scientific computing and data science. It is easy to use and is equipped with numerous libraries to perform common numerical tasks.

What is the difference between Python 2 and Python 3? Python 3 was a major revision of the language, with the main changes to do with print syntax, integer division, and library support. Unfortunately, Python 3 is not fully backwards compatible with Python 2. A large body of existing code is written in Python 2, but Python 3 is the future. Accordingly, this course will work exclusively in Python 3.

What is Anaconda? Anaconda is a popular and free Python distribution that comes prepackaged with all of the commonly needed Python libraries and sets up an environment in which to manage and control the Python implementation on your system.

How do I install Anaconda Python 3? Navigate to https://www.anaconda.com/download/ and download and install the latest Python 3 release appropriate for your operating system. Windows, Linux, and MacOS X are all supported. Specific installation instructions are here: https://docs.anaconda.com/anaconda/install/

What is a Jupyter Notebook? A Jupyter Notebook is a library within Python (and other programming languages) that provides an open-source web application within which one can develop code, provide narrative comments, import and export data, and generate visualizations. In short, it provides a place to develop code within an environment that is useable, readable, reproducible, and sharable.

Plagiarism

Students are responsible for producing their own quiz answers, code to solve projects, and project reports. Collaborative interaction in small groups is encouraged, but each student must perform all calculations themselves, and write their own reports. **Plagiarism will not be tolerated, and verified incidents will result in all parties receiving a zero and formal academic sanctions.** Students are responsible for familiarizing themselves with the definition of and penalties for plagiarism in the Student Manual (https://studentmanual.uchicago.edu/Policies). Note that plagiarism includes copying another student's paper or working with another person when both submit similar papers without authorization to satisfy an individual assignment. Further details on what does and does not constitute plagiarism is available here: https://internationalaffairs.uchicago.edu/page/honest-work-and-academic-integrity-plagiarism

Course Coverage

I. Introduction to Python

Anaconda installation, Python programming, modules, Jupyter notebooks, operators, containers, basic flow control, I/O, visualization, scoping, virtual environments, libraries (numpy, scipy, matplotlib, pandas, seaborn, scikit-learn, pytorch)

II. Linear Algebra

Systems of equations, conditioning, direct methods (Gaussian elimination, LU decomposition), indirect methods (Gauss-Seidel), diagonalization (QR decomposition), eigenvalue problems

III. Nonlinear Equations: Finite Differences, Roots, and Minimization

Numerical gradients, Richardson extrapolation, relaxation, Newton's method, secant method, gradient descent, simulated annealing

IV. Quadrature and Monte Carlo Integration

Trapezoidal rule, Simpson's rule, Romberg quadrature, Monte Carlo integration

V. Numerical Integration of ODEs

Initial value problems (IVP): Euler integration, midpoint integration, Runge-Kutta methods, Verlet algorithm, 1^{st} order differential equations, nth order differential equations
Boundary value problems (BVP): shooting method (IVP + root finding)

VI. Numerical Integration of PDEs

BVP: Gauss-Seidel method
IVP: forward-time centered-space (FTCS) method, Crank-Nicolson method

VII. Machine Learning

Unsupervised, supervised, semi-supervised learning, regression vs. classification, linear vs. nonlinear learning, dimensionality reduction, feature engineering, symmetries, invariances, and equivariances, principal components analysis, diffusion maps, support vector machines, neural networks

Tentative Schedule

Class	Date	Day	Module	Topic	Due
1	Jan 7	M	I	Introduction to Python	
2	Jan 9	W	I	Introduction to Python	
3	Jan 14	M	II	Linear Algebra	
4	Jan 16	W	II	Linear Algebra	Quiz 1 Project 1 (ungraded)
	Jan 21	M		**MLK DAY – NO CLASS**	
5	Jan 23	W	II	Linear Algebra	
6	Jan 28	M	II	Linear Algebra	
7	Jan 30	W	III	Nonlin Eqns: Finite Diff, Roots, and Minimization	
8	Feb 4	M	III	Nonlin Eqns: Finite Diff, Roots, and Minimization	Quiz 2 Project 2
9	Feb 6	W	III	Nonlin Eqns: Finite Diff, Roots, and Minimization	
10	Feb 11	M	IV	Quadrature and Monte Carlo Integration	
11	Feb 13	W	IV	Quadrature and Monte Carlo Integration	Quiz 3 Project 3
12	Feb 18	M	V	Numerical Integration of ODEs	
13	Feb 20	W	V	Numerical Integration of ODEs	Quiz 4 Project 4
14	Feb 25	M	V	Numerical Integration of ODEs	
15	Feb 27	W	VI	Numerical Integration of PDEs	
16*	Mar 4	M	VI	Numerical Integration of PDEs	Quiz 5 Project 5
17	Mar 6	W	VII	Machine Learning	
18	Mar 11	M	VII	Machine Learning	Quiz 6 Project 6
19	Mar 13	W	VII	Machine Learning	
	Mar 20	W			Quiz 7 Project 7

* Prof. Ferguson likely on travel these dates, appropriate arrangements TBA.