MENG 25500 / MENG 35500
Classical Molecular and Materials Modeling
Winter 2023

Instructor: Prof. A.L. Ferguson
389 ERC
andrewferguson@uchicago.edu

TA: Kirill Shmilovich
118 ACC
kirills@uchicago.edu

CRN: MENG 25500 / MENG 35500 (100 units)

Lecture: Ryerson Phys Lab 255 ◆ 9:30–10:50 CST ◆ Tue, Thu

Discussion: ERC 381 ◆ 17:00-18:00 CST ◆ Thu – Instructor office hours
KPTC 101 ◆ 9:30–10:20 CST ◆ Fri – TA office hours

Dates: 01/03/2023 – 03/11/2023

Course Summary

This course will introduce students to the methods of molecular modeling. The topics covered will include an introduction to the origin of molecular forces, a brief introduction to statistical mechanics and ensemble methods, and an introduction to molecular dynamics and Monte Carlo simulations. The course will also cover elements of advanced sampling techniques, including parallel tempering, umbrella sampling, and other common biased sampling approaches. Students will also establish expertise in scientific programming in Python 3. Course work or research experience is strongly recommended in elementary programming and physical chemistry or thermodynamics.

Prerequisites

- [**REQUIRED**] MENG 21400 – Molecular Engineering Thermodynamics
  or CHEM 26200 – Thermodynamics
  or PHYS 27900 – Statistical and Thermal Physics
- [**REQUIRED**] MATH 20100 – Mathematical Methods for Physical Sciences II
  or PHYS 22100 – Mathematical Methods in Physics
- [**RECOMMENDED**] Prior experience in elementary scientific programming

Required Text

None.

Recommended Texts

Classical Molecular Simulation:

D. Frenkel and B. Smit *Understanding Molecular Simulation: From algorithms to applications* (Academic Press, 2002)
→ https://catalog.lib.uchicago.edu/vufind/Record/11155428

M.P. Allen and D.J. Tildesley *Computer Simulation of Liquids* (Oxford University Press, 1989)
→ https://catalog.lib.uchicago.edu/vufind/Record/965300

Scientific programming in Python 3:

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) immersive project-based learning. **Attendance to the lectures is expected.**

Discussion: The discussion section serves as TA office hours and an opportunity for additional time to work on hands-on exercises and projects. This time may occasionally be used to provide make-up lectures.

Assessment

Quizzes: Short, online multiple-choice quizzes will be issued to gauge understanding and mastery of the mathematical and algorithmic principles underlying the numerical techniques. Quizzes will be available online for a specified time period, and solutions posted after the quiz closes. **Accordingly, no extensions can be granted.**

Topic Projects: Projects associated with each topic are a 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. **Late submissions will not be accepted, but students with legitimate requests for extensions should contact Prof. Ferguson well in advance of the due date.**

Term Project: Students will design and execute a term project to code up and implement an advanced molecular simulation technique in Python. The topic projects will provide a foundational code base upon which to construct the advanced simulation implementation. Students may select from a list of potential project topics or select their own. Graduate students will complete projects individually; undergraduate students will complete projects in small teams. All project topics must be approved by the course instructor. There are three deliverables associated with the term project:

(i) 10-minute presentation (posted to YouTube) + slide deck (pdf)
– intro/motivation ~2 slides, method ~3 slides, preliminary results ~3 slides
– all team members must participate in the presentation
(ii) code implementing and demonstrating the advanced simulation technique (ipynb)
(iii) written report on theory, implementation, and demonstration (pdf)
– ~5 pages (or more) single column, single spaced, excl. figures

Exams: None.

Grading

Breakdown:

Quizzes		5%
Project 1 –	Intro to Python **(ungraded)**	0%
Project 2 –	Elements of Molecular Simulation	12.5%
Project 3A –	Monte Carlo I	12.5%
Project 3B –	Monte Carlo II	12.5%
Project 4A –	Molecular Dynamics I	12.5%
Project 4B –	Molecular Dynamics II	12.5%
Project 5 –	Molecular Dynamics in LAMMPS	12.5%
Term Project		20%

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 will be submitted via this portal. It is students' responsibility to check Canvas for announcements and updates. Canvas is best viewed using Chrome or Firefox web browsers; Safari is known to cause problems.

Slack

We will be using Slack as a virtual and public forum for questions discussion. The system is allows for fast responses from the instructor, TA, and classmates, and allows students to see previous questions and answers. Rather than emailing questions to the teaching staff or using Canvas chat, please post your questions publicly on Slack. If you have not already been automatically enrolled, please sign up via https://meng255003550-awe4935.slack.com

Plagiarism

Students are responsible for producing their own quiz answers, code to solve projects, and project reports. Collaborative small group interactions are 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 awareness of the definition and penalties for plagiarism in the Student Manual (https://studentmanual.uchicago.edu/academic-policies/academic-honesty-plagiarism/). Further details on what does and does not constitute plagiarism is available here: https://internationalaffairs.uchicago.edu/page/honest-work-and-academic-integrity-plagiarism

Disability Statement

University of Chicago is committed to ensuring equitable access to our academic programs and services. Students with disabilities who have been approved for the use of academic accommodations by Student Disability Services (https://disabilities.uchicago.edu) and need a reasonable accommodation to participate fully in this course should follow the procedures established by SDS. Timely notifications are required in order to ensure that your accommodations can be implemented. Please set up a time to meet with the instructor to discuss your access needs in this course after you have completed the SDS procedures for requesting accommodations.

Equity, Diversity, and Inclusion Statement

The Pritzker School of Molecular Engineering advances the mission of translating advances in basic physics, chemistry, biology, and computation into new tools to address important societal problems and to create a research and teaching environment that enhances and transmits these capabilities to future generations. The school affirms a commitment to equity, diversity, respect, and inclusion, and aims for broad representation, accountability, and

participation among our faculty, other academic appointees, research and administrative staff, and students across age, gender, race, nationality, ethnicity, socioeconomic status, sexual orientation, ability and disability, religion, belief, and backgrounds. For resources for the PME community, as well as a how-to guide on reporting bias, please visit https://pme.uchicago.edu/equity-diversity-inclusion.

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.

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. **Programming 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.

Course Coverage

I. Introduction to Python 3

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. Elements of Molecular Simulation

Variables, ensembles, dimensions and units, boundary conditions, initialization, statistical uncertainty, ergodicity, mechanical and thermal observables, thermodynamic averages and fluctuations, dynamical equilibrium and detailed balance

III. Monte-Carlo Simulation

Multidimensional integrals, stochastic and deterministic algorithms, variance and scaling, random numbers, boundary conditions, detailed balance, acceptance probabilities, importance sampling, trial moves, convergence, interaction potentials, Metropolis-Hastings algorithm

IV. Molecular Dynamics Simulation

Lagrangian and Hamiltonian formulations, Integrators, reversibility, boundary conditions, force fields, thermostats, barostats, electrostatics, dynamical observables, transport properties and time-correlation functions

V. Molecular Dynamics Simulation in LAMMPS

Tutorial and practice using the LAMMPS molecular dynamics simulation suite

VI. Advanced Simulation Techniques

C*hain molecules, configurational bias, parallel tempering, pruned-enriched Rosenbluth, expanded ensembles, Wang-Landau density of states, Gibbs ensemble, histogram reweighting, umbrella sampling, metadynamics, committor probabilities*

Tentative Schedule

Class	Date	Day	Module	Topic	Due
1	Jan 3	T	I	Introduction to Python	
2	Jan 5	R	I	Introduction to Python	
	Jan 6	F			Quiz 1 Project 1 (ungraded)
3	Jan 10	T	II	Elements of Mol Sim	
4	Jan 12	R	II	Elements of Mol Sim	
	Jan 13	F			–
5	Jan 17	T	III	Monte-Carlo	
6	Jan 19	R	III	Monte-Carlo	
	Jan 20	F			Quiz 2 Project 2
7	Jan 24	T	III	Monte-Carlo	
8	Jan 26	R	III	Monte-Carlo	
	Jan 27	F			Quiz 3 Project 3A
9	Jan 31	T	IV	Molecular Dynamics	
10	Feb 2	R	IV	Molecular Dynamics	
	Feb 3	F			Project 3B
11	Feb 7	T	IV	Molecular Dynamics	
12	Feb 9	R	IV	Molecular Dynamics	
	Feb 10	F			Quiz 4 Project 4A Term Project Topic (+ Team)
13	Feb 14	T	V	LAMMPS	
14	Feb 16	R	V	LAMMPS	
	Feb 17	F			Project 4B

15	Feb 21	T	V	LAMMPS	
16	Feb 23	R	VI	Advanced Techniques	
	Feb 24	F			Project 5
17	Feb 28*	T	VI	Advanced Techniques – TA Kirill Shmilovich guest	
18	Mar 2	R	VI	Advanced Techniques	
	Mar 3	F			Term Project Video + Slides
	Mar 10	F			Term Project Code + Report

* Prof. Ferguson likely on travel these dates and not available for class / discussion, appropriate arrangements TBA.