CHE 210D: Principles of Modern Molecular Simulation Methods, F19

UCSB Department of Chemical Engineering

meeting times:	TR 11:00am-12:15pm, Engineering II 3301
web page:	www.engr.ucsb.edu/~shell/che210d/

Course description

The goal of this course is to equip students of a variety of research interests with the basic skills necessary to design and carry out molecular simulations: (1) formulation of both atomistically detailed and simplified molecular models, (2) basic and advanced algorithms for computing thermodynamic and kinetic behavior, (3) modern analysis techniques and visualization packages, (4) physical intuition for developing and interpreting new simulation "experiments", and (5) knowledge of computational issues and methods for improving efficiency.

This course focuses more on concepts, algorithms, and tools than on specific programming styles and languages, although enrolling students should have had at least minimal exposure to coding or mathematical software (e.g., have used any of Matlab, Mathematica, C, C++, Visual Basic, or Fortran). From early in the course, a strong emphasis will be placed on students performing and visualizing their own simulation projects.

Topics discussed in the course include: ab initio methods, classical semi-empirical force fields, energy minimization, molecular dynamics techniques, Monte Carlo methods, free energy algorithms, advanced sampling strategies, coarse-graining and multiscale methods, and rare events algorithms. Case studies in soft condensed matter, materials, and biophysics will be presented throughout the lecture material.

Instructor

Professor Scott Shell
Office: Engineering II 3321
Phone: (805) 893-4346
Email: shell@engineering.ucsb.edu
Office hours: TBA, or drop by if my office door is open, or by appointment

Recommended course texts (optional, but highly recommended)

Understanding Molecular Simulation: From Algorithms to Applications (2nd edition)
Daan Frenkel and Berend Smit, Academic Press (2007)

Statistical Mechanics: Theory and Molecular Simulation
Mark Tuckerman, Oxford University Press (2010)

Molecular Modelling: Principles and Applications (2nd edition)
Andrew R. Leach, Prentice-Hall (2001).

Coursework

Coursework will consist of a series of exercises in which students write their own small simulation programs based on methodologies discussed in lecture, run these, and provide analysis of results. At the end of the course, students will complete a project in which they simulate a system of interest, based on current topics in the literature or in their research, using methodologies discussed in class.

Grading

Grades will be based on 50% exercises and 50% final project. In addition to correctness, codes will be assessed on appropriate *Pythonic* implementations of routines and good use of NumPy idioms and approaches.

Software to install

Programming exercises in the course will require:

- a Python installation
- the NumPy and SciPy add-on libraries for Python
- C/C++ and Fortran compilers (e.g., the open-source gfortran compiler)
- a Python script editor.

In this course, examples will use the Python 2.7 series (latest version 2.7.2), not the Python 3 series that breaks compatibility with the earlier version of the language. You are welcome to use Python 3 for your code, however.

Python, a Python editor, plus all of the support libraries that you will need for scientific computing are conveniently combined in the Anaconda Distribution, which is available on Windows, Mac, and Linux platforms at

https://www.anaconda.com/distribution

You can download it for free. In particular, it includes a nice Python editor called Spyder that you are encouraged to explore and use.

Anaconda installs a default set of Python modules that are generally sufficient for most tasks. However, as a part of this course we will need to write Fortran code that is much faster for numerically-intense calculations. Fortunately, Fortran code can be compiled into Python-importable functions, rather automatically. To enable such a workflow, you will need to add several additional packages to your Anaconda Installation, which you can do with the Anaconda Navigator:

- gfortran, an open source Fortran compiler

For Windows machines, you will also need:

- m2w64-gcc-fortran
- m2w64-gcc-libgFortran
- libpython

The course will present Python as a particularly powerful and freely available programming platform for scientific computing and highly encourage its use.

Course outline

1) **Overview of molecular simulation methods**
capabilities, kinds of models and simulations performed, case studies

2) **Introduction to scientific computing with Python**
overview and capabilities of Python, SciPy, Numpy; basic coding tutorial, numerical tools in Python

3) **Review of probability and statistical mechanics**
continuous and discrete probability distribution functions, microstates and macrostates, statistical mechanical ensembles

4) **Quantum chemical methods (brief)**
Born-Oppenheimer approximation, hydrogen and multielectron atoms, Hartree-Fock theory and basis sets, density functional theory, ab initio MD

5) **Classical semi-empirical force fields**
bonded and nonbonded interactions, multipole expansion & van der Waals interactions, combining rules, parameterization, common forms (AMBER, CHARMM, etc), implicit solvation (PBSA and GBSA)

6) **Exploring the energy landscape and its minima**
energy landscape concepts, steepest descent minimization, conjugate gradient method, Newton and quasi-Newton methods, normal mode analysis

7) **Molecular dynamics: basic principles**
phase space and distribution functions, integrators for Newton's equations, timestep and timescale considerations, stability

8) **Computation of thermodynamic and transport properties**
equilibration and block averages, correlation times, pressure and temperature estimators, radial distribution functions, measuring transport properties

9) **Practical aspects of simulations**
simulation of bulk phases, periodic boundaries, initialization, force computation, storage, potential truncation, long-range interactions

10) **Molecular dynamics: advanced techniques**
thermostats and barostats, Langevin dynamics, constraints, multiple time steps

11) **Monte Carlo: basic principles**
canonical phase space distribution, separation of kinetic and configurational integrals, move sets, Markov processes and detailed balance, acceptance criteria, random number generation, comparison to MD

12) **Monte Carlo: other ensembles**
NPT ensemble, grand canonical ensemble, Gibbs ensemble, microcanonical ensemble

13) **Monte Carlo: advanced move sets**
biased insertions, force bias moves, hybrid MC/MD, lattice polymer moves, configurational bias

14) **Histogram-reweighting free energy techniques**
connection between histograms and ensemble probabilities, measuring histograms, potentials of mean force, reweighting, Bennet's method, Ferrenberg-Swendsen reweighting and WHAM, phase equilibria

15) **Monte Carlo: biased sampling and related free energy techniques**
basic idea and formalism, umbrella sampling, multicanonical method, stratification, Wang-Landau method, transition matrix methods

16) **Other free energy techniques**
perturbation theory, Widom particle insertion method, thermodynamic integration, absolute free energies, ideal gas and Einstein crystal reference states, alchemical transforms

17) **Advanced sampling techniques**
simulated tempering, replica exchange MD/MC and variants, extended ensembles

18) **Coarse graining and multiscale techniques**
kinds of simplified models, advantages of coarse-graining, development and parameterization of coarse-grained models, inverse MC, force-matching, relative entropy

19) **Rare event methods**
transition state theory, transition path sampling, finding transition states, computing rate coefficients & rates

20) **Introduction to molecular editing and visualization software**
viewing simulation structures and movies, making movies from simulation trajectories

Approximate Course Schedule (highly tentative)

date		**topic**	**HW**
9/26	Th	Capabilities and applications of molecular simulations; review of probability and statistical mechanical concepts	
10/1	Tu	Quantum chemical and ab initio methods (Hartree-Fock), DFT	
10/3	Th	Classical force fields Simpler models (surfaces, implicit solvation, coarse-grained)	
10/8	Tu	Exploring the energy landscape (minimization) Normal mode analysis	1
10/10	Th	TBA	
10/15	Tu	Molecular dynamics	2
10/17	Th	Visualization Computation of bulk properties	
10/22	Tu	Computation of bulk properties	
10/24	Th	Simulations of bulk phases Continuum solvation	
10/29	Tu	Advanced molecular dynamics techniques	3
10/31	Th	Monte Carlo: basic principles	
11/5	Tu	Monte Carlo: basic principles & other ensembles	
11/7	Th	Monte Carlo: other ensembles	4
11/12	Tu	TBA	
11/14	Th	TBA	
11/19	Tu	Advanced move sets	
11/21	Th	Histogram reweighting and related free energy calculations	
11/26	Tu	Histogram reweighting Biased sampling and related free energy calculations	
11/28	Th	Biased sampling and related free energy calculations	
12/3	Tu	Other methods for free energy calculations	
12/5	Th	Other methods for free energy calculations Advanced sampling techniques Coarse graining and multiscale methods	