Teaching Computational Materials Science for Nanoscale Science and Engineering

S.C. Glotzer[1a,b], M.A. Horsch[a], M.H. Lamm[a,2] , C.R. Iacovella[a], and Z.L. Zhang[a]
Department of Chemical Engineering[a] and Department of Materials Science & Engineering[b],
University of Michigan, Ann Arbor, Michigan 48109-2136

ABSTRACT

We describe the development of a graduate level course designed to teach computational materials science and its application to nanoscale science and engineering. We discuss the use of MatDL, a web-based digital library and materials science resource, as a collaborative learning tool within the context of the course.

INTRODUCTION

Computational methods are pervasive throughout materials research. Because of remarkable recent increases in computer speed, memory, and accessibility, as well as important advances in algorithms and simulation methods, materials simulation is now quantitatively as well as qualitatively predictive. As a result, computational materials science is a scientific tool on par with theory and experiment. Many experiments today are guided by predictions of materials simulations, and data are interpreted through comparison with simulated properties and behavior. Experimentalists now routinely collaborate with simulators, and many use commercial materials simulation software in their labs. For this reason, it is important that all students of materials science and engineering, and of any disciplines home to materials research, learn the fundamentals, potential, and limitations of computational materials science.

RESULTS

To teach computational materials science as it applies to nanotechnology to science and engineering students, we developed a course suitable for first year graduate and advanced undergraduate students. The course is entitled, "Computational Nanoscience of Soft Matter," and is listed as Chemical Engineering (ChE)/Materials Science and Engineering (MSE) 557 at the University of Michigan (U-M). The course is one in a sequence of computational courses taught in the MSE and ChE departments at U-M, and is open to students across the university. The aim of the course is to provide students with an understanding of the strategies, methods, capabilities and limitations of computer simulation as it pertains to the modeling and simulation of soft materials at the nanoscale. Students completing the course learn to (i) appreciate the role of computer simulation in modern research and development, and understand the relationship between theory, simulation and experiment; (ii) assess the relevance and quality of molecular and mesoscale simulation studies reported in the literature; (iii) judge which simulation models and methods are useful for describing processes and phenomena on different length and time scales; (iv) ascertain which, if any, simulation methods are relevant to their research, and, if so,

[1] sglotzer@umich.edu
[2] Present address: Department of Chemical Engineering, Iowa State University, Ames, IA

how to develop or find codes and how to use them; and (v) understand the workings and limitations of commercial molecular and mesoscale simulation software.

In teaching the methodology underlying materials simulation techniques, it is important to provide examples of the application of these techniques to real materials problems. Here, simulation methods are taught in the context of physical phenomena of soft matter, in particular as applied to current problems in nanoscience and engineering. Soft materials include polymers, liquid crystals, gels, glasses, and complex fluids such as surfactants and suspensions. In this way, students gain a deeper understanding of the behavior of soft materials and their potential application to nanotechnology.

Graduate students enrolled in the course are typically (i) doing research involving materials simulation; (ii) interested in or doing materials research and would like to understand what simulations can teach them about the materials and processes they study; (iii) interested to learn more about molecular processes in soft materials; and/or (iv) interested in nanoscience and nanoengineering. Students enrolled in the course have diverse educational backgrounds, and are engaged in masters- and PhD-level research in a variety of disciplines, including chemical, biomedical, and aerospace engineering, physics, chemistry, macromolecular science and engineering, mechanical engineering, civil and environmental engineering, nuclear engineering, and materials science and engineering. The disparity in courses taken by the students poses a challenge to teaching the course, as many disciplines do not require their students to take courses in, e.g., quantum mechanics or statistical mechanics. These two subjects provide the basis for ab initio and classical simulation methods, respectively, and as such students who have had previous exposure to those subjects get more out of the course than those who have not.

A challenge in teaching computational materials science in a one-semester, three-credit course is that the three major topics of ab initio methods, molecular simulation methods, and mesoscale simulation methods, could each easily fill an entire semester course. As a consequence, many aspects and subtleties must be omitted, and only the most central issues retained. In this course, we chose to reduce the time spent on ab initio methods so that more time may be spent on molecular and mesoscale methods. Our goal in the short two-three weeks allotted for ab initio methods is to provide the basic background to first principles methods, define key terms and "simulator jargon", and point the students toward today's most popular codes, their applications, and their limitations. In designing the curriculum we strived to adopt a modular approach, such that each module is, for the most part, independent from the other modules. After each module, the students have seen sufficient material and had sufficient hands-on experience with in-class simulation labs that they should be able to read research papers based on simulations

Figure 1. Graduate students in MSE/ChE 557 running a DPD simulation module during an in-class lab. The students are comparing equilibrium, ordered structures obtained by each team at different state points. Instructions for the lab are displayed on the overhead screens. Left to right in foreground: M.A. Horsch, K. Benjamin, Y. Gebremichael. Fall semester, 2002.

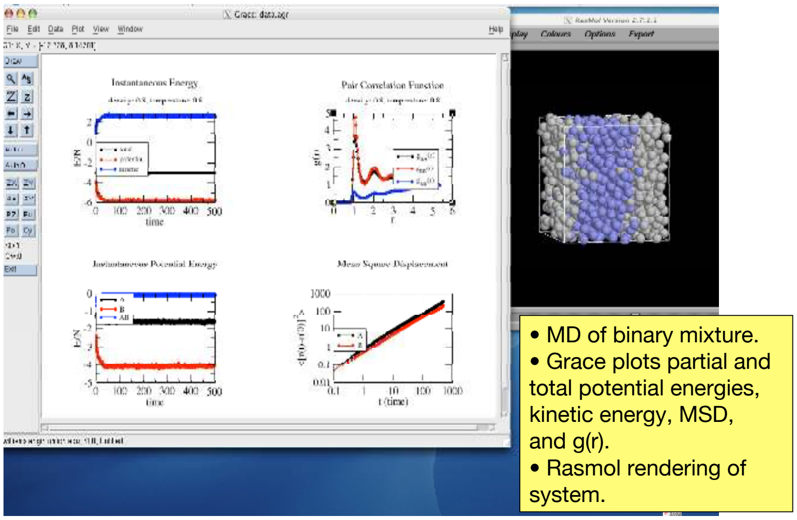
and understand the terminology and basic methodologies underlying the simulation techniques, and know what to look for in the quality of the simulations conducted.

The modules covered in this 15-week course are as follows:

1. Overview of ab initio methods for soft materials: Schrödinger's equation. Born-Oppenheimer approximation. Molecular orbital theory and density functional theory. Overview of ab initio software packages: E.g. CASTEP, Gaussian98, SIESTA.
   Examples: Electronic density calculations of organic/inorganic nanomaterials
2. Force fields for classical molecular and mesoscale simulations. Explicit atom, united atom, and coarse-grained force fields for hard and soft matter, including molecular and polymeric materials.
3. Molecular dynamics (MD) methods for soft materials. Basic method: initialization, force calculation, integration of equations of motion. Boundary conditions. Equilibration issues. Interrogating molecular simulation data: correlation functions; probing structure and dynamics. Data averaging.
   Example applications: Liquids, glasses, crystals. Nanocomposites and nanomaterials.
4. Advanced MD methods: Statistical mechanics review. MD in different ensembles. Tricks of the trade. Neighbor lists. Long-range interactions. Integration schemes. Multiple time-stepping vs. constraints. Parallel MD. Acceleration algorithms. Non-equilibrium MD. Etomica, LAMMPS, Cerius2. Car-Parinello ab initio MD.
5. Collision dynamics for hard spheres, Brownian dynamics methods.
   Example applications: Nanopatterning of block copolymers; self-assembly of polymer-tethered Buckyballs in solution.
6. Monte Carlo (MC) methods for soft materials. Basic method for off-lattice MC: Metropolis method. Detailed balance. Trial moves. Acceptance ratios. Ensembles. Parallel MC. Cluster moves. Advanced Metropolis methods. Smart and biased MC for macromolecular systems. Cluster moves. Parallel tempering. Histogram reweighting techniques. Density of states MC methods (Wang-Landau).
   Example applications: Biomolecular assembly of nanoparticles. Self-assembly of crosslinked organic/inorganic nanocomposites.
7. Dissipative particle dynamics method for complex fluids and soft materials, including fluids, colloids, and polymers. Comparison with MD and BD. Etomica, Cerius2.
   Example application: Nanopatterning of block copolymers and complex fluids
8. Field-theoretic methods for multicomponent and anisotropic soft materials. Time-dependent Ginzburg Landau methods. Parallel TDGL. Dynamic density functional theory: OCTA and Mesodyn. Application to polymer blends and liquid crystals.
   Example application: Nanopatterning of diblock copolymers and polymer blends. Nanopatterning of surfaces for molecular electronics.
9. Multiscale simulation strategies for bridging molecular to macroscopic scales.
   Example application: Self-assembly of polymer-tethered silica cubes (POSS).

The course is taught in a state-of-the-art computational classroom on U-M's North Campus. Students work in teams at 22 Mac G4 workstations to carry out simulation labs during class. In 2002 and 2003, over 35 students (and several postdocs) were enrolled in the course each year from nearly a dozen different departments. The teaming of students caused by the large number of students relative to the available computer screens appears to enhance the learning process by facilitating discussion and interaction both within and among teams (Fig. 1). Often, computationally weaker students inadvertently team with computationally stronger students, and students without prior background in, e.g. statistical mechanics or quantum mechanics team with students who'd taken such courses. This combination fortuitously sets the stage for positive mentoring experiences for all students.

Lectures are presented at the beginning of each class session via powerpoint on two overhead screens to provide the background on the simulation methods to be explored during that day's lab. The lectures typically last one hour, followed by an hour of hands-on lab activities. A typical lab consists of a series of simulations to be carried out using compiled simulation codes provided to the students. In this course, we use three types of codes. The majority of the codes used by the students in the course are developed by PhD students and postdocs in the Glotzer group by modifying their own research codes (Figs. 2 and 3). In some cases, the codes are stripped down versions of more complicated group codes – "sanitized" codes that run from a unix command line using an editable input file. Students receive instructions in class as to what each line of the input file contains and how to sensibly vary the input parameters. The codes are configured to record data in suitably labeled output files for later study by the students. They are also configured to automatically display, during run time, (i) visualizations of the data files using Data Explorer or Rasmol, and (ii) plots of key data using Grace. Each of these graphics and plotting packages are freely available, public domain codes that run under Mac OS X. We have found that the development of classroom lab modules by students and postdocs is a positive experience that engages them in the teaching process. On the days their codes are used in class, those students and postdocs participate as instructors by helping the students whenever problems or questions arise concerning the code or the materials problem they are studying with the code. Our lab modules were used in a tutorial workshop co-organized by the author on Computational Nanoscience at Oak Ridge National Lab in August 2003, and several have recently been used in a new graduate level molecular simulation course at North Carolina State University. They were also demo'd during a tutorial on Computational Nanoscience at the Fall 2003 meeting of the American Institute of Chemical Engineers.



• MD of binary mixture.
• Grace plots partial and total potential energies, kinetic energy, MSD, and g(r).
• Rasmol rendering of system.

Figure 2. Screen shot of in-class MD simulation lab module. Students run pre-compiled code. Plotting program Grace and visualization program Rasmol automatically display physical quantities (here, potential and kinetic energies, pair correlation function, mean square displacement) and system snapshots, respectively.

The second type of classroom simulation code used is provided by the Etomica molecular simulation environment developed by David Kofke at SUNY-Buffalo (www.etomica.com). Etomica is written in Java with its own windowing toolkit and employs an easy to use GUI that the user can adapt when developing a simulation module. It is designed to be extensible (accommodates easily new methods and models), computationally efficient, and easy to use. Etomica consists of two distinct parts: an Application Programming Interface (API) and an

Integrated Development Environment (IDE). The API can be used to construct a wide variety of simulations from a common set of building blocks. The IDE provides an alternative graphical environment for the development and application of complex molecular simulations. We have used several Etomica modules as demos and as lab modules to demonstrate simulation issues such as statistics, running averages, radial distribution functions for characterizing structure, periodic boundary conditions for mimicking infinite systems, square well and hard sphere potentials, Maxwell-Boltzmann velocity distributions for equilibrium systems, etc. We've also used an Etomica simulation module for dissipative particle dynamics developed last year for this course by Kofke in collaboration with Glotzer, Iacovella, and Horsch.

The third type of code we use in the course is commercial software such as Cerius2 from Accelrys or Gaussian98. Because of the limited number of site licenses usable at any one time, it has not been feasible to use these codes for in-class labs. Instead, we use these codes as demos to demonstrate the use of certain simulation techniques, such as MD, DPD, and ab initio MD, and some students use them outside of class for their term projects.

In the first year the course was offered, no mechanism was in place for saving the results of the students' simulations in a central location accessible by everyone in the course. In the second year, we used MatDL, a new web-based digital materials library (www.matdl.org). We are part of a multidisciplinary team of researchers comprised of information scientists, computer scientists, and materials scientists from Kent State University, the Materials Science and Engineering Laboratory at the National Institute of Standards and Technology, Massachusetts Institute of Technology, University of Michigan, and University of Colorado at Boulder working to construct the Materials Digital Library as part of the National Science Foundation's National Science Digital Library Program (NSDL). The NSDL is composed of over 100 digital libraries spanning the science disciplines as a premier portal of scientific information. A primary goal of this project is the development of tools to help researchers and students describe as well as share materials they contribute to MatDL. A key aspect of the project is to understand how users gather, store and evolve scientific information in the context of materials science information, in order to help develop distributed peer-to-peer systems that will facilitate knowledge to flow digitally from laboratories to the classroom. The students in ChE/MSE 557 served as a pilot user group for MatDL and used the resource to store simulation nanostructures and share them online with their classmates [1,2].

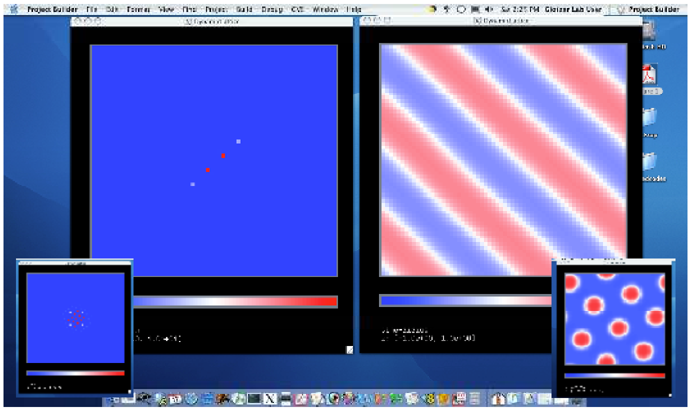


Figure 3. Screenshot of time-dependent Ginzburg-Landau simulation of microphase ordering in a block copolymer melt, showing both the structure factor S(q) and real space concentration profile for two different state points. Images are rendered using Dynamic Lattice from Cornell.

Two supplementary texts are used in the course: *Understanding Molecular Simulation: From Algorithms to Applications* by D. Frenkel and B. Smit [3], and *Molecular Modeling: Principles and Applications* by A.R. Leach [4]. Together, these two texts provide a good coverage of most of the topics covered in the course (except field-theoretic methods and multiscale simulation). Grading in the course is based on performance on the weekly in-class labs, a final exam, and a term project. In the term project, students write a ten-page proposal to perform a detailed simulation study of a problem important to nanoscience and nanotechnology. Each student chooses a problem of interest to him or her, describes its importance and relevance, and discusses the open questions he or she wishes to address with simulation. He/she then describes the simulation methods most appropriate to study the problem and why, how they will be implemented, and what the limitations, challenges, and possibilities, pros and cons are to using these methods to address the problem(s) of interest. An estimate of the computational resources that will be required to undertake the study is also included. The students are asked to describe what will be learned from such a simulation study, and what the importance of that knowledge or understanding is in the context of the problem. The students are graded on the depth of their understanding of the problem and the simulation approach, the appropriateness of the proposed method(s) for the problem described, the thoroughness of the project description, and how convincing their proposal is. Feedback provided by the students indicated the project was extremely worthwhile, and many of the students subsequently carried out the research proposed in their term project in connection with their thesis research, which in many cases was primarily experimental.

CONCLUSION

We have described a new course on Computational Nanoscience of Soft Matter developed at the University of Michigan. This course teaches computational materials science as applied to nanoscale science and engineering of soft and hybrid materials. We described the use of MatDL for sharing classroom-generated simulation nanostructures.

ACKNOWLEDGEMENTS

We are grateful to D.A. Kofke for providing Etomica modules and for help with the development of new Etomica simulation lab modules for ChE/MSE 557, and to L. Bartolo for assistance with MatDL. We thank the many students who enthusiastically participated in the course in Fall 2002 and Fall 2003 and provided helpful feedback to improve the labs and curricula. Funding for the development of research codes that were later developed into classroom codes was provided by the National Science Foundation under grants DMR-0103399 and CTS-0210551. Development of MatDL is supported by NSF grant DUE-0333520.

REFERENCES

[1] L. Bartolo, C.S. Lowe and S.C. Glotzer, Proceedings of International Society for Knowledge Organization (ISKO), in press.
[2] L. Bartolo, S.C. Glotzer, A.C. Powell IV, D. Sadoway, and J.A. Warren, MRS Symposium BB, this proceedings.
[3] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd Edition, Academic Press, 2002.
[4] A.R. Leach, *Molecular Modeling: Principles and Applications*, 2nd Ed., Prentice Hall, 2001.