

## **Teaching the molecular simulation of materials to a diverse cross-section of engineering graduate students**

**Michael L. Falk**

**Department of Materials Science and Engineering,  
University of Michigan, Ann Arbor, MI 48109-2136**

Over the past two years the Materials Science and Engineering (MSE) department at the University of Michigan has offered a graduate level class in molecular simulation of materials. This class has attracted enrollment both from within MSE and from other engineering disciplines as well as Physics and Applied Physics. The curriculum for this class covers the modeling of molecular interactions, minimization techniques, Monte Carlo calculations and molecular dynamics simulation. The primary challenge of this class has been to teach an understanding of these methods to students with diverse backgrounds in computer programming, mechanics and statistical physics. In order to meet this challenge the class reviews fundamental concepts in each of these subject areas. Computer programming assignments play a key role in developing student understanding, but programming assignments can easily overwhelm students unfamiliar with programming. Object oriented programming techniques using C++ mitigate the complications involved in programming assignments. Providing sample codes and pre-defined classes hide the unimportant programming details while allowing access to high-level algorithm design.

### **Introduction**

Over the past two decades Materials Science and Engineering (MSE) has changed significantly due to the integration of computation into nearly every aspect of materials manufacturing, development and basic research. For example, computer simulation is increasingly used to predict cooling histories and resulting microstructures in metal alloys, to unravel the details of molecular conformations that determine the function of biological molecules, and to understand the basic science behind a variety of electronic materials processes. However, the curriculum in MSE has not kept pace with these advances in the field.

This report details a class that has been offered at the University of Michigan over the past two years. The course, entitled “Molecular Simulation of Materials” presents a subset of these simulation skills at the graduate level. The students who have enrolled in this class come from a variety of academic disciplines. Over the course of one semester these students become familiar with the theory and techniques of atomistic simulation. These techniques sit between the more standard continuum mechanics methods and significantly more detailed electronic structure calculations. Continuum methods do not address atomistic processes directly, instead subsuming these phenomena into empirical constitutive laws. Electronic structure calculations are based formally on quantum mechanical precepts, but the number of atoms and time scale accessible to these methods are correspondingly limited. Atomistic methods are ideal for investigating the cooperative behavior of collections of atoms or molecules during equilibrium and non-equilibrium processes in materials, and for modeling thermodynamically relevant statistical phenomena on the nanometer to micron scale.

### **Course content**

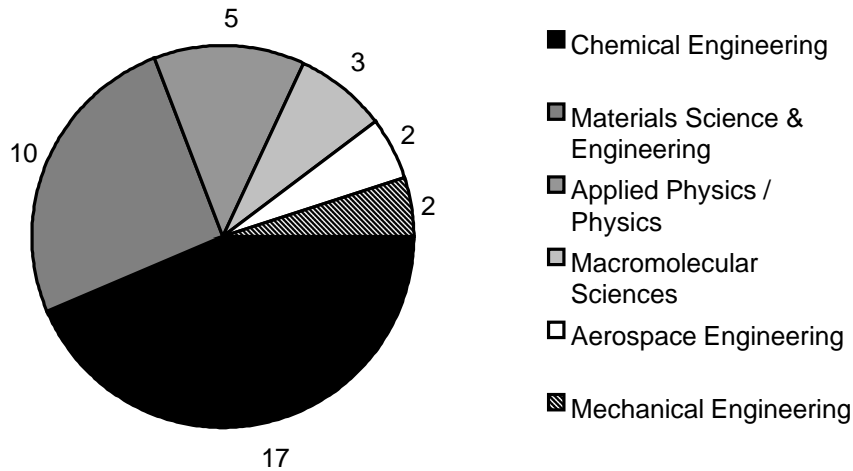
The course covers empirical theories of bonding, minimization methods applied to atomic structure, molecular dynamics simulation, Monte Carlo calculations, kinetic Monte Carlo simulation, and tight-binding models of bonding. The theoretical basis of each of these topics is covered. Students are then presented with a number of opportunities to implement these techniques both in homework assignments and a final project.

The material covered requires background in a number of disparate areas. Areas of primary importance include statistical mechanics, thermodynamics, bonding and structure of materials and Hamiltonian mechanics. Also, since the course requires students to program, it is important that students have some familiarity with programming and algorithm design. In addition, a working knowledge of elasticity, rheology and quantum mechanics can be helpful for understanding basic concepts.

### **Student enrollment**

One of the successes of this class is that it has attracted a broad cross-section of students from both MSE and related disciplines. Students from Chemical Engineering have, in fact, made up the majority of the students enrolled. In addition, the class has attracted students from Applied Physics, Physics, Macromolecular Sciences, Aerospace Engineering and Mechanical Engineering. The breakdown of the enrollment is shown in Figure 1. Two undergraduate students have enrolled and successfully completed the coursework.

### Enrollment in Molecular Simulation of Materials (Fall 2000/2001)



**Figure 1** – Enrollment over two years in a graduate level class on “Molecular Simulation of Materials” showing the range of majors participating.

This diversity of student backgrounds provides a challenge when presenting the course material. As discussed above, the course necessitates at least a working knowledge of a wide number of different subject areas. As illustrated in the table, the students entering the class typically have some subset of these skills, but may be entirely lacking in others. For example while the MSE and Macromolecular Sciences students can be expected to begin with an understanding of structure and bonding in solids, the same cannot be said of the students from Aerospace, Chemical and Mechanical Engineering. Conversely, while the Physics, Aerospace and Mechanical Engineering students typically have been exposed to the Hamiltonian formulation of mechanics, MSE and Macromolecular Sciences students will not have been introduced to these concepts. For this reason each of these topics is introduced and reviewed in the context of the class. This is primarily accomplished through three written problem sets that focus on the topics of (i) relating bonding to mechanical properties (ii) Hamiltonian mechanics and (iii) statistical mechanics. Other topics are reviewed briefly in lecture without the reinforcement of written work outside of class.

The wide disparity in the programming experience of students is the most difficult issue in teaching the class. Students who have had significant previous experience in programming are quickly able to begin applying the concepts in practice, while others find themselves learning the

	Elasticity	Fluids/ Rheology	Hamiltonian Mechanics	Quantum Mechanics	Statistical Mechanics	Structure and Bonding
Materials Science and Engineering	Dark	Light	Light	Light	Dark	Dark
Applied Physics/Physics	Light	Light	Dark	Dark	Dark	Light
Chemical Engineering	Light	Dark	Light	Light	Dark	Light
Macromolecular Sciences	Light	Dark	Light	Light	Dark	Dark
Mechanical/Aerospace Engineering	Dark	Light	Dark	Light	Light	Light

**Table** – A table showing the relevant background subject areas students are likely to have familiarity with depending on their home department. Darker boxes indicate increased chance of prior coursework in this area.

fundamentals of programming as the course progresses. To address this issue code is provided to students both as an example and to serve as a foundation on which students can build their understanding. This allows students to directly apply high-level concepts without becoming lost in the minutiae of programming.

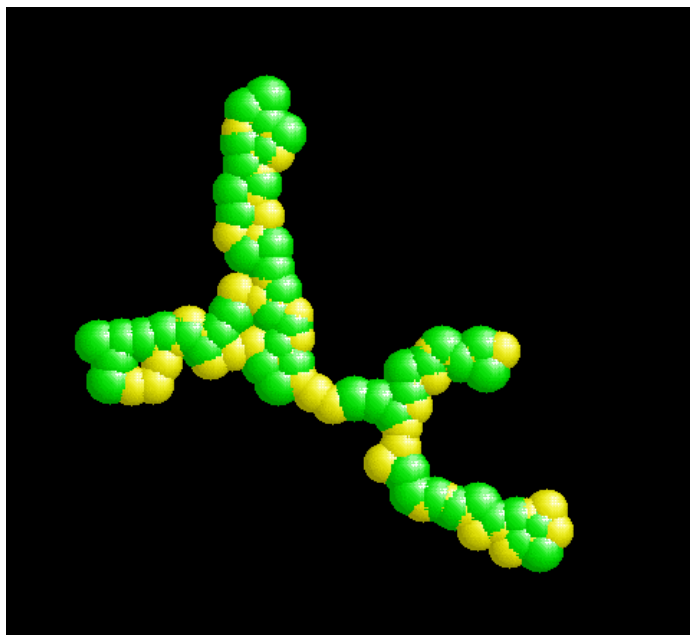
**Example programming assignment: Finding the energetic minima of a charged polymer**

The second programming assignment of the course will be used to illustrate how direct experience with simulation methodologies was provided without the need to perform extensive programming. The assignment in question considers a simple model of a polymer as a chain comprised of evenly spaced charged monomers. The interactions between monomers that are not bonded are modeled by an empirical  $1/r^2$  repulsion and an unscreened electrostatic interaction.

$$U = \sum_{\langle i,j \rangle} \left[ \frac{A}{r^2} + \frac{q_i q_j}{4\pi\epsilon_0 r} \right]$$

The students consider two different model polyelectrolytes. One is a chain of length 20 with alternating segments of 5 monomers each of charge  $+q_0$  or  $-q_0$ . The second is a chain of size 100 with alternating segments of size 10.

Students are required to consider two different algorithms for finding minimum energy configurations of these model polyelectrolytes: subsequent minimizations and simulated annealing.<sup>1,2</sup> In the implementation prescribed both methods utilize a “kink” operation that involves choosing a monomer at random and moving the monomers further along on the chain by rotating them about the chosen monomer. For example on a length 10 chain, if the 6<sup>th</sup> monomer was chosen then the 7<sup>th</sup> through 10<sup>th</sup> monomers would be rotated about an axis passing through the 6<sup>th</sup> monomer. This move preserves all bond lengths while allowing conformational changes to occur in the polymer. While such an algorithm would be unsuitable for manipulating a polymer melt where chains are very closely packed, it is suitable for exploring these minimization procedures in the context of an isolated molecule. In the direct minimization procedure monomers are chosen randomly and the molecule is carefully bent so as to minimize the energy. The simulated annealing procedure on the other hand considers a stochastic process where moves are accepted or rejected according to an imposed Boltzmann weighting such that, when the effective temperature is high, moves that raise the energy are common while at very low temperature only moves that lower the energy are likely. The temperature is lowered according to a schedule from a thermal energy on the same order as the energy scale of the potential to zero temperature. If students perform the assignment correctly they will discover that although direct minimization provides a more direct means of finding an energy minimum,



**Figure 2** – Chime images generated using the .xyz format files output by the Vector class. These animation images can be used to visualize polymer conformations that aid students in debugging their algorithms and evaluating simulation results.

this minimum is rarely a global minimum and can be significantly higher in energy than the minimum found by simulated annealing. A sample result of simulated annealing is shown in Figure 2.

In order to facilitate this programming assignment students are provided with C++ code that can be used to construct a data structure that will contain all the information about the polymer and that facilitates kinking the polymer. As an object-oriented programming language C++ facilitates the creation of procedures that can essentially function as black boxes to an end user. This is accomplished by the creation of a class and a set of operations on that class. These operations can include the redefinition of standard operations such as + (addition), \* (multiplication), = (assignment) and [] (referencing an element). In addition higher-level operations such as the kink operation mentioned above can be defined. For the purposes of this assignment a class called Vector was defined which holds the positions of a sequence of monomers and an operation called Kink was defined to facilitate the minimization operations. The header information for this class is provided in Appendix A.

```

/*****/
float picknum(float range){
// returns a random floating point number between 0 and range
    return range*(float)rand()/(float)RAND_MAX;
}
long picknum(long range){
// returns a random floating point number between 0 and range
    return (range*(long)rand())/(long)RAND_MAX;
}
/*****/
float AtomList::Anneal(long steps, float T, ofstream& file){

    long i, accept=0;
    float newE, axis[DIM]={0.0,0.0,1.0};
    Vector newpos;

    for(i=0;i<steps;i++){
        newpos=position.Kink(axis,picknum((float)(2.0*PI)),
                            1+(long)picknum((long)(position.length-2)));
        newE = Energy(newpos);
        if (newE<=energy || (newE+T*log(picknum((float)1.0)))<energy){
            position=newpos;
            energy=newE;
            accept++;
        }
    }
    return (float)accept/(float)steps;
}
/*****/

```

**Figure 3** – The simulated annealing procedure written with the use of the Vector class

With the Vector class provided, writing the procedures to perform the minimizations becomes tractable even for students with minimal programming experience. The simulated annealing procedure is shown in Figure 3 and is comprised of only 16 lines of code. In addition the Vector class allows easy output of data to freely available visualization software such as Chime by MDL Systems.<sup>3,4</sup> This software allows students to see the final and intermediate conformations being produced by their algorithms. Such visualizations facilitate the debugging of their algorithms and the evaluation of the simulation results.

## Conclusions

- Classes that introduce students to the fundamentals of materials simulation can provide an opportunity for MSE departments to reach beyond their core student constituencies to serve broader segments of the engineering research community
- Diversity of student background can be addressed through a combination of focused coursework and object-oriented programming methodologies.
- Providing computer code that can form the foundation of student work allows students to work on a high conceptual level without becoming bogged down in programming details.
- Powerful visualization tools are readily available on the internet to support such an effort.
- Class projects from the first offering of this class can be viewed at <http://msewww.engin.umich.edu:81/research/groups/falk/MSE556/>

## Appendix A – The Vector class

```
class Vector
{
public:
    float *v;           //the array of DIM*length floating point numbers
    long length;       //the number of atoms in the system

    // Constructors
    Vector() {length=0;}
    Vector(long newlen);
    Vector(const Vector& v2);

    //Destructors
    ~Vector() { delete [] v;}
```

```

// Redefinitions of the standard C++ operators
float& operator[](const long term) { return v[term];}
    // Vector[n] will return the nth item in the v array
    // this item is associated with the (n/DIM)th particle

float* operator()(const long term) { return &(v[term*DIM]);}
    // Vector(n) will return a pointer to the array starting with the
    // information for the nth particle. This can be used to pick out
    // the information for a particular atom

float& operator()(const long term, const long dim) { return v[term*DIM+dim];}
    // Vector(n,i) will return the dimension=i information for the nth particle
    // e.g. position(10,0) will return the x position of the 10th particle

Vector& operator=(const Vector& v2);
    // So you can assign one Vector to another as Vector1=Vector2

Vector operator*(const float factor);
    // So you can multiply a Vector times a scalar as Vector1=Vector2*5.0

Vector operator+(const Vector& v2);
    // So you can add two Vectors as Vector1 = Vector2+Vector3

void zero();           // sets all values of the Vector to zero
int iszero();          // returns 1 if the vector is zero, otherwise returns 0
float squared();       // computes the sum of the squares of all terms in v
float norm();          // computes the norm of the vector

Vector Kink(float *axis, float rads, long start);
    // introduces a kink in the chain by rotating all the atoms beginning
    // with start+1 around the atom at start. rads is the angle of the
    // rotation. axis is the axis about which the rotation takes place

void Print();          // print to standard output
void Print(ofstream& file); // print to a file
void Print(ofstream& file, int i); // print the ith particle to a file

};

```



## **Bibliography**

- [1] S. Kirkpatrick, C.D. Gelat and M.P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, pp. 671-680 (1983)
- [2] A.R. Leach, *Molecular Modelling: Principles and Applications*, (Harlow, England: Pearson Education Limited, 2001), Chapter 5.
- [3] T.G. Gardner, "Chemistry on the Internet: II. Communicating Chemical Images Over the Internet," *Chem. Educator*, Vol. 1, No. 2, S1430-4171(96)02029-8 (1996).
- [4] P.M. Lahti, E.J. Motyka, R.J. Lancashire, "Interactive Visualization of Infrared Spectral Data: Synergy of Computation, Visualization and Experiment for Learning Spectroscopy," *Journal of Chemical Education*, Vol. 77, pp. 649-653 (2000).

## **Biographical Information**

Michael Falk is an assistant professor in the Department of Materials Science and Engineering at the University of Michigan with a joint appointment in the Applied Physics Program. He received his Ph.D. in physics at the University of California, Santa Barbara in 1998 and was awarded the American Physical Society's Nicholas Metropolis Award for outstanding dissertation in computational physics in 2000. His research focuses on materials processes far from equilibrium including fracture and deformation in non-crystalline solids and nanometer scale pattern formation during heteroepitaxial growth.