Molecular Simulations "On the Fly"

Tuhin Ghosh[1], Pawel Keblinski[2], and Shekhar Garde[1†]

[1] *Department of Chemical Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180*

[2] *Department of Materials Science and Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180*

Interactive Web based tools possess enormous potential for being highly effective pedagogical tools. In particular, understanding of molecular-level phenomena that are hard to visualize can be made easy by employing such tools. Here we present application of Web based tools, especially for instructional purposes, and provide a brief primer on the "nuts and bolts" that go into their development. We also present representative instances of groups around the world that have contributed significantly towards making these tools available on the World Wide Web.

I. INTRODUCTION

Chemical Engineering research and education today places increasing emphasis on a molecular-level perspective towards phenomena that had been traditionally relegated to study at much larger scales. Applications where such molecular-level perspective has had direct impact range from biological systems (e.g., protein structure-function, biomolecular binding and recognition, drug design) to tailoring and design of nanomaterials for specific use. Chemical Engineers will, no doubt, play an important role in research and education in these areas.

Over the last few decades, molecular simulation methods (e.g., Molecular Dynamics, Monte Carlo) have been accepted by the Chemical Engineering community as effective tools for simulating and understanding phenomena at the molecular level. For a given force field, movement of a system of molecules is tracked as a function of time by solving Newton's equations of motions numerically in a typical application of molecular dynamics (MD)[1–3]. For given thermodynamics conditions, trajectories of molecules obtained from MD simulations not only yield detailed microscopic states of the system (thermodynamics) but also the mechanisms of transitions from one state to another (kinetics). The success of the scheme can be gauged from the wide spectrum of areas that it finds applications in.

In addition to being an effective tool for performing cutting edge research at the molecular level, MD simulations can be used as powerful pedagogical tools. When combined with an appropriate visual interface (see for example, VMD at **http://www.ks.uiuc.edu**), time dependent coordinates of atoms and molecules can be turned into a movie. One may then follow simple processes, such as dissolution of salt in water, mixing or demixing of liquids as a function of their intermolecular interactions, or visualize more complex self-assembly phenomena like micelle formation in water or binding of a drug molecule to its target protein.

One shortcoming of the approach described above is that trajectories of molecules in the system need to be available *apriori*. Obtaining such a trajectory for realistic interactions between, for example a protein and water molecules, is time consuming and can keep state-of-the-art workstations busy for hours or even days. For systems with simple, short-ranged, intermolecular interactions and small sizes, however, computational expense is small. Thus, interfacing MD simulations of simple systems with a graphic interface can provide "on the fly" molecular-level visualization of the system. Providing an interactive graphic interface to a MD simulation can open up a whole new vista of possibilities including the demonstration of concepts such as diffusion or mixing, thermodynamics and phase behavior, self-assembly in solution, at the molecular level to undergraduates or even high school students.

II. EXAMPLE

Borrowing from the original work by Prof. Dennis Rapaport (**http://www.ph.biu.ac.il/~rapaport/java-apps/index.html**), we have recently begun developing Java applets specifically geared towards this purpose. Motivation for this effort is to create a multi-purpose visualization tool that can be easily modified and can be applied to display the dynamics of systems such as liquids, mixtures, polymers etc. at the molecular level.

Our prototype applets imbibe the following approach (**http://www.rpi.edu/~gardes/javamd.html**), the Newton's equations of motion are solved as a function of time beginning with an initial configuration and initial velocities for molecules whose interactions are governed by the Lennard-Jones potential function. For vi-

†Electronic mail: gardes@rpi.edu

sual and computational simplicity, the simulation is performed in 2-dimensional space. Figure 1 shows snapshots taken from an applet that illustrates the diffusion of an ink drop (red) in a fluid (green). Essentially, a few particles in the fluid are painted red; the underlying process is therefore that of the self-diffusion in a fluid of 2-dimensional Lennard-Jones disks. Using buttons on the top strip, temperature and fluid density can be increased or decreased to enhance or inhibit the diffusion process. Similar applets available on the web page demonstrate the mixing of two liquids and the phase changes observed in a single component system as a function of thermodynamic conditions.

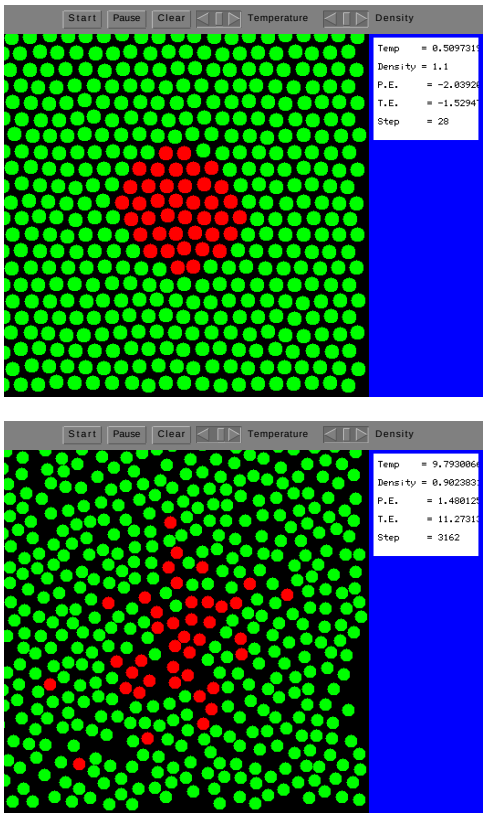

FIG. 1: Display of the prototype Java Molecular Dynamics software. Ink drop diffusion experiment

III. JAVA MD ON THE WEB

A pool of such readily available resources over the World Wide Web would facilitate their integration into a molecular thermodynamics curriculum. Several molecular simulation groups around the world have already made major contributions towards the development of these training tools. Here we list five excellent instances of web pages we found that offer interactive Java applets to perform molecular dynamics simulations.

1. Prof. Dennis Rapaport's group at Bar-Ilan University in Israel has developed a variety of applets for the visualization of phenomena in physics and related areas. The page (**http://www.ph.biu.ac.il/~rapaport/java-apps/index.html**) contains links to molecular dynamics applets that can be used to demonstrate interactions in rigid body and soft disk fluids.

2. Prof. David Kofke's group at SUNY Buffalo has developed applets that perform interactive simulations (both Monte Carlo and molecular dynamics) for various statistical mechanical ensembles for simple fluids such as those interacting with a square well like potential (**http://wings.buffalo.edu/eng/ce/kofke/applets**).

3. The Center for Polymer Studies at Boston University(**http://polymer.bu.edu/wamnet**) offers interesting demonstrations such as those for intermolecular interactions and the configurations of a polymer sampled using a "infinitely growing self-avoiding walk" algorithm.

4. The Virtual Laboratory on PhysicsWeb (**http://physicsweb.org/TIPTOP/VLAB**) contains links to an extensive set of applets including ones that illustrate a molecular model for an ideal gas and concepts in kinetic theory through the simulations of a gas of hard spheres.

5. The Webphysics project page at Davidson College has a large collection of MD applets that demonstrate pressure and temperature equilibration, the deviations from ideality of a gas etc. The page also contains tutorials and download instructions for the incorporation of these "physlets", as the authors call them, into existing course material (**http://webphysics.davidson.edu/Applets/java11_Archive.html**)

This list is, of course, far from complete. The above links are only representative instances of applets available on the Web that could prove invaluable in getting across concepts that are often not easy to comprehend.

IV. NUTS AND BOLTS

The programming of these applets, atleast for non-commercial purposes, offers relatively little complexity in terms of software development to a programmer familiar with the C/C++ syntax. Java offers the advantages of being a simple, object-oriented language and is well suited to Internet related tasks and is a solid, general-purpose language that has been developed by *Sun* for a wide variety of applications. Because Java

code is run by an interpreter, Java programs will run unmodified on any platform to which the Java interpreter has been ported. Moreover Java applets are accessible on the client-side, hence are faster than programs running on remote servers. Java is extensively documented (**http://www.java.sun.com**) and is freely available for most platforms. In addition, comprehensive tutorials are offered for beginners on getting started with building a simple graphical user interface for the applets.

The development of an applet to perform molecular dynamics simulations essentially comprises the following: (i) the main program or the so called "engine" of the code that performs computation of atomic coordinates and velocities for every time step of the MD simulation (ii) a graphics module that periodically reads atomic coordinates and updates the display to represent the most recent configuration and (iii) a user interface that allows users to modify either of or both the parameters used in the display or in the MD calculations. This can be difficult to extend and maintain in conventional programming languages, however, the Java Application Programming Interface (API) allows these three components to be easily integrated into one unified environment. Programming the graphical user interface in itself can draw upon the large collection of pre-defined classes for building graphical user interfaces in Java collectively known as the Abstract Window Toolkit (AWT).

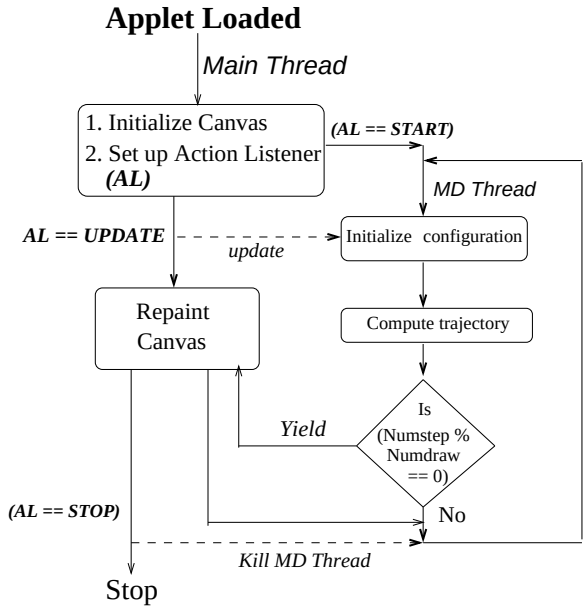

FIG. 2: Schematic illustration of the basic design of an applet to demonstrate molecular dynamics (MD) simulations.

How does one integrate the three different modules? Well, Java enables us to have multiple flows of control (each of which is called a *thread*); each flow of control can be thought of as a separate mini-program. So in running a Java applet, you will often have several threads handling the various activities. Since these threads run in the same memory space, they can easily communicate among themselves. For instance, to create a simple MD applet that demonstrates diffusion (illustrated in Figure 1), the main thread initializes the display and the system we wish to simulate. When prompted by the user to start the simulation, a new thread is created that computes the MD trajectory for the system. The MD thread periodically "yields" to the main thread to allow it to refresh the display using the current atomic coordinates. Buttons and scrollbars on the main graphic display talk to threads through easily implemented AWT classes. The schematic in Figure 2 illustrates the basic design of an MD applet.

Using these applets at present is not entirely devoid of pitfalls. After the code is written, the Java program compiles the source code into an intermediate state called class files. Class files are downloaded into the browser along with other objects, such as sound files or GIF images. The browser is now responsible for the layout of the page on the monitor and translation of the machine independent class files into native binary codes. This is a relatively slow process and makes Java unsuitable for computationally intensive tasks. Fortunately, browser vendors are now developing compilers that translate the entire class file into native machine code after downloading. These Just-In-Time, JIT, compilers have the potential of making Java almost as fast as C++ codes.

V. CONCLUSIONS

Java applets thus hold tremendous potential in terms of providing an interactive environment for "learning by doing", especially, in understanding phenomena at the molecular level. Disciplines, such as statistical mechanics and molecular thermodynamics, stand to gain enormously from the use of these applets for instructional purposes. Numerous groups around the world have made major strides towards the development of these applets. The goal of this short article is to emphasize the utilitarian value of such programs and motivate further development of these tools especially geared towards the needs of a changing Chemical Engineering curriculum.

REFERENCES

[1] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids* (Clarendon press, Oxford UK, 1987).

[2] D. Frenkel and B. Smit, *Understanding Molecular Simulation. From Algorithms to Applications* (Academic Press, Boston, 1996).

[3] D. Rapoport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge UK, 1995).