Real-time Visualization and Computational Steering of Molecular Dynamics simulations of Materials Science

C.F. Sanz-Navarro and S.D. Kenny

Modelling of Materials Group, Department of Mathematical Sciences, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK

A.R. Porter and S.M. Pickles

SVE Group, Manchester Computing, University of Manchester, Manchester, M13 9PL, UK

Abstract

This paper focuses on the use of real-time visualization and computational steering techniques in molecular dynamics (MD) simulations of problems in materials science. Simulations of nanoindentation are mainly used to illustrate the interplay between the computational steering, real-time visualization and the scientific discovery process. We will describe a VTK based viewer that we have developed for the visualization of MD simulations and the techniques we have utilised so that this can be used to view data in real time. We will give examples of how real-time visualization and computational steering can be used to aid the scientific discovery process. Further to this we will show how these techniques can be used to make maximum use of the models that we develop and can extend simulations beyond what was previously feasible.

1. Introduction

Nanoindention is an experimental technique for measuring the hardness of materials on the nanoscale and as such it can be used to measure the properties of thin films and nanoparticulate materials. A sharp tip attached to the end of a cantilever is pushed into the material and the required force is measured as a function of the depth the tip is pushed in. From this both the material's hardness and its Young's modulus can be measured. On retraction of the tip from the material any piling up of material around the hole made by the indenter can be studied. This gives some insight into the nature of the atomistic processes taking place in the sample.

Although the technique is becoming established as a useful experimental methodology for the measurement of the properties of materials on the nanoscale little is known about the atomistic processes that take place during the nanoindentation process. These can be elucidated through the use of modelling and to this goal we have modelled the nanoindentation process using molecular dynamics (MD), describing both the tip and the sample atomistically. This has allowed us to study the important atomistic processes that take place during the indentation and the retraction of the tip. We have used this approach to study a number of systems including Fe [1] and Si [2]. The comparison of the results with the experiments has shown that the model is able to pick up a number of macroscopic features that are observed in the experiments and provide an explanation for these at the atomic level.

2. The Model

The MD model consists of an atomistic substrate of the material to be indented and a hard diamond indenter, which is pushed into the substrate. The indenter was constructed from a diamond cube by cutting along a (111) plane [3]. Thus a 90˚ pyramidal indenter with three (100) diamond-like side planes and a (111) diamond-like basal plane was obtained. The indenter was blunted by rounding the tip to more accurately simulate the experiments. The interatomic interactions between the metal atoms have been modelled by using EAM type potentials [4], the Si and C interactions using Tersoff type potentials [5, 6] and the interaction between metal atoms and the C tip using a repulsive ZBL potential [7]. The effect of the cantilever was modelled by means of a vertical spring acting on the centre of mass of the top three layers of the indenter similar to that previously employed in the study of the stick-slip phenomenon [8, 9], as shown in figure 1. The position of the spring support was moved down towards the surface with a constant velocity during the loading before entering a slowing down period followed by a reversing of the velocity so as to simulate the retraction stage. The simulations of the nanoindentation

process are performed at constant temperature by first raising the temperature of the substrate and the tip to the desired temperature using a Nosé-Hoover thermostat [10,11] and then by applying a Nosé-Hoover thermostat to the atoms near the edge of the substrate and towards the top of tip during the simulation, as shown in figure 1. A 5 Å layer of atoms are held fixed on the edge of the substrate so the substrate does not simply move downwards during the simulation.

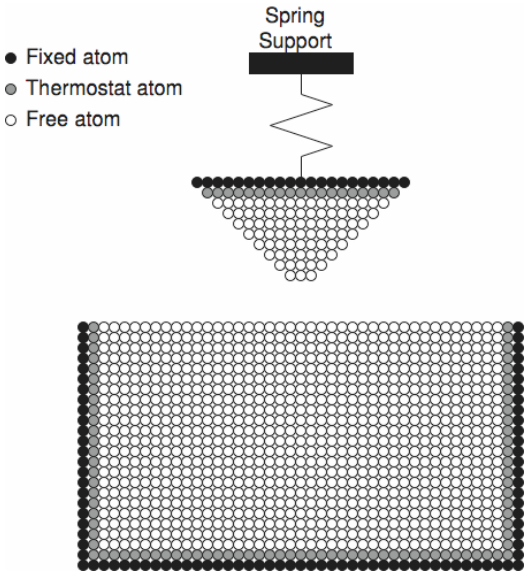

FIG 1. Schematic of setup used for modelling nanoindentation process, showing the spring used to mimic the cantilever and the boundary conditions employed.

3. Computational Steering and Real-time Visualization

A number of challenges face us when we use MD to model the nanoindentation process. If we take a typical system containing 2 million atoms and a typical simulation time of 150 ps with a time step of 1 fs, then the data created during the simulation would be about 5 TBs. The traditional approach to this problem would be to store data from one in every 500 (say) steps, thus massively reducing the data storage requirements. However, we are interested in studying dynamical events in this system and such a coarse sampling of the motion of the system often results in us missing the important dynamical events that take place. Through the utilisation of real-time visualization in conjunction with computational steering we can address this problem by altering the frequency with which we emit data from the simulation, when we arrive at interesting events in the simulation. When this is used with checkpointing, so that we can rewind a simulation and collect a greater amount of data during dynamical events, this allows us to study dynamical events in greater detail. In order to steer the MD simulation we make use of the RealityGrid steerer [12, 13]; figure 2 illustrates how the MD code (LBOMD) interacts with the steerer and the real-time visualizer (LBOVIS). The steering data is communicated through the use of SOAP which allows the data to be passed both from the MD code to the steerer and the steerable parameters to be passed from the steerer back to the MD code. The visualization data is communicated to the LBOVIS code through the use of sockets with the initial handshaking to set up the sockets being performed using SOAP.

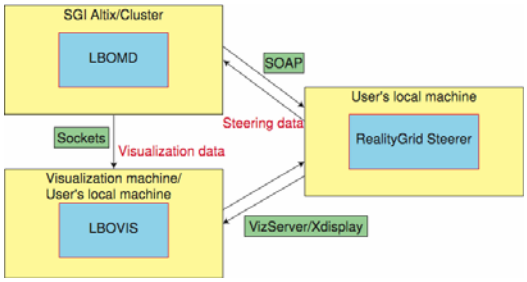

FIG 2. Illustration of the connection between the different components utilised when performing computational steering.

The use of sockets to communicate the visualization data provides significant improvements over our previous implementation, which utilised a shared file-system as the mechanism for transferring data. If we compare the time the visualization code spends retrieving the data from the socket with the time it takes reading it from a file, then we find that for an example case with just over 1 million atoms it takes 14.8 seconds to read the data from a file whereas it takes 2.3 seconds to retrieve the data from a socket. Similar savings are also seen in the time taken by the MD code to send data to the visualization. This allows us to increase the rate at which we can render images from the MD code, and to reduce the lag between what we visualize and the present state of the MD simulation.

In order to visualize the dynamical events that are happening in the system we have to filter our data in such a way that we only view the “important” atoms in the system. As well as

allowing us to focus on the atoms involved in the dynamical events this also allows us to visualize the system in "real" time. With systems typically containing a few million atoms rendering the complete system of atoms as spheres is not feasible in real time. We employ a number of filtering techniques depending on the problem being studied including energy filtering [1], automatic defect identification [14], coordination number filtering [2] and slip vector filtering [9, 15]. We then render the resulting images using the visualization toolkit (VTK) [16], with the atoms in the system being represented by spheres, which we colour according to the magnitude of the quantity that we use for our filtering. For example, figure 3 illustrates the patterns of dislocations found during the nanoindentation of a Ag substrate; here we use slip-vector filtering to do this, with the green atoms being ones that are on a stacking fault and the red atoms are those that are displaced by a full Burger's vector.

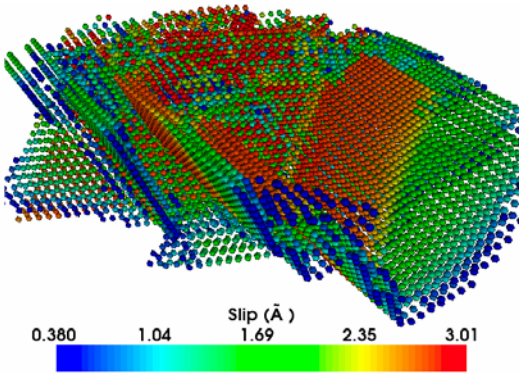

FIG 3. The pattern of dislocations around the indentation of a Ag (100) surface illustrated using the slip vector. Red atoms show the region of the lattice where the atoms have undergone a perfect dislocation, and the green atoms ones that are on a stacking fault.

Recent work on modelling the stick-slip process [9] has illustrated the role that dislocations play during stick-slip events in Ag. In a new model of this we have introduced the effect of temperature into the model to study its interaction with the stick-slip process. Conventional MD runs, which emit image files at regular intervals, show no signs of any dislocations in the system when we model the stick-slip process. However, when we study the system utilising the real-time visualisation techniques described above we see dislocation motion in the system. The problem that arises here is that the dislocation motion is so fast that they have disappeared from the simulation in the timescale of the regular image files that are emitted during a traditional MD approach.

An illustration of another role that computational steering can play in such modelling work comes from one of the issues that arises when modelling nanoindentation, which is that the depth of indentation reached during the experiments is much larger than what we can achieve in our models. For example, using a system containing more than 7.5 million Si atoms which has dimensions 600 Å x 600 Å x 400 Å and a tip containing nearly 0.5 million C atoms we can indent to about 8 nm; in a typical experiment they would indent to 50-100 nm. Given this, then for any particular system we want to achieve the maximum possible indentation depth with our model. During the process of the indentation of the sample both the spring, used to provide the elastic force, and the atomistic tip compress, this makes it impossible to estimate the true depth that the tip of the indenter will reach for a given movement of the spring support. Furthermore during the indentation phase it is necessary to monitor the stress in the crystal and the strain that arises from this to monitor how this interacts with our boundary conditions. Therefore, if we can steer the time for which the indentation proceeds, whilst monitoring both the depth of the indenter and the strain in the lattice in real-time, then we can optimise our indentation time for a given set-up. The tools that we utilise for this are illustrated in figure 4 and allow the user to visualize a filtered set of atoms as well as plot any of the parameters that are being monitored in the steering client.

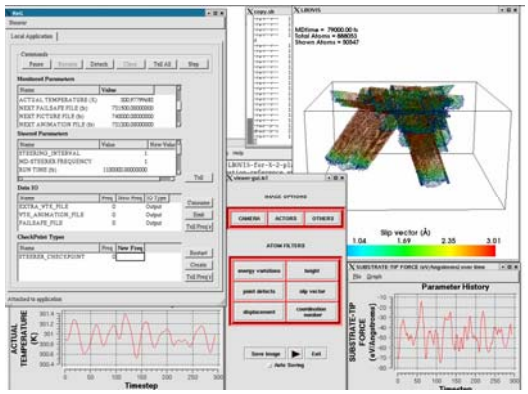

FIG 4. The figure illustrates a typical snapshot of the desktop tools available to the scientist during the computational steering of a MD simulation.

4. Discussions and Conclusions

We have shown that visualization is an essential part of the scientific discovery process due to the large number of particles modelled in modern MD simulations. Furthermore, in order to understand the dynamical events that take place during MD simulations we need to utilise real-time visualization to aid in the identification of defects. This is due to the vast amount of data that is created during massively parallel MD simulations, which makes it impossible for the scientist to gain insight without the use of visualization tools. We have also illustrated how computational steering along with the ability to checkpoint and restart codes can greatly facilitate this process.

We have described how we are using the computational steering tools to make better use of our models so that we can make the maximum use of any given model. This approach makes it possible to simulate systems that were not previously feasible using more conventional MD simulations. Furthermore, it makes more efficient use of the available computational resources by minimising the wastage of computer time.

Acknowlegements

This work has been funded by the EPSRC, GR/R67699/02. We would also like to thank Roger Smith of Loughborough University, Asta Richter of the University of Applied Sciences, Wildau, and Bruce King of the University of Newcastle, Australia for useful discussions.

References

[1] R. Smith, D. Christopher, S. D. Kenny, A. Richter, and B. Wolf, Physical Review B **67**, 245405 (2003).
[2] C. F. Sanz-Navarro, S. D. Kenny, and R. Smith, Nanotechnology **15**, 692 (2004).
[3] D. Christopher, R. Smith, and A. Richter, Nuclear Instruments and Methods in Physics Research B **180**, 117 (2001).
[4] G. Ackland, G. Tichy, and M. Finnis, Philosophical Magazine A **56**, 735 (1987).
[5] D. Brenner, Physical Review B **42**, 9458 (1990).
[6] D. Brenner, Physical Review B **46**, 1948 (1992).
[7] J. Biersack, J. Ziegler, and U. Littmack, The Stopping and Range of Ions in Solids (Pergammon, 1985).
[8] J. Shimuzuki, H. Eda, M. Yoritsune, and E. Ohmura, Nanotechnology **9**, 118 (1998).
[9] D. Mulliah, S. D. Kenny, and R. Smith, Physical Review B **69**, 205407 (2004).
[10] S. Nosé, Molecular Physics **52**, 255 (1984).
[11] W. G. Hoover, Physical Review A **31,** 1695 (1985).
[12] S. M. Pickles, R. Haines, R. L. Pinning, and A. R. Porter, in Proceedings of the UK e-Science All Hands Meeting (2004).
[13] S. M. Pickles, R. L. Pinning, A. R. Porter, G. Riley, R. Ford, K. Mayes, D. Snelling, J. Stanton, S. D. Kenny, and S. Jha, The RealityGrid Computational Steering API, http://www.sve.man.ac.uk/Research/AtoZ/RealityGrid/Steering/ReG_steering_api.pdf (2004).
[14] H. Hurchand, S. D. Kenny, R. Smith, and P. E. J. Flewitt, Submitted (2004).
[15] O. R. de la Fuente, J. A. Zimmerman, M. A. González, J. de la Figuera, J. C. Hamilton, W. W. Pai, and J. M. Rojo, Physical Review Letters **88**, 036101 (2002).
[16] K. Martin, W. Schroeder, and B. Lorensen, The Visualization Toolkit (VTK), http://public.kitware.com/VTK (2004).