PARALLEL PROCESSING OF MOLECULAR DYNAMICS SIMULATION IN A DISTRIBUTED COMPUTING ENVIRONMENT AND APPLICATION TO THE MODIFIED EMBEDDED ATOM METHOD AND NANOINDENTATION

By

DAVID L. STOKES

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1998

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2000

PARALLEL PROCESSING OF MOLECULAR DYNAMICS

SIMULATION IN A DISTRIBUTED COMPUTING

ENVIRONMENT AND APPLICATION TO THE

MODIFIED EMBEDDED ATOM METHOD

AND NANOINDENTATION

Thesis Approved:

[illegible]

Thesis Advisor

[illegible]

[illegible]

[illegible]

Dean of the Graduate College

ACKNOWLEDGEMENTS

Thanks are due to my wife, Kecia Stokes, for her patience, support, and understanding. Thanks also to my daughter, Drew – while she is far too young to realize it, she has been a great source of inspiration.

I also want to thank my advisor, Dr. Ranga Komanduri, for his help and advice. I have had the privilege of working for Dr. Komanduri for many years and I feel that this experience has been as valuable a part of my education as my coursework. Thanks to Dr. Lionel Raff for the many meetings and frequent and invaluable suggestions. Thanks also to Dr. Young and Dr. Price for serving on my committee.

Thanks are also due to my fellow research assistants. Naga Chandrasekaran has led all of us by example, and has paved the way with regards to molecular dynamics research at our lab. Kalyan Mavuletti and I worked in conjunction in the early stages of the MEAM project – without this initial group effort, my own work would have been infinitely more difficult. And thanks to Matt Lee, architect of MDWulf, without whom there would be no distributed computing cluster.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Molecular Dynamics Modeling of Ultra Precision Machining and Grinding

Advancements in technology are due in great part to the ability to achieve increasingly finer machining tolerances. Ultra Precision Machining (UPM) and Ultra Precision Grinding (UPG) are terms which refer to such highly precise material removal techniques. These methods are used in a number of important applications, including the manufacture of high-precision optics, hard drives, and wafers for the semi-conductor industry.

As for any physical process of importance to industry or the scientific community, having an accurate theoretical and/or computer-based model of UPM and UPG is highly desirable. Experimental investigation can be subject to high costs, experimental error, concerns about the repeatability of observed phenomena, and even potential health and environmental issues. An accurate computer model can augment experimental research or even, once the model has reached maturity, partially replace experimental research or testing.

For example, billions of dollars worth of design decisions are based partly or wholly on computer-based analysis with the finite element method (FEM). Despite the pioneering work of Hrenikoff, Turner, Clough, Arhyris, et. al. [1] a few decades ago, application of FEM to complex problems awaited advancements in computer science and computer engineering. Today, with the availability of inexpensive and powerful computers, FEM is recognized as a viable simulation tool in the areas of deformation and stress analysis, heat transfer, and flow problems.

Molecular dynamics (MD) modeling has the potential to be as significant as FEM. MD, one of the most widely employed simulation methods for studying the condensed state, is concerned with the interactions and behavior of individual atoms. Atoms are treated as point masses having defined potential interactions with neighboring atoms. The specified potential model is used to calculate forces on all atoms in the system, and the Hamiltonian equations of motion are integrated for each atom.

Because MD describes the behavior of materials at the atomic level, MD can be used to model UPM and UPG. When small amounts of material are removed during machining, the discrete nature of matter becomes important – at this level, continuum mechanics-based methods (e.g. FEM) fail.

Specifically, molecular dynamics is an ideal simulation tool for UPM and UPG for two reasons:

- MD offers the same potential advantages as other simulation tools – the cost and complexity of experimental research can be reduced or eliminated altogether. UPM and UPG is performed using machine tools which are highly precise and rigid. Contamination, vibration, and other external influences must be avoided, and single crystal diamond tools and diamond-abrasive grinding wheels must be used. Such equipment is prohibitively expensive for many researchers.
- MD can model processes which are impractical or impossible to investigate experimentally. Even with state-of-the-art technology, precise control and observation of material removal at the atomic level is not possible. MD, however, becomes increasingly viable when fewer numbers of atoms are considered. MD can be used, therefore, to promote and guide the development of new technology to achieve increasingly more precise machining tolerances.

As mentioned above, molecular dynamics simulation of materials becomes more viable as the scale of the problem is reduced. This is due to the fact that, as with FEM in the 1960's, the application of MD to real-world problems is limited by the speed of modern computers. With six coupled differential equations to be integrated per atom per time step, MD is computationally expensive. As a result, manufacturing process engineers employing MD for their research are often forced to use two-dimensional models and/or very high cutting velocities, to machine over limited cutting distances and with limited depths of cut, or to instigate clever techniques such as Length Restricted Molecular Dynamics (LRMD) as developed by Chadrasekaran [2].

Despite these limitations, nano-indentation molecular dynamics experiments can be used to effectively model the action of a single abrasive grain during grinding and the behavior of workpiece materials in general. Direct observation of the mechanism of chip formation and workpiece deformation at this level is, as mentioned above, difficult, time consuming, and expensive. Variation of experimental parameters, such as shape of the indenter, indentation depth and speed, and crystal orientation can be achieved precisely and easily with MD, without the need for expensive diamond tooling, single crystal samples, or machine tools.

The computationally intensive nature of molecular dynamics simulation can be partially compensated for by the use of parallel processing. The effective application of parallel computer hardware and algorithm design to MD, and to the *n*-body problem in general, is a challenging and potentially expensive endeavor. Molecular dynamics simulation is inherently parallelizable, but at a high level – vector processing, the basis for the success of supercomputers, does not apply well. An alternative approach is necessary, both in terms of hardware design and algorithm design.

1.2 Parallel Processing via the Message Passing Paradigm

While parallel computing in a distributed computing environment incorporating the "components-off-the-shelf" (COTS) philosophy and the message passing paradigm is still a new and rapidly evolving field, one thing remains true – for a fraction of the cost, researchers can build their own high performance computing environment which approaches the performance of expensive multi-processor machines made by SGI, Cray, IBM, and Compaq.

Because increased performance from a single processor is limited primarily by the ability to dissipate waste heat, computer scientists have resorted to using multiple processors to tackle larger and larger engineering and scientific simulation models. To quote an amusing analogy: If one wants to pull a heavy cart, it's easier to team up several horses rather than breed a single "superhorse".

This is the approach taken by the developers of traditional supercomputers, such as Cray. Two or more processors are housed in the same case which share the same memory and storage resources. At this point, however, the "horse and cart" analogy fails. While it is easy for many horses to efficiently subdivide the total workload, things are much more difficult in the computer science world.

There are three basic approaches to this difficulty:

- One is to develop "smart" compilers which take sequential code (a "normal" program in which each computation is performed only after the previous computation has been competed) and attempt to identify which steps are independent and can be executed simultaneously on different processors. This approach is called a "vector" approach. For example, it is common to perform the same operation on each member of an array (i.e. vector) – a "smart" compiler can take sequential code and subdivide this work among separate processors, with no extra effort on the part of the programmer.

- Another method is to use a true parallel architecture, such as a transputer. In this method, many very simple processors, each with their own small memory space, are interconnected on the same circuit board. Programming for this type of architecture necessitates the use of custom, low level languages, and is frequently exceedingly difficult for complex problems, such as MD.
- The final method is to use multiple full-featured processors, each with their own separate memory space. This architecture requires the manual parallelization of the desired simulation model, such as subdividing the grid or simulation space over the many processors. Algorithm design is governed by the desire to maximize processor independence – when data exchange is necessary, it is done via "message passing", whereby the programmer specifies exactly how and when to exchange data.

A big advantage to multiple processors in a single machine (frequently called SMPs, or "symmetric multi-processors") is that they share the same memory space. That is, if one processor needs the results of a computation performed by another processor, the data is right there in local RAM. As a result, inter-processor communication is fast. The disadvantage to this approach is that no compiler can automatically parallelize code as well as an experienced programmer who is willing to invest the time to do it himself. Also, as more processors share the same address space, it becomes increasingly more difficult to manage access to memory and to maintain fault tolerance. In addition, many problems (of which MD is a prime example) do not greatly benefit from vector processing.

As a result, the MD researcher looking for the best performance out of his supercomputer is often forced to abandon vector processing and take the message passing approach on his supercomputer. Here, the programmer manually parallelizes the code at a high level. A parallel algorithm is designed in which independent processing paths are identified and inter-process communication is manually performed via messages. In this approach the programmer has complete control but the programming complexity and debugging difficulties can be extreme. This method of algorithm design is also known as the Multiple-Instruction, Multiple-Data (MIMD) method.

While message passing eliminates the primary difficulty of using SMPs for high performance computing, it also eliminates the primary advantage – shared memory space. For this reason, processors housed in separate cases with separate memory and storage spaces may just as well be used. This approach results in slower communication of data between processors, but with careful algorithm design, this disadvantage is overshadowed by the advantages.

Such a collection of separate computers is called a distributed computing environment, or a Beowulf cluster. This is potentially much less expensive than a high performance SMP especially if the designer follows a COTS approach – this simply refers to the use of inexpensive and readily available components, such as 686 Intel processors and other components typically used in common home and business PCs.

It's possible to have over one hundred such PCs (or nodes) in a cluster. At a cost of a few to several hundred dollars per node, a 128 node cluster, for the right application, would have a much better cost/performance ratio as compared to a SMP with the same number of processors. Also, a cluster is easily expandable whereas an SMP is not.

Many issues come into play for such a large cluster. Fast message passing and robustness is crucial. There is a great deal of research going on in the areas of hardware, operating system and network configuration, and algorithm design. While these issues in and of themselves command the full attention of computer scientists, motivated researchers in other fields may, with work, develop enough basic knowledge to reap the benefits of the distributed computing concept for their simulation work.

1.3 The Modified Embedded Atom Method (MEAM)

When implementing MD simulations, the choice of potential model is all-important. There are many potential models which have been developed over the years, primarily by physical and organic chemists – Morse, Lennard-Jones, Embedded Atom Method, Modified Embedded Atom Method, Tersoff, and Brenner, just to name a very few. The majority of available potential models are defined for only a few materials. Others may be in a parameterized form, and can be applied to many different materials, but have been demonstrated to give poor results in some cases, such as the Morse potential with BCC materials [3].

The Morse potential has been the basis for much of the research carried out by Dr. Raff and Dr. Komanduri to study nanometric cutting, tribological processes, and nanomaterials properties. The Morse potential has the following advantages and disadvantages:

- As a simple pair potential, it is easy to implement and computationally much less intensive than other potential models.
- It is parameterized, and can model many FCC transition metals by variation of a few parameters.
- The Morse potential, and pair-potentials in general, give vacancy-formation energy to be the same as the cohesive energy, a major error for transition metals [4].
- The elastic properties of simulated materials are poorly represented due to the lack of volume-dependence in the model [5].
- The Morse potential does not prescribe parameters for interactions between different materials, so special consideration must be given to tool/workpiece interaction.
- Because of these facts, and the recent development and refinement of more sophisticated potential models, the Morse potential has come under increasing attack by members of the research community.

The Modified Embedded Atom Method, or MEAM, overcomes the disadvantages of pair potentials. Based on density functional theory and the quasiatom theory, it includes many body effects and accounts for angular components of bonding. It is also parameterized, and can model many FCC, BCC, and HCP transition metals, alkali metals, and even silicon, diamond, and germanium. Fundamental properties of matter are well-modeled with the MEAM, such as vacancy formation energy, sublimation energy, elastic properties, surface potential effects, and crack and dislocation effects [6]. The MEAM even specifically provides for interactions between unlike materials.

For these reasons, the MEAM has the potential to allow comprehensive and comparative modeling of many different materials and for workpiece/tool interactions. The current work focuses, in part, on the application of the MEAM to indentation of various materials. Results are compared to available experimental and theoretical values, and to previous work with the Morse potential.

In conjunction with the above mentioned advantages, the MEAM has one major disadvantage – it is computationally much more demanding than pair potentials. As a result, the MEAM is an ideal candidate for parallel processing. While the parallel processing techniques developed in this study are applicable to any potential model, they were developed and tested using the MEAM the model indentation.

The computationally intensive nature of the MEAM potential model has prompted additional investigation into methods of decreasing simulation time. One such technique discussed in this work is the use of the conjugate gradient method for crystal relaxation.

CHAPTER 2

LITERATURE REVIEW

2.1 Parallel Molecular Dynamics Modeling

Many architectural enhancements and parallel processing philosophies have been developed over the years. Each of these methods is intended to circumvent the fact that computer hardware is fundamentally clock-rate limited and that distribution of the computing workload over multiple processors can potentially offer a huge efficiency boost. Work in this area can be divided into roughly three hardware-design categories: transputer arrays, symmetric-multiprocessor (SMP) supercomputers, and distributed clusters. Each of these areas require a very different approach to the design of efficient and effective algorithms. Each specific type of architecture may also be best suited for particular applications.

Transputer arrays gained popularity throughout the 1980s and into the early 1990s. Transputers are an example of single-instruction-multiple-data (SIMD) parallelization at the hardware level. Multiple simple processors (often only 1 bit) are interconnected on the same circuit board. Each of these processors has its own RAM, apart from the RAM of other processors. Parallelization at the data level is accomplished by spreading out the algorithm over these processors.

Transputers were applied to molecular dynamics with the realization that vector processing yields limited speed increases. The first proposed implementation of transputers to MD was by Slaets and Travieso [7] in 1989. It this paper, an MD algorithm was designed which would apply well to a transputer. Use of the linked-cell method without neighbor lists was suggested. Each processor would perform the computation for one cell. A ring interconnection between processors was suggested.

Performance analysis indicated that this algorithm on T800 transputer would exceed that of a Cray X-MP for larger systems of atoms. It is interesting to note, however, that these researchers did not implement their proposed algorithm. This was due to the fact that the T800 transputer had an insufficient amount of memory for MD simulation, and that the low-level programming necessary on a transputer for a such a complex problem as MD is exceedingly tedious.

Later, in 1995, Woods and Alford [8] implemented MD on an array of transputer nodes configured as a scaleable torus. By interconnecting several transputers, a multiple-instruction-multiple-data (MIMD) approach was possible. In this paper, a highly detailed performance analysis was given to determine optimum hardware configuration, algorithm design, interconnection topologies, and communication procedures. A great deal of insight was derived into the efficient and scaleable implementation of MD models on networked transputers.

While there are still some adherents today to this technology, for the most part, transputers have been left by the wayside. The main problem with parallel algorithm development for transputers is the lack of programming tools. In most cases, a custom transputer language, OCCAM, is used. Use of this language is tedious, as low-level subdivision of the algorithm must be done manually for optimum performance. Researchers have generally preferred parallel architectures which allow the use of higher level languages, such as Fortran or C.

A second type of parallel architecture in which MD has been implemented is traditional vector processing supercomputers. To take advantage of vector processing, careful modification of the MD algorithm is necessary. This was first demonstrated by Heyes and Smith [9] in 1987, and carried out by Rapaport [10] in 1988.

Complications arise from the fact that when using the linked cell (LC) method, the typical vector length is of the order of the cell occupancy list, and is too short for efficient vectorization. Rapaport devised the layered linked cell (LLC) method, whereby larger flat cells are used to span the simulation space. In this layered link cell algorithm, the inner loops are completely vectorizable and of a length on the order of the number of cells. This LLC method, when carefully optimized, was the fastest algorithm for vector supercomputers reported in the literature as of the late 1980s.

In 1994, Everaers and Kremer [11] made an improvement on the LLC method. A fine grid is superimposed over the layered linked cells, fine enough such that these small cells are occupied by at most one particle. This obviates the need to distinguish between inter- and intra-cell interactions, and allows for longer loops, which are more easily vectorizable. Additional complexity is imposed by the necessity to search multiple cells for neighbor interactions, but with careful implementation, Everaers and Kremer reported a three times speedup as compared to the simple LLC method for uniformly distributed particles. With density variation, the amount of speedup decreases, but was still significant.

Despite these efforts, it is impossible to escape the fact that higher-level MIMD-style parallelization is much better suited for efficient MD simulation. MIMD-style parallelization for MD usually involves spatial decomposition – that is, each processor is assigned to a particular region of space and handles all computation for atoms in this region of space.

An early example of such an approach is given by the work of Liem et. al. [12] in 1991. While a transputer array was used in this work, the focus was to develop an algorithm which would work on any distributed memory type of cluster. In this work, a method of using one processor per linked cell was described. The benefits and requirements of this method were enumerated. These included the need for each processor to communicate only with the processors which represent contiguous cells, and the need to do particle reallocation, in the event that particles move into a sub-region associated with another processor.

This approach has been the basis for all subsequent MIMD-style parallel short-range MD algorithms. Later work has focused on load balancing, portability, and fast message passing. Srinivasan et. al. [13,14] reported on an interesting project in 1997 to address the lack of portability suffered by most parallel MD implementations. A runtime library, called Adhara, was written specifically for parallel MD simulation over distributed nodes. Written in C, able to be compiled on any system, it provided functionality for atom definition, partitioning and distribution over processors, and other features to allow programmers to take advantage of the parallelism inherent in MD. This library depended, however, on an SMP-type of processor arrangement.

Portable parallel programming, with the capacity for load balancing and fault tolerance, and with an emphasis on fast message passing, is implemented most widely today using the Parallel Virtual Machine (PVM) or Message Passing Interface (MPI) standard libraries. PVM was developed by Al Geist et. al. [15] at the Oak Ridge National Lab in 1993 [15]. MPI was first developed in 1994 by the MPI Forum [16,17,18]. Each of these is a set of libraries which may be compiled and used on a variety of architecture types. With these high-level libraries, portable and efficient parallel programs are easier to develop. A free and highly portable implementation of MPI called MPICH was used as the basis for parallel algorithm development in this study.

2.2 MD Modeling of Nano-Indentation

Alder and Wainwright [19,20], of the Lawrence Radiation Laboratory, conducted the first MD simulation studies in the late 1950s. They initiated numerical simulation of many atom or many molecule systems because of the great mathematical difficulty involved in analytical treatment. Alder and Wainwright recognized the potential of MD simulation to generate more detailed information as compared to actual experimentation. Also, MD offered the possibility of studying problems for which present theory had difficulties addressing, such as the behavior of a pure liquid, phase transactions, or the nucleation problem.

It is interesting to note that, due to the limitations of computers at the time of Alder's and Wainwright's pioneering work, it was necessary to used a vastly simplified potential model, the square-well potential. Also, the models used were limited to 500 atoms – even then, one-half hour of processing time was required with an average of only one collision per atom using a Univac or IBM 704 computer. These early MD models were found to accurately describe some equilibrium properties of the simulated hard spheres.

Cutting and indentation was first addressed by Belak et. al. [21] at the Lawrence Livermore National Laboratory in the late 1980s and early 1990s. Using a SPRINT 64-transputer, they were able to run million atom MD simulations of nonequilibrium indentation. The Lennard-Jones potential in conjunction with the Embedded Atom

Method (EAM) was used to simulate two-dimensional crystals of copper and nickel. Microhardness and yield strength were found to be between 10% to 25% of the shear modulus, which is in reasonable agreement with the theoretical values given in Hertzberg [22].

In 1993, Belak et. al. [23] also investigated nanoindentation of copper and silver. In these simulations, indentation of (1 1 1) surfaces with a triangular diamond tip were performed. Initial attraction between the indenter and the work material occurred, and elastic-plastic deformation of the substrate was observed. Critical yielding with a corresponding decrease in loading force was observed after penetration by the indenter to a depth of about 1.5 atomic layers. With greater indentation depth, pile up of atoms around the tool occurred.

In 1990, Landman et. al. [24] published experimental and simulation results on the indentation of a gold surface by a nickel indenter, using atomic force microscopy (AFM) and MD. They reported that, as the indenter approaches the gold surface, atomic-scale instability occurs causing a jump-to-contact (JC) phenomenon. At contact, adhesion-induced flow was observed, followed by plastic yielding and slip in the surface region of the gold substrate. Withdrawal of the indenter resulted in wetting of the nickel tip by gold atoms, formation of an atomically thin connective neck, followed by fracture of the neck. AFM analysis could not confirm many of these features predicted by the simulation.

Landman, Leudtke, and Ringer [25] expanded the above investigations in 1991 to include interionic (CaF_2) and thin alkane films adsorbed onto a gold surface. These molecules were represented by "pseudo-atoms" using the EAM potential. Tip sliding was also performed, to simulate the mechanism by which AFM is conducted. Oscillatory variation of force as a function of lateral displacement was observed. This was attributed to atomic-scale stick-slip behavior.

Rentsch et. al. [26] used the Lennard-Jones potential in 1994 to model the indentation and scratching of copper by a diamond indenter. The purpose of this study was to simulate the action of a single abrasive grain during UPG. Chip formation and pile up was observed. The scope of this study was limited by the computational demands of MD – to scratch for long enough lengths to observe more of the chip formation processes required unreasonably long simulation times.

Indentation of silicon by a diamond indenter was investigated by Brenner et. al. [27] in 1996. Using the Tersoff potential for silicon, and the Lennard-Jones potential at the interface, pure elastic deformation was observed in the substrate with an indentation depth up to 0.6 nm. Further indentation produced irreversible damage to the substrate, with the profile of the damage normal to the surface matching the indenter profile, and circular deformation on the in-plane profile.

In 1998, Yan and Komvopoulos [28] used the Lennard-Jones potential to model FCC crystal substrates with a single atom or with rigid indenters at various temperatures.

Jump-to-contact phenomena during indentation were observed and the normal force on the indenter varied in a saw-tooth-like pattern. Materials differences had strong influence on the resulting force-distance curves, and elastic stiffness and yield strength were found to decrease with increasing temperature.

In 2000, Komanduri et. al. [29] used the Morse potential to investigate anisotropic effects during the indentation and scratching of single crystal aluminum. Measured hardness was found to increase with a decreasing indentation depth. Variation in measured hardness and friction coefficient values resulted from variation of the crystal orientation. Also, a method was presented to quantify and graphically display the amount of disorder at various positions in the machined substrate throughout the indentation process.

2.3 Development of the MEAM

In 1972, R. A. Johnson [30] addressed the inadequacies of pair potentials when addressing point defects in metals, including vacancies and substitutional and interstitial impurity atoms. The Lennard-Jones, Morse, and Born-Mayer functional forms were investigated, and functional parameters were derived. These simple two-body potentials were found to yield poor agreement with vacancy properties. Johnson made the observation that the only way to obtain physically realistic results is with a short ranged empirical type of potential.

Specifically, Johnson suggested that a new approach to the development of potential models was needed. The commonly used pair potentials, Lennard-Jones, Morse, and Born-Mayer, were based on the behavior of only a very few atoms. These potential models were then assumed to apply to larger systems of atoms. The empirical approach taken by Johnson involved pure curve fitting to attain agreement with known physical properties of the material in question.

The concept of the "quasiatom" was introduced by Scott and Zeremba [31] in 1980. Using density-functional theory, a method of estimating the energy of an impurity in a host lattice of atoms was presented. The entire impurity atom, the nucleus and electron cloud, was treated as a single unit, or "quasiatom". The focus of the quasiatom theory was on the effect of the host on the embedded quasiatom, not on the effect on the host lattice. The potential energy of an embedded impurity was considered to be a function of the electron density contributed by the host at the site of the impurity. This electron density was calculated without the impurity in place.

This was the key issue – without the impurity, the electron density of the otherwise pure lattice can, for many materials, be estimated with density functional theory. With this new method, the authors were able to reproduce correct qualitative trends for both hydrogen and helium, which are chemically very different.

Puska et. al. [32] expanded this work in 1981. Energies for atoms of hydrogen through argon in a homogenous electron gas were determined using the density-function scheme. Comparisons to the work of Stott and Zeremba were favorable.

Daw and Baskes [33] introduced the Embedded Atom Method (EAM) in 1984. The quasiatom concept was borrowed and applied to any atom in a host lattice. The electron density-dependent term was augmented by a pairwise term to represent the core-to-core repulsion between atoms. These pairwise and density terms were derived for nickel and palladium empirically using properties such as elastic constants, heats of solution, and migration energy. These methods were applied to calculate surface energies for various crystal orientations, and hydrogen migration in bulk Ni and Pd and on Ni and Pd surfaces with good results.

In 1986, Foiles, Daw, and Baskes [34] published a consistent set of EAM embedding and pairwise functions for Cu, Ag, Au, Ni, Pd, and Pt. Equations were presented in a parameterized form, which permitted the simulation of each of these FCC materials with the same basic model. Good agreement with experimental values for each material was reported when calculating migration energy of vacancies, formation energy, surface energies, and other properties.

Baskes [35] extended the EAM to model silicon, a covalent material, in 1987. It was found that a simple first-neighbor EAM model was sufficient to describe the geometry and structure of many metastable phases of silicon, but could not describe its shear behavior. This was due to the fact that the EAM model spherically averages the electron density contributions from each atom. Directional components of bonding, important for covalent materials such as silicon, are not addressed by the EAM model. To compensate for this, Baskes adjusted the EAM to account for bond-bending forces. Baskes' model reproduced the bulk lattice constant, sublimation energy, and bulk modulus of diamond-cubic silicon to within 5% of experiment.

Johnson and Oh [36,37] applied the EAM to HCP metals in 1988 and BCC metals in 1989. However, the most significant extension to the EAM came from Baskes et. al. [38] in 1989 with the Modified Embedded Atom Method, or MEAM. In this model, the angular components of bonding were formalized. A set of functions were presented which were applicable to silicon and germanium, and their alloys. Comparison of the predictions of this model to first-principle calculations and experiment were favorable for the calculation of the energetics and geometrics of point defects, surfaces, metastable structures, and small clusters.

The MEAM was further extended and formalized by Baskes [39] to a set of parameterized equations for silicon, germanium, diamond, many BCC and FCC transitions metals, and some alkali metals. Baskes cautioned that this extension was empirical and was not justified by strong physical arguments, as had been the EAM and many pairwise potentials. Despite this caution, Baskes confirmed that his model could describe the elastic behavior and simple defect properties of these diverse materials, as well as bulk structural and surface properties. Also, with this MEAM model, it was sufficient to consider first-nearest neighbors only, since energy differences were accounted for by angular terms in the electron density functions, not by long range interactions.

This model, as set forth by Baskes, was used as the basis for this investigation into nanometric indentation.

CHAPTER 3

PROBLEM STATEMENT

Three basic aspects of molecular dynamics simulation are addressed in this study with the intent of offering improvement in each area. These three objectives, the focus of this study, are as follows:

Objective 1: It is necessary to have a computing environment and algorithm design which can perform the necessary calculations in a reasonable amount of time for a reasonably large system of atoms. While "reasonable" is a relative term, it is defined here to mean a few to several thousand atoms in no more than a day or two. Also, while a certain level of performance may be acceptable, additional speed is always desirable. As a particular MD algorithm becomes faster, more simulations may be run in the same amount of time, or larger systems of atoms may be studied. The desire to achieve additional computational speed always exists.

For this reason, the present study includes the development of a MIMD-style parallel algorithm to decrease simulation time. This is especially necessary for the MEAM because its mathematical complexity makes it slower as compared to pairwise potentials. Also, since this is an entirely new area of work for this research group, the methods used and discoveries made in this investigation will provide guidance for the future expansion of the distributed computing cluster. Also, these techniques are by no means limited to the MEAM potential – future parallelization of the Morse potential, based on the techniques developed in this study, would allow the modeling of many more atoms that would otherwise be possible. The software written in support of this research is modular and may be extended to other potential models without great difficulty.

Objective 2: When conducting MD simulation, an accurate potential model is desired which, for the physical processes and materials to be simulated, accurately describes the behavior of the atoms in the system. This study focuses, in part, on the implementation of the MEAM to model nanometric-scale indentation. The deficiencies of the Morse potential, and pairwise potentials in general, have already been enumerated. The MEAM will be developed and its suitability for nanoindentation will be evaluated. The insight gained will also allow assessment of the possible application of the MEAM to the modeling of other manufacturing processes.

Objective 3: For successful MD simulation, a program is needed which correctly implements the desired potential model, and passes validation tests, such as conservation of energy and back-integration. Due to the mathematical complexity of the MEAM potential, this step requires additional attention and care. This investigation will document the accurate implementation of the MEAM. Verification of the model via validation tests will be demonstrated, for interactions between both like and unlike materials.

CHAPTER 4

THEORY OF MD SIMULATION

4.1 Introduction

This chapter will discuss the details and mathematics of molecular dynamics as applied to manufacturing processes simulation. The discussion is divided into six broad sections which include the selection of physical units used in the MD model, calculation of interatomic forces, integration of the equations of motion, boundary and thermal considerations, crystal relaxation, and algorithm design issues.

4.2 Units Used in the MD Model

The selection of physical units is an extremely important point when designing a computer-based model of a physical process. Computers have a limited precision available when performing floating-point arithmetic. This important fact can cause a perfectly valid model to yield invalid results if the programmer does not take care. Limited precision results in round-off and truncation error when performing computer-based calculations. For example, when adding numbers of various magnitudes, different answers may result depending on the order in which the summations are made. While some error is unavoidable, it can be minimized to the point of irrelevance with the careful selection of units.

Units should be chosen such that the raw values used in the simulation do not become too large or small. If all numbers are close to 1.0, plus or minus a few orders of magnitude, rounding and truncation errors will generally be insignificant. This guiding principle was used when selecting the following physical units for the MD simulation model used in this study:

Table 4.1 - Physical Units of the MD Model

Fundamental Parameter	Unit Used
length	angstrom (A), 10^{-10} m
mass	atomic mass unit (amu), 1.66×10^{-27} kg
energy	electron volt (eV), 1.60×10^{-19} J
time	1.02×10^{-14} s

The time unit, or time step is not selected – it is determined by the choices for length, mass, and energy.

4.3 Calculation of Interatomic Forces

Successful trajectory-based molecular dynamics simulation has as one basic requirement the ability to calculate accurately and quickly all forces on all atoms in the system. The force on a single atom *i* is defined as the negative of the rate of change of the total potential of the system as the position of atom *i* is varied, with all other atoms remaining stationary. All atoms *k* which bond to *i* must be taken into consideration:

$$\vec{f}_i = -\sum_{k \neq i} \frac{\partial V_{total}}{\partial r_{ik}} \vec{r}_{unit} \tag{1}$$

A different but equivalent formulation can be derived by separate consideration of the three orthogonal space dimensions:

$$f_{i,x} = -\sum_{k \neq i} \frac{\partial V_{total}}{\partial x_{ik}} \tag{2}$$

$$f_{i,y} = -\sum_{k \neq i} \frac{\partial V_{total}}{\partial y_{ik}} \tag{3}$$

$$f_{i,z} = -\sum_{k \neq i} \frac{\partial V_{total}}{\partial z_{ik}} \tag{4}$$

While either formulation may be used, the second is more convenient in some cases. This is particularly true when forming analytical expressions for forces for the electron density terms of the MEAM.

Forces may be found via the above definitions using analytical or numerical means. Analytical forms of the forces are necessary for accuracy and speed, while numerical estimates of forces serve as a useful diagnostic tool. Numerical derivatives were determined in this study using a formula suggested by Raff [40] as being well-suited for MD force estimation:

$$\left.\frac{\partial V}{\partial x}\right|_{x=x_0} = 0.75S_1 - 0.15S_2 + 0.01666\overline{6}S_3 \tag{5}$$

where:

$$S_1 = \frac{V(x_0+\Delta x) - V(x_0-\Delta x)}{\Delta x} \tag{6}$$

$$S_2 = \frac{V(x_0+2\Delta x) - V(x_0-2\Delta x)}{\Delta x} \tag{7}$$

$$S_3 = \frac{V(x_0+3\Delta x) - V(x_0-3\Delta x)}{\Delta x} \tag{8}$$

The parameter Δx is selected to give the greatest accuracy. With infinite floating-point precision, a smaller value for Δx would always give more accurate estimates. However, with the 64-bit double precision on the Digital Alpha C compiler and processors used in this study, there is a lower limit below which accuracy suffers. Determination of the optimum value of Δx was done through trial and error.

While these formulas for numerical estimates of forces are easily implemented, the force values they generate have some inaccuracy built into them. Also, use of these formulas requires a great many evaluations of the potential of the system of atoms which is very time consuming. To ensure a valid simulation which executes in a reasonable amount of time, analytical forms must be derived using Eqns. 1-4.

For the MEAM, this is a tedious and time-consuming task. The MEAM (as discussed in Chapter 5) is mathematically complex, especially as compared to simple pairwise potentials. To reduce the complexity of this task in this investigation, each individual term in the MEAM potential expression was handled one by one, and comparisons of analytical derivatives were made to numerical derivatives to verify correctness at each stage.

4.4 Integration of the Equations of Motion

Given the state of the MD model at a given point in time (i.e., the positions and momentums of all atoms in the system), and forces calculated as above, a new state of the molecular dynamics model may be determined at a more advanced point in time through integration of Hamilton's equations of motion. For each atom, the following six coupled first-order differential equations must be solved:

$$\frac{dx_i}{dt} = \frac{p_{i,x}}{m_i} \tag{9}$$

$$\frac{dy_i}{dt} = \frac{p_{i,y}}{m_i} \tag{10}$$

$$\frac{dz_i}{dt} = \frac{p_{i,z}}{m_i} \tag{11}$$

$$\frac{dp_{i,x}}{dt} = f_{i,x} \quad (12)$$

$$\frac{dp_{i,y}}{dt} = f_{i,y} \quad (13)$$

$$\frac{dp_{i,z}}{dt} = f_{i,z} \quad (14)$$

In these formulae, *p* represents momentum, *f* force, *m* mass, and *t* time. In other words, two coupled integrations over the three space dimensions must be performed. Force is integrated to yield an updated momentum and momentum is integrated to yield an updated position. This processes is repeated to establish the trajectories of the atoms in the MD model through time.

The integration method used in this study was the fourth-order Runge-Kutta method. While other methods may also be used with success, the Runge-Kutta method is self-starting, stable, it yields good accuracy for reasonable time steps, and it is not difficult to implement.

4.5 Boundary and Thermal Considerations

Due to the computationally intensive nature of trajectory-based molecular dynamics simulation, it is important to include in the simulation model a minimal number of atoms. For the computing resources available in this study, an atom count of roughly 10^5 atoms was the maximum system size possible in order for the simulation to complete in a reasonable amount of time. However, even many orders of magnitude more atoms would still be wholly insufficient to represent actual tools or substrates. For this reason, it is necessary to impose artificial conditions at the boundaries of the MD model to mimic the effects of bulk material outside the model.

Bulk material beyond the modeled portion of the substrate contributes two basic effects: resistance to deformation and translation, and thermal influences. These effects are simulated through the creation of three distinct atom types: moving atoms, peripheral atoms, and boundary atoms as proposed by Riley et. al. [41] in 1988.

Moving atoms represent the majority of atoms in the MD model, and are considered as "normal" atoms, with no special constraints or considerations imposed upon them. Moving atoms are free to move under the influences of forces applied due to neighboring workpiece and/or tool atoms.

Peripheral atoms are also free to move under the influence of applied forces. Peripheral atoms are established in a layer with a thickness of one lattice constant over the surfaces of the simulation model where additional bulk material is assumed to exist. To emulate heat transfer to the bulk workpiece, the velocities of the peripheral atoms are reset periodically according to:

$$\vec{v}_i^{\,new}(t_n) = (1-w)^{0.5}\,\vec{v}_i^{\,old}(t_n) + w^{0.5}\,\vec{v}^{\,random}(\xi, T) \tag{15}$$

That is, at some reset time t_n, a new velocity for peripheral atom i is set to a random value (as determined by ξ) selected from a Boltzmann distribution at temperature T. The random velocity is modified by the old velocity as determined by the parameter w, which determines the strength of the reset. Following Riley, the frequency of the reset is set to five times the Debye frequency of the lattice, and w is set to 0.1047.

Typical machining processes generate significant amounts of thermal energy near the cutting, grinding, or indentation action. This thermal energy, manifested in the kinetic energy of the moving and peripheral atoms in the system, will typically be attenuated through the peripheral velocity reset function, simulating heat loss to the bulk workpiece.

The outermost edges of the MD model where bulk workpiece is assumed to exist are comprised of boundary atoms. Equations of motion are not integrated for boundary atoms, and they are not allowed to move under the influence of applied forces. In fact, forces on boundary atoms are usually not calculated to save time. Boundary atoms simulate the resistance to deformation and translation which bulk material would provide. In addition, when a workpiece or tool is moved to model some process, such as indentation, the boundary atoms are moved at the desired velocity. The forces they exert on moving and peripheral atoms results in translation of the entire workpiece or tool.

An illustration of a typical arrangement of these three atoms types is provided in Figure 4.1. All surfaces but one have these layers of peripheral and boundary atoms. This single "free" surface is where indentation occurs.

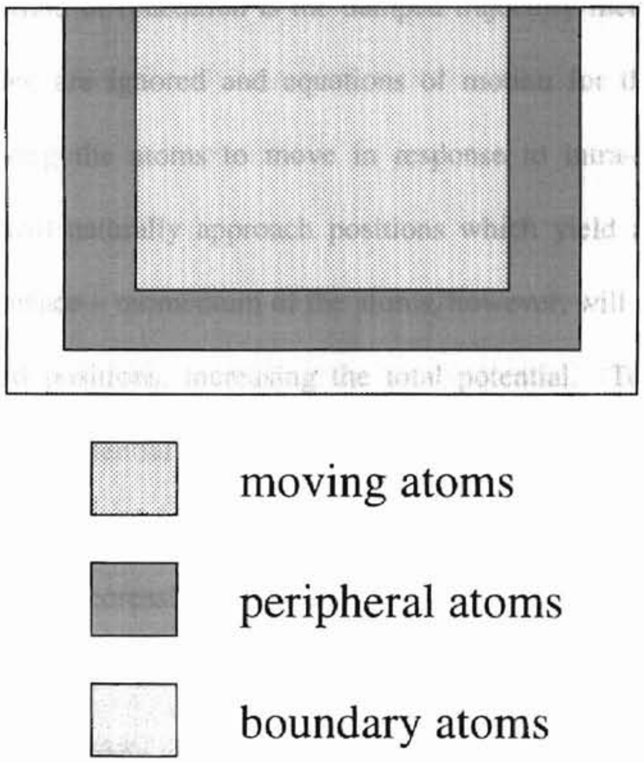

Figure 4.1 – Arrangement of Peripheral and Boundary Atoms

4.6 Crystal Relaxation

A pure crystal of any given transition metal at room temperature has a long-range order (face centered cubic or body centered cubic) and a lattice spacing associated with it. Within the bulk of a perfect crystal, there is little or no deviation from this geometric definition of atomic arrangement – near the edges of the crystal, however, deviation from this ideal arrangement is necessary for the crystal to achieve a minimum potential state.

To reach this minimum potential state in an MD simulation, the workpiece lattice is relaxed prior to the start of the experiment. A widely used, effective, and easily implemented method of relaxation is the damped trajectory method. With this method, effects of the tool are ignored and equations of motion for the workpiece atoms are integrated, allowing the atoms to move in response to intra-crystal forces. By this process, atoms will naturally approach positions which yield a local minimum in the potential hypersurface – momentum of the atoms, however, will cause the atoms to move past their relaxed positions, increasing the total potential. To prevent this, when an increase in the total potential occurs, momentums of atoms are set to zero. This process is continually repeated, causing the system to oscillate about the minimum potential configuration with a decreasing amplitude of oscillation. When the minimum has been approached to a reasonable degree of accuracy, the process is terminated, and the crystal is completely relaxed.

This damped trajectory method has been used exclusively by Dr. Raff, Dr. Komanduri, and their research groups. It is the best choice for simple pairwise potentials, and is easily programmed. With the MEAM, however, the damped trajectory method proved to be very slow, and an alternative method was implemented. This method, the conjugate gradient method, treats the relaxation process as a pure optimization problem – its use in this study is documented in Chapter 6.

4.7 Algorithm Design Issues

While accuracy of the molecular dynamics algorithm is the primary concern when coding the model, efficiency is second. Knowledge of basic computer science and computer engineering principles, and familiarity with the specifics of the computer system being used allow the MD simulation programmer to write programs which run faster than would otherwise be possible.

When designing a computer algorithm where performance is an issue, there are two pairs of conflicting goals which must be balanced against each other:

- Redundant calculations may be avoided at the expense of increased memory usage.
- Speed may be increased at the expense of program readability, simplicity, and debugging ease.

A few decades ago, there was less freedom when balancing the first pair of issues due to the limited memory available on typical computers. As memory became cheaper and smaller, bookkeeping techniques, such as the neighbor list and linked cell method, became feasible.

Neighbor lists refers to the practice of determining which atoms are sufficiently close to bond and storing in memory these pairs of bonded atoms. Typically, atoms are considered to bond if there separation distance is less than some fixed distance, known as the cutoff radius. This neighbor list is potentially very large, but the same list may be used throughout potential and force calculations for the same time step. Additionally, neighbor lists may be used over multiple time steps if bonds are stored for atoms with a separation distance larger than the cutoff radius. With this approach, it is assumed that this augmented bond is valid for a few or more time steps, and that it requires less frequent updating. Determining the extended cutoff radius and the number of time steps for which it is "safe" to use this bond lists requires careful attention and typically depends on the potential model used.

The simplest way to determine the bond list is to compute the square of the distance between all possible pairs of atoms, and compare this distance to the square of the cutoff radius to see if a bond is formed (using squares of distances avoids square roots). This step takes an amount of time on the order of the number of atoms squared.

However, if the simulation space is subdivided into cubes, or cells, and if the atoms are presorted within these cubes, then to determine bonds, atoms need to be compared only to atoms in neighboring cells. This technique is called the linked cell method – it is not difficult to implement, but determination of optimum parameters, such as cell size and number of cells is not easily determined. Also, with different sizes of MD models, the optimum choice of parameters changes. However, with well chosen parameters, the time required to form a bond list with the linked cell method is close to being on the order of the number of atoms to the first power.

These two techniques are standard in most modern and well-written MD algorithms. They illustrate the need for compromise mentioned above: use of these methods gives greater speed, but more memory is required, and their use complicates the algorithm considerably, making it more difficult to debug, modify, and harder for others to understand.

These features are incorporated into the MD algorithm implementing the MEAM developed in this investigation. Additionally, some lower-level optimizations were used which reduced computation time by a few percent. With a simple pairwise potential, these efforts would have a negligible effect – with the computationally intensive MEAM, however, they were worth the effort. On a computer, floating point multiplication is faster than division, so division was eliminated and/or minimized where possible. Loop counters were specified to be stored in CPU registers where possible. The number of variables used within a loop was minimized to allow, if possible, complete caching (that is, storage of data in memory which is faster than RAM) of data used in a loop. Lastly, while extensive use of global variables is generally considered poor programming technique, they are preferable when speed is an issue – passing values to and from functions and procedures requires pushing/popping data from the stack, which requires additional time.

CHAPTER 5

THE MODIFIED EMBEDDED ATOM METHOD (MEAM)

5.1 Introduction

The quasiatom principle in conjunction with density functional theory gave rise to the Embedded Atom Method (EAM). The MEAM is an extension of the EAM where angular bond dependence has been incorporated into the computation of electron density at the site of any atom. Further details on the history and development of the MEAM can be found in Chapter 2.

The MEAM as set forth by Baskes [6] is the basis of this study. This chapter will describe the MEAM, how it was implemented in this study, and the methods used for verifying the accuracy of the program.

It is important to note that Baskes warned about his extension of the MEAM – the extension was empirical and has not been justified by strong physical arguments, as have the EAM and pairwise potential methods. Baskes' purpose was not to derive optimum parameters for each material considered, but rather to set up a framework for atomistic calculations.

For the extension to binary systems, Baskes assumed that the partial electron density weights depend only on the embedded atom type, and not on the type of atom contributing the density. Also, electron densities are not scaled. Baskes warned that these assumptions are extreme, and recommended further investigation before widespread application.

In this investigation, these warnings are kept in mind. Evaluation of the accuracy and suitability of the MEAM is presented in Chapter 9.

5.2 Formulation of the MEAM for Interactions Between Like Materials

The total potential energy of a system of atoms using the EAM is given by:

$$V = \sum_i \left(F_i(\overline{\rho}_i) + \frac{1}{2} \sum_{k \neq i} \phi_{ik}(r_{ik}) \right) \tag{16}$$

This formula gives the total energy as a sum over all atoms i. (Note that subscripts denote material types as well as specific atoms). The potential energy for a single atom i is a function of a linear superposition $\overline{\rho}_i$ of spherically averaged electron densities at the site of atom i. In this initial formulation, the pairwise term, ϕ_{ik}, was purely repulsive, and represented core to core electrostatic effects.

In the MEAM, the electron density at site i, $\overline{\rho}_i$, is augmented by angular terms and is renormalized by the number of nearest neighbors, Z_i:

$$V_i = F_i\left(\bar{\rho}_i / Z_i\right) + \frac{1}{2}\sum_{k \neq i} \phi_{ik}(r_{ik}) \quad (17)$$

The pairwise term has taken a more general form in the MEAM. For interactions between the same material (that is, atoms *i* and *k* are the same material):

$$\phi_{ii} = \frac{2}{Z_i}\left(E_i^u(r) - F_i\left(\rho_i^0(r) / Z_i\right)\right) \quad (18)$$

The first term represents an average of the energy per atom of the reference lattice at each of the nearest-neighbor distances. The second term is formed by the difference between the embedding energy at the background electron density actually seen by atom *i* and the average embedding energy of this atom in the reference lattice at each of the nearest-neighbor distances.

The total energy of a system of atoms of the same type using the MEAM is therefore:

$$V_{total} = \sum_i \left(F_i\left(\bar{\rho}_i / Z_i\right) + \frac{1}{Z_i}\sum_{k \neq i} E_i^u(r_{ik}) - \frac{1}{Z_i}\sum_{k \neq i} \frac{F_i\left(\bar{\rho}_i^0(r_{ik})\right)}{Z_i} \right) \quad (19)$$

The first partial background electron density at site *i*, $\bar{\rho}_i^0$, is given by:

$$\bar{\rho}_i^0 = \sum_{k \neq i} \rho_k^{a(0)}(r_{ik}) \quad (20)$$

The total electron density at site i is defined in terms of the first partial background density (Eqn. 20) and three correction terms that explicitly depend upon the relative positions of neighbors of atom *i*:

$$(\rho_i)^2 = \sum_{l=0}^{3} t_i^{(l)} \left(\rho_i^{(l)}\right)^2 \tag{21}$$

The correction terms are:

$$\left(\rho_i^{(1)}\right)^2 = \sum_{\alpha} \left(\sum_{k \neq i} x_{ik}^{\alpha} \rho_k^{a(1)}(r_{ik}) \right)^2 \tag{22}$$

$$\left(\rho_i^{(2)}\right)^2 = \sum_{\alpha,\beta} \left(\sum_{k \neq i} x_{ik}^{\alpha} x_{ik}^{\beta} \rho_k^{a(2)}(r_{ik}) \right)^2 - \frac{1}{3} \left(\sum_{k \neq i} \rho_k^{a(2)}(r_{ik}) \right)^2 \tag{23}$$

$$\left(\rho_i^{(3)}\right)^2 = \sum_{\alpha,\beta,\gamma} \left(\sum_{k \neq i} x_{ik}^{\alpha} x_{ik}^{\beta} x_{ik}^{\gamma} \rho_k^{a(3)}(r_{ik}) \right)^2 \tag{24}$$

where $x_{ik}^{\alpha} = {r_{ik}^{\alpha}}/{r_{ik}}$ and r_{ik}^{α} is the α component of the distance vector between atoms *k* and *i*. The forms are chosen such that the partial background electron densities are invariant to lattice translation and rotation, scale simply with atomic electron density for homogeneous deformation, and equal zero for a cubic lattice with a center of symmetry.

The terms of the form $\rho_i^{a(l)}(r)$ are given by:

$$\rho_i^{a(l)}(r) = \exp\left(-b^*\right) \tag{25}$$

where:

$$b^{*} = \beta_i^{(l)}\left(r/R_i^0 - 1 \right) \tag{26}$$

The energy to embed an atom at site i is a function of this density, and is given by:

$$F_i(\rho) = A_i E_i^0 \rho \ln \rho \tag{27}$$

This form is unchanged from the EAM and has been shown previously [4] to give the correct coordination dependence between bond length and energy.

Finally, expanding the pairwise term E_i^u:

$$E_i^u(r_{ik}) = -E_i^0(1 + a^{*})\exp(-a^{*}) \tag{28}$$

where:

$$a^{*} = \alpha_i\left(r_{ik}/R_i^0 - 1 \right) \tag{29}$$

In these formulations, the terms E_i^0, R_i^0, α_i, A_i, $\beta_i^{(0)}$, $\beta_i^{(1)}, \beta_i^{(2)}, \beta_i^{(3)}$, $t_i^{(0)}$, $t_i^{(1)}, t_i^{(2)}, t_i^{(3)}$, Z_i, $S_i^{(0)}$, $S_i^{(1)}, S_i^{(2)}$, and $S_i^{(3)}$ are all specified parameters for atoms i of a given material.

5.3 Formulation of the MEAM for Interactions Between Unlike Materials

In previous work [4], when formulating the EAM's application to alloy systems and impurities, the dilute heats of solution were used to scale the electron densities, and unlike-atom pair potential parameters where determined using a geometric mean between the like-atom potentials.

In the MEAM a different approach is used. Pairwise parameters are derived by considering, not a dilute alloy system, but an equiatomic binary intermetallic alloy system, where each i atom has only k neighbors, and vice versa. With this approach, the following pair potential terms are derived for interactions between unlike materials:

$$\phi_{ik}(r_{ik}) = \frac{1}{Z_{ik}}\left[2E_{ik}^{u}(r_{ik}) - F_i\left(\frac{Z_{ik}\rho_k^{a(0)}(r_{ik})}{Z_i}\right) - F_k\left(\frac{Z_{ik}\rho_i^{a(0)}(r_{ik})}{Z_k}\right)\right] \tag{30}$$

where $E_{ik}^{0} = \left(\frac{E_i^0 + E_k^0}{2}\right) - \Delta_{ik}$ (Δ_{ik} is the heat of formation for equiatomic compounds of material i and k), $\alpha_{ik} = \frac{(\alpha_i + \alpha_k)}{2}$, Z_{ik} is the number of nearest neighbors to an i or k atom, and R_{ik}^{0} is calculated from the assumed equilibrium intermetallic atom volume.

No modification to the embedding energy term is necessary. However, care must be taken with the specified subscripts.

In summary, with the above noted modification to the pairwise terms, Eqn. 17 also applies to unlike materials.

5.4 Angular Screening

In the application of the MEAM, Baskes reported "it was found that a simplification to first-neighbor interactions for all crystal structures was possible" [6]. This minimizes the number of bonds which must be considered, which increases simulation speed. However, it becomes necessary to define what a first nearest neighbor is, especially in situations where crystal structure has been distorted.

Such a procedure must be continuous in the energy and its first two derivatives to ensure that force discontinuities do not occur. One method is to institute a fixed cutoff distance, beyond which potential and force effects are ignored, or beyond which all distance-dependent functions are smoothly forced to zero. While this method is sufficient for pairwise interactions, Baskes proposed implementing a "screening" method in addition to a static cutoff.

The initial screening method proposed by Baskes [6] had some deficiencies, namely a discontinuity in the second derivative. Baskes [42] later developed an alternative method which corrected these errors – this corrected version was used in this study and is presented here.

A screening method should take into account the actual geometry of the atoms under consideration. A pair of atoms, *i* and *k*, may be fully screened, partially screened, or unscreened depending on atoms which may be between *i* and *k* (see Figure 5.1). If an atom *j* is inside the ellipse $C = 0.8$, atom *j* completely screens *i* and *k*, and *i* and *k* are considered to have no interaction. If atom *j* is outside the ellipse $C = 2.8$, then *i* and *k* are completely unscreened, and their potential effects are unmodified as given by Eqn. 17. Finally, if an atom *j* lies in the intermediate region (such as atom *j* in Figure 5.1), then atoms *i* and *k* are partially screened.

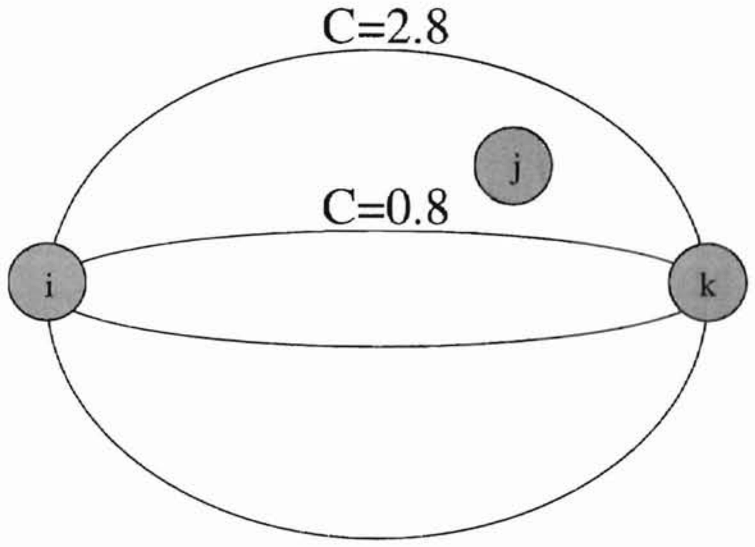

Figure 5.1 - Illustration of Angular Screening Method

This screening method is formulated as follows. Equations of ellipses formed by atoms *i* and *k* on the major axis, with parameter *C*, are:

$$x^2 + \frac{1}{C}y^2 = \left(\frac{1}{2}r_{ik}\right)^2 \tag{31}$$

where the x and y axes are in the plane defined by atoms i, j, and k. For a given atom j, the parameter C is determined by:

$$C = \frac{2(X_{ij} + X_{jk}) - (X_{ij} - X_{jk})^2 - 1}{1 - (X_{ij} - X_{jk})^2} \tag{32}$$

where $X_{ij} = \left({r_{ij}}/{r_{ik}} \right)^2$ and $X_{jk} = \left({r_{jk}}/{r_{ik}} \right)^2$, and r is the distance between the denoted atoms. A screening factor S_{ijk} for any three atoms i, j, and k is defined by a smooth function of the parameter C defining the ellipse formed by i, j, and k:

$$S_{ijk} = f_c\left(\frac{C - C_{\min}}{C_{\max} - C_{\min}} \right) \tag{33}$$

Here, $C_{\max}$ and $C_{\min}$ are the limiting values of C as shown in Figure 5.1. The smooth cutoff function is the following piecewise continuous function:

$$f_c(x) = \begin{Bmatrix} 1 & x \geq 1 \\ \left(1 - (1-x)^4\right)^2 & 0 < x < 1 \\ 0 & x \leq 0 \end{Bmatrix} \tag{34}$$

where the argument x is the argument defined in Eqn. 31.

Finally, the screening factor between a pair of atoms i and k is given by:

$$S_{ik} = \prod_{j \neq i,k} S_{ijk} \tag{35}$$

In other words, the screening between two atoms i and k is the product of all partial screening factors formed by considering the screening effects of all other atoms j in the system. If any atom j completely screens i and k, the product becomes zero, and complete screening occurs. If no atoms screen i and k, all S_{ijk} 's are 1, giving $S_{ik} = 1$, corresponding to no screening. Finally if partial screening occurs without any complete screening, S_{ik} will be some value between zero and one.

This screening function is incorporated into the MEAM for like materials by modifying the pairwise terms in Eqn. 19 as follows:

$$V_{total} = \sum_i \left(F_i\left(\bar{\rho}_i \big/ Z_i \right) + \frac{1}{Z_i} \sum_{k \neq i} S_{ik} E_i^u(r_{ik}) - \frac{1}{Z_i} \sum_{k \neq i} S_{ik} \frac{F_i\left(\bar{\rho}_i^0(r_{ik})\right)}{Z_i} \right) \tag{36}$$

Also, the screening function affects the electron density through modification of the partial electron densities (Eqns. 20-24):

$$\bar{\rho}_i^0 = \sum_{k \neq i} S_{ik} \rho_k^{a(0)}(r_{ik}) \tag{37}$$

$$\left(\rho_i^{(1)}\right)^2 = \sum_{\alpha} \left(\sum_{k \neq i} S_{ik} x_{ik}^{\alpha} \rho_k^{a(1)}(r_{ik}) \right)^2 \tag{38}$$

$$\left(\rho_i^{(2)}\right)^2 = \sum_{\alpha,\beta} \left(\sum_{k \neq i} S_{ik} x_{ik}^{\alpha} x_{ik}^{\beta} \rho_k^{a(2)}(r_{ik}) \right)^2 - \frac{1}{3} \left(\sum_{k \neq i} S_{ik} \rho_k^{a(2)}(r_{ik}) \right)^2 \tag{39}$$

$$\left(\rho_i^{(3)}\right)^2 = \sum_{\alpha,\beta,\gamma}\left(\sum_{k\neq i} S_{ik} x_{ik}^{\alpha} x_{ik}^{\beta} x_{ik}^{\gamma} \rho_k^{a(3)}(r_{ik})\right)^2 \tag{40}$$

Modification of the MEAM of unlike materials is done in a similar fashion.

5.5 Validation of the Implementation

The MD program implementing the MEAM was tested using the following two rigorous tests:

- Conservation of energy test
- Back-integration test

The conservation of energy test determines if the analytical forms of the interatomic forces derived according to Eqn. 1 are correct. If energy is conserved, the sum of potential and kinetic energy of the system of atoms should remain constant to within a reasonable degree as determined by the accuracy of the integration of equations of motion.

For a system of fourteen copper atoms, energy was conserved as shown in Figure 5.2. The same constancy is seen for any material and any size system.

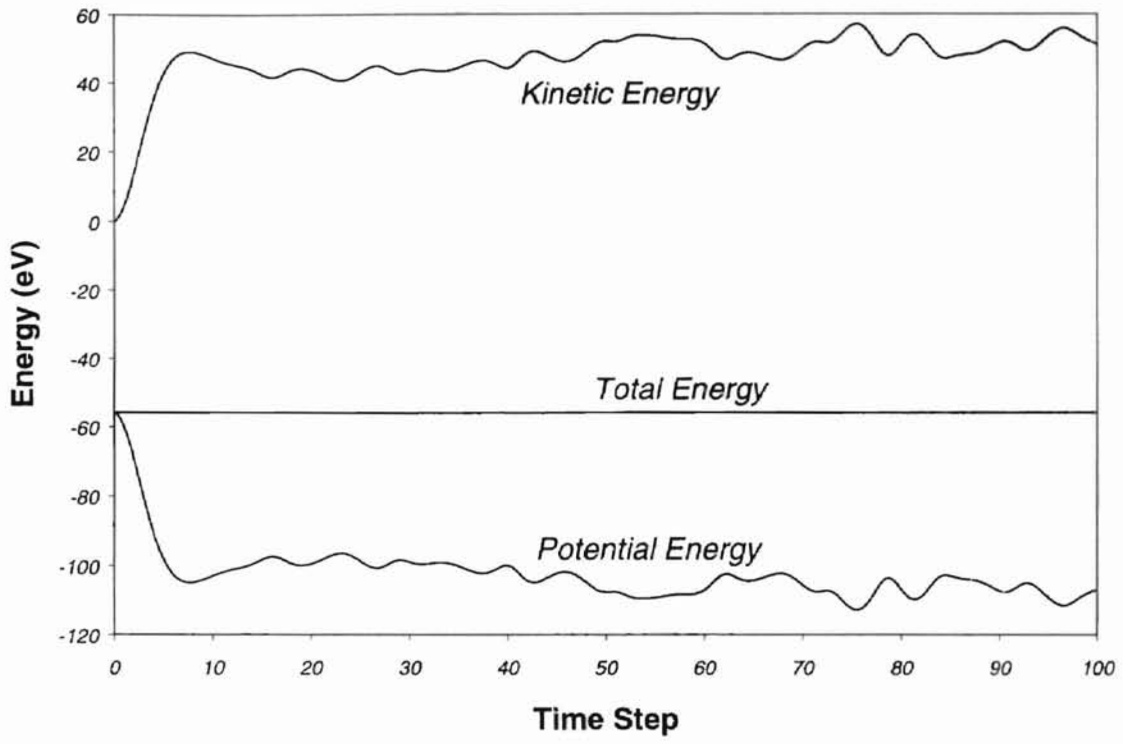

Figure 5.2 - Conservation of Energy for Cu System

For interactions between unlike materials, the MEAM formulates different equations. Because this necessitates the use of code within the program separate from the code tested above, it is also necessary to test for conservation of energy when different materials interact. For a system of seven silver atoms and seven iron atoms, refer to Figure 5.3.

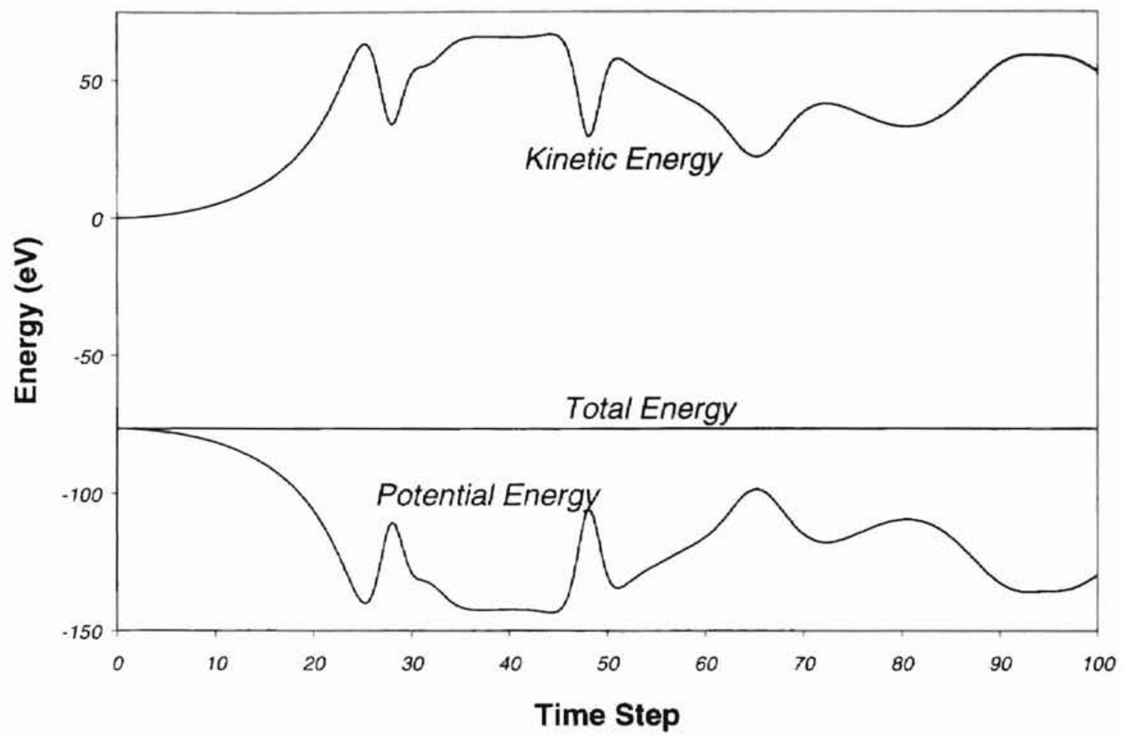

Figure 5.3 - Conservation of Energy for Ag-Fe System

A rigorous test of the integration function may be done using the back-integration test. If the integration method is properly coded, using a negative time step will simulate the system of atoms backwards in time. The back-integration test involves integrating the equations of motion for a system of atoms normally for some period of time, followed by reversing the integration direction. A properly working integration function will be able reproduce the same motion of the atoms, along with the same potential and kinetic energy values. This should occur for many time steps until normal integration error forces the system to another path. Figures 5.4 and 5.5 illustrate back-integration tests for the same systems used in the conservation of energy tests using an integration time step of 0.1.

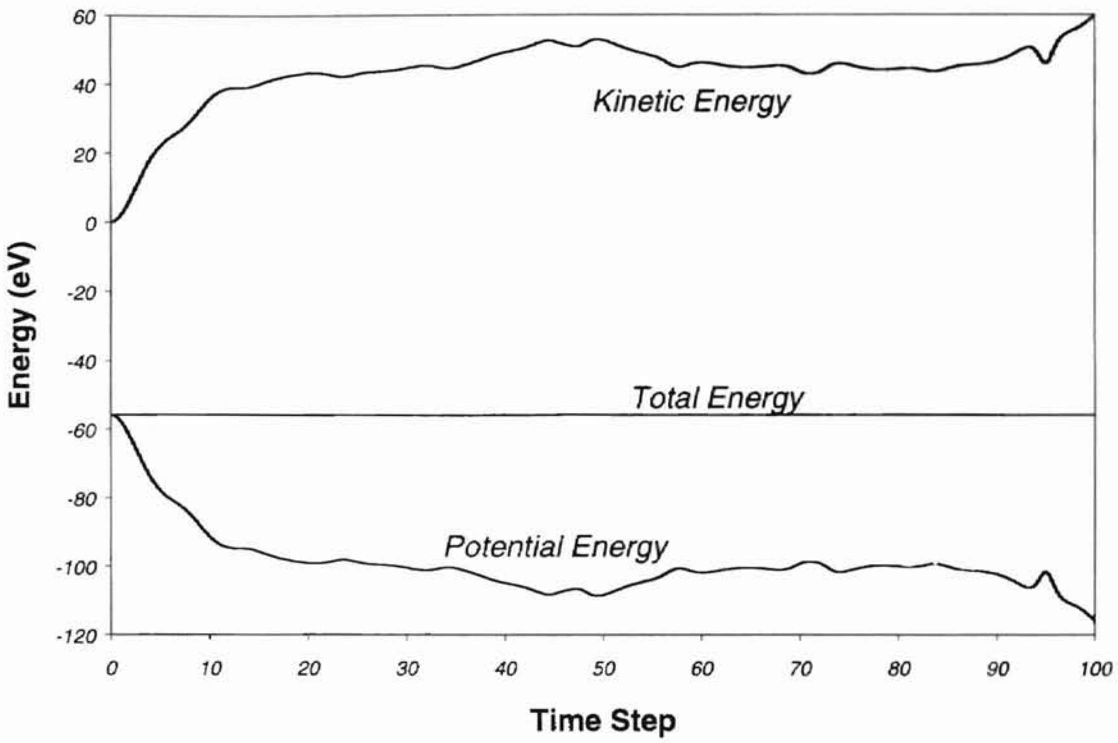

Figure 5.4 – Back-integration for Cu System

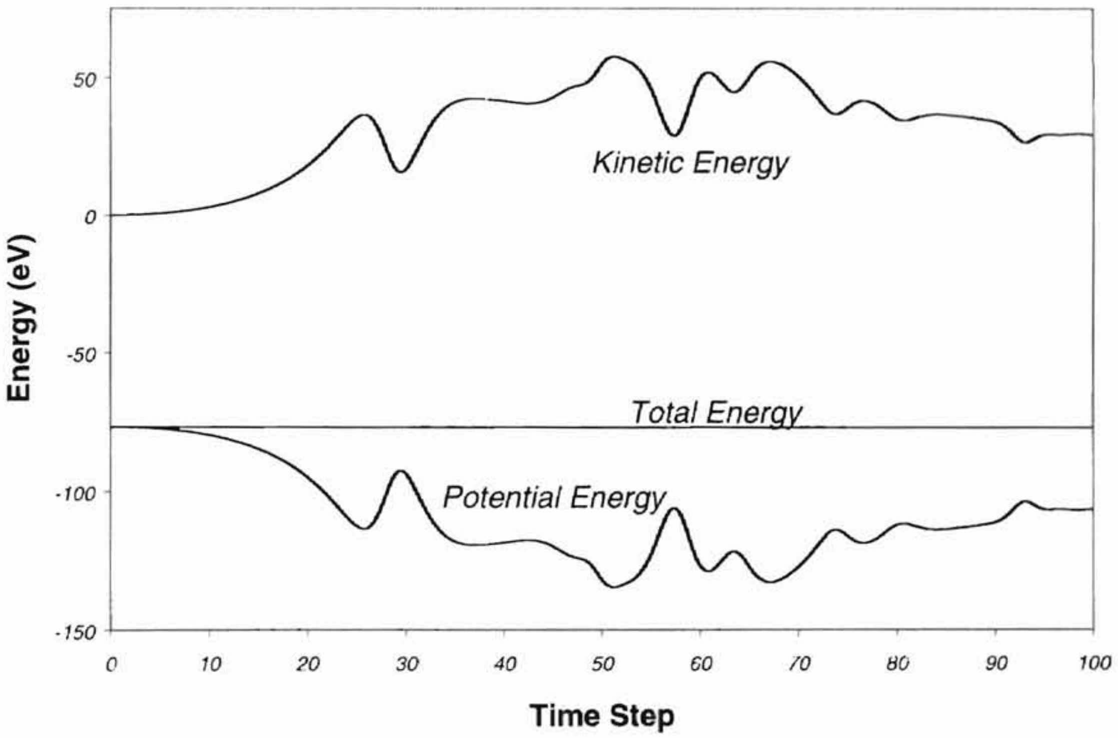

Figure 5.5 – Back-integration for Ag-Fe System

As can be seen, the MD model backintegrates very well. Only a slight thickening of the lines demonstrates the slight deviance which results from the small amount of unavoidable integration error.

CHAPTER 6

RELAXATION WITH THE CONJUGATE GRADIENT METHOD

6.1 Introduction

In molecular dynamics simulation, a computer program which correctly implements the desired potential model, which computes correct forces on all atoms in the system, and which properly integrates the equations of motion, is essential. However, a functional program which runs very slow is almost as useless as a non-functional program.

This issue became evident with the completion of a working MD simulation code implementing the MEAM. Before large-scale application of the code to problems of interest, such as indentation, significantly greater speed was needed. While use of neighbor lists, the linked cell method, and various low-level optimizations (as discussed in Chapter 4) helped, much more improvement was necessary.

As a first step in addressing this issue, it was necessary to identify the bottleneck in the MEAM program. The important functions in the code were timed for various system sizes as shown in Figure 6.1. Reported run times are for a single processor Digital Alpha 500au workstation and comparisons are made to the Morse potential.

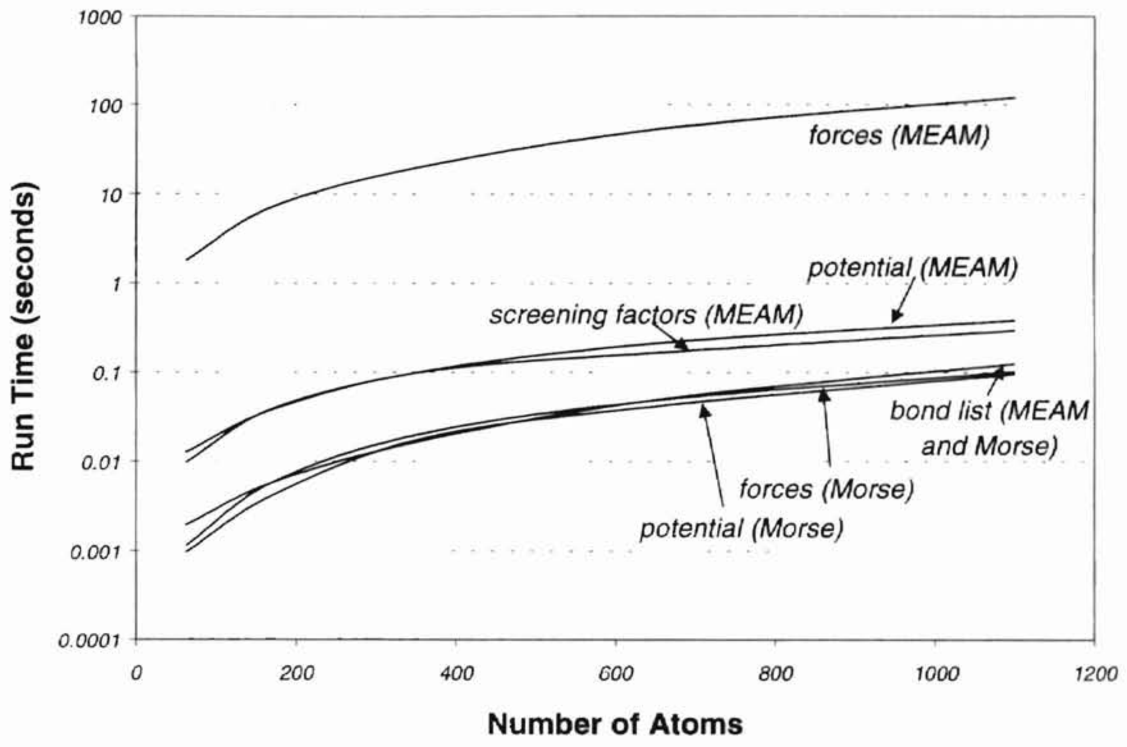

Figure 6.1 – Run Times of Significant Functions in the MEAM and Morse Code

As can be seen, the MEAM is slower that the Morse potential. With the Morse potential, the formulation is quite simple, so calculation of potential or forces will be dominated by the function which determines the bond list. Also with the Morse potential, there is no large difference between the times required to calculated forces and potential.

With the MEAM, however, additional overhead is imposed due to the need to calculate screening factors. More significant, however, is the most time consuming function, the forces function. Due to the complex form of the potential, and screening factors, a great many complex terms appear in the analytical forms for the forces. Calculation of these many terms is quite time consuming.

While parallel processing can be applied with great success, as discussed in Chapter 7, a less drastic first measure is possible, namely the implementation of a new method for relaxation – the conjugate gradient method. This new technique takes advantage of the large run time disparity between the forces and potential functions with the MEAM.

6.2 Optimization of the Relaxation Process

As discussed in Chapter 4, the damped trajectory method is an effective method for relaxation. It is easily implemented, since it reuses existing code. As discussed in Chapter 4, equations of motion are integrated and atoms are forced by the laws of physics toward a minimum potential configuration. Convergence is assured through timely cancellation of momentum.

The relaxation process is not required to conserve energy. As a result, a large integration time step may be used as integration error is not a key issue. Also, the masses of atoms may be temporarily set to small values so they will accelerate more quickly under the action of applied forces.

The damped trajectory method, by integrating equations of motion, makes many calls to the forces function. This is fine for pairwise potentials, where calculation of forces is no more time consuming than calculation of potential. For the MEAM, however, calculation of forces is much more time consuming than calculation of potential. For this reason, a relaxation method which can make more efficient use of gradients of the potential hypersurface would be well suited for the MEAM. The conjugate gradient technique is such a method.

6.3 The Conjugate Gradient Method

The conjugate gradient method is a well-established optimization technique. The form of the conjugate gradient method used in this study is based on the formulation by Hagan and Demuth [43] for the training of neural networks.

The conjugate gradient method involves the determination of a search direction based upon the local gradient of the potential hypersurface. An interval is found which contains a local minimum along this search direction. This interval is then reduced to a specified tolerance. These steps are repeated until the change in potential from the previous step falls below another specified tolerance. With this method, forces are calculated once per step to determine the search direction – the interval location and reduction steps involve potential function calls only.

An overview of the conjugate gradient algorithm is as follows:

1) Set initial search direction to be the negative of the gradient, $\overline{p}_0 = -\overline{g}_0$.
2) Locate an interval in the current search direction containing a local minimum.
3) Reduce this interval to a specified tolerance using a Golden Section Search.
4) If the change in the potential from the previous step is below a specified tolerance, quit.
5) Set new search direction as follows:
    a) If the number of iterations exceeds some specified number, reset the algorithm by setting the new search direction to the new gradient.
    b) Otherwise, determine the new search direction according to: $\overline{p}_k = -\overline{g}_k + \beta_k \overline{p}_{k-1}$, where $\beta = \dfrac{\overline{g}_k^T \overline{g}_k}{\overline{g}_{k-1}^T g_{k-1}}$
6) Go to step 2.

The potential function is called repeatedly during each step in order to locate and minimize the current interval, but for each step, the gradient (forces) needs to be calculated only once. For the MEAM potential (and presumably for any potential model where forces are significantly more computationally expensive the potential), this speeds up the relaxation process considerably.

Figure 6.2 shows a comparison of the two relaxation methods. For the damped trajectory routine, the usual techniques are used to speed up convergence – atomic weights are set to 1.0, and a large time step of 0.25 is used. The algorithm is repeated until 30 potential increases occur, which has been found to be sufficient to reach a minimum to almost all the available digits of precision for smaller time steps ($\lesssim 0.05$). Tolerances for the conjugate gradient method are 10^{-7} and the algorithm is reset every five iterations. The frequency of the algorithm reset was determined heuristically to give the quickest convergence for the tests summarized in Figure 6.2. As before, run times are for a single processor Digital Alpha 500au workstation. Also, in every case tested, the two methods converged to the same atom arrangement.

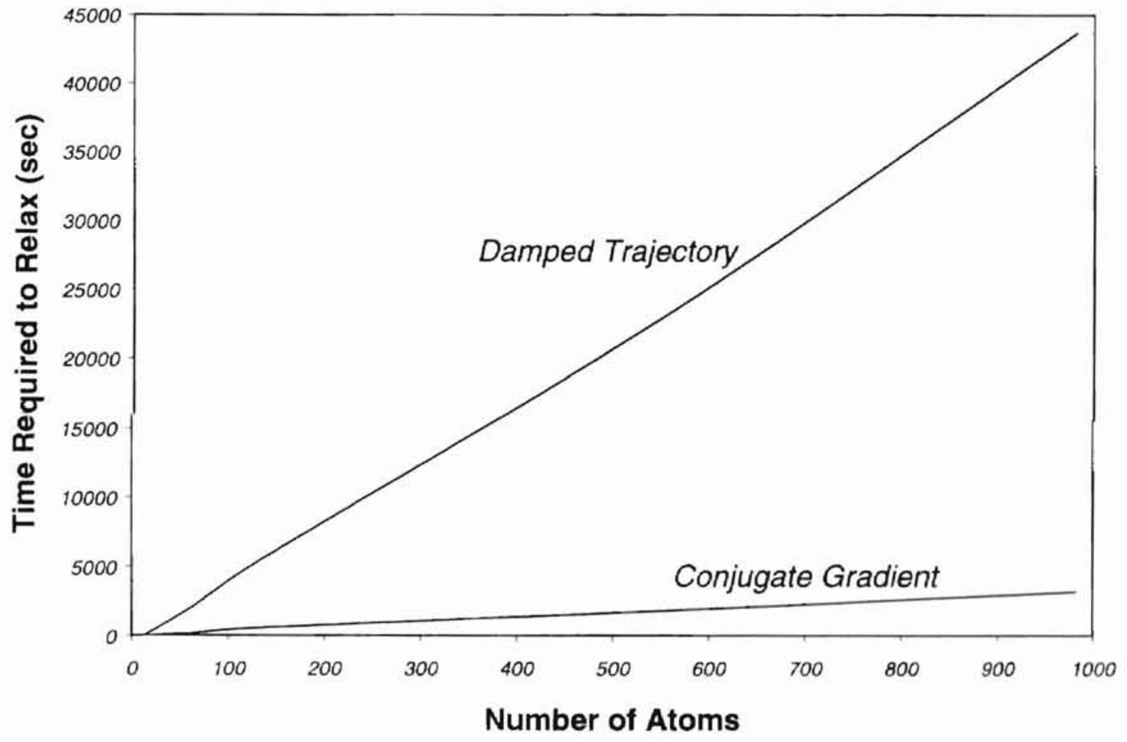

Figure 6.2 – Comparison of the Damped Trajectory and Conjugate Gradient Methods

As can be seen, the conjugate gradient method can converge to the desired minimum much more efficiently than the damped trajectory method for the MEAM. For 1000 atoms, the time required to relax has been reduced by a factor of approximately nine. Similar improvement is to be expected for any potential model where determination of forces on atoms is significantly slower than determination of the potential of the system of atoms.

CHAPTER 7

PARALLEL ALGORITHM DESIGN FOR MD

7.1 Introduction

The history and development of the application of parallel processing to molecular dynamics simulation has been discussed in Chapter 2. As was discussed, there are three main schools of thought in this area, each of which is best suited for certain hardware and certain applications – vector processing, data-level single-instruction-multiple-data (SIMD) processing, and procedure-level multiple-instruction-multiple-data (MIMD) processing. Even though each of these has been applied to molecular dynamics, MIMD methods offer the greatest potential for accelerated simulation times.

This fact is both fortunate and unfortunate. It is fortunate because a parallel processing environment which is well-suited for MIMD is inexpensive to build. It is unfortunate because programming for and building such an environment is difficult. However, if one is willing to invest the time and has the required patience, the rewards are well worth the effort, as this chapter will demonstrate.

7.2 The Parallel Computing Environment

Parallel processing is an entirely new endeavor for our research group. In order to test this new approach while minimizing cost, existing resources were reconfigured to support both parallel and sequential processing. Figure 7.1 illustrates this new computing environment.

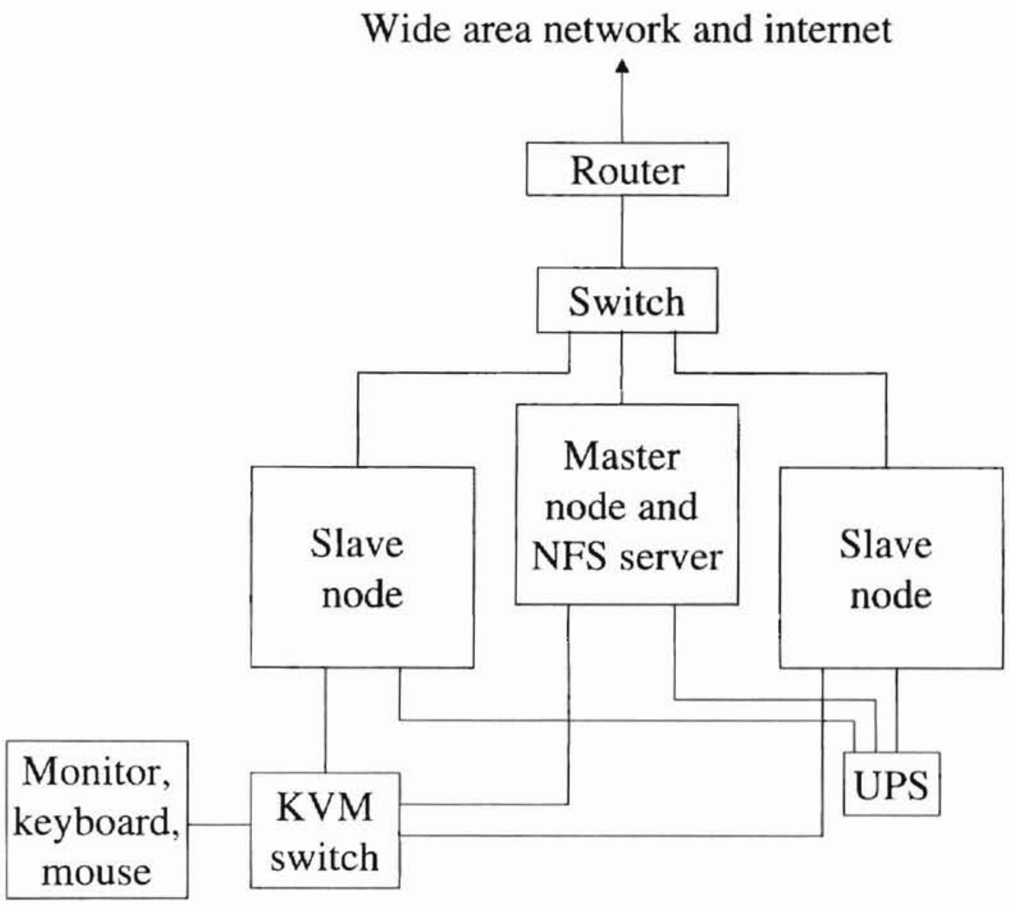

Figure 7.1 – Configuration of the Parallel Processing Environment

Three Digital Alpha 500au workstations were clustered with a 100 Mbps Ethernet switch. Connection of the master node to the wide-area-network and internet was made through a router for security reasons. Similarly, an encrypted terminal protocol known as secure shell (ssh) was used for remote login instead of telnet. Due to these security considerations, a low-level unsecure protocol, remote shell (rsh) was able to be used for interprocess communication. A partition on the hard drive of the master node was shared among the slave nodes via networked file system (NFS). The operating system used was the Redhat distribution of Linux, a free Unix clone. Power to the nodes was provided through a uninteruptable power supply (UPS) to avoid problems due to power loss. Also, a single monitor, keyboard, and mouse was shared among the three computers with a keyboard-video-mouse (KVM) switch.

A free implementation of the Message Passing Interface (MPI) standard, known as MPICH, was used. MPICH is a set of libraries which provide the names, calling sequences, and results of subroutines to be called from C, C++, or Fortran programs to exchange messages between processes. MPICH also provides functionality to start, abort, and manage multiple processes on multiple machines, and various tools for testing and performance analysis. These testing procedures were used to assure that the cluster was in working order.

7.3 Design and Performance of the Parallel Algorithm

As discussed in Chapter 2, the optimum MIMD-style parallel domain decomposition method for pairwise potential-based molecular dynamics (and the *n*-body problem in general) is well-documented in the literature. While no specific discussion of domain decomposition techniques for non-pairwise potentials (e.g. the MEAM) was found in the literature, the cases are similar enough for n-body techniques to be effective. As compared to pairwise potentials, the MEAM additionally requires propagating electron density values.

Briefly summarizing the discussion in Chapter 2, the most efficient approach for MIMD-style algorithm design is to use neighbor lists and the linked cell method. Ideally, each cell is assigned to its own node within the cluster. If enough nodes are not available, multiple contiguous nodes may be alternatively be assigned to a single node. This *dynamic* domain decomposition technique, when properly implemented, will result in the fastest possible computation. However, proper implementation is difficult – load balancing must be considered (density variation in the system may overload certain nodes), and the topology of the model must be carefully established to minimize communication requirements. Also, algorithms running on nodes which represent cells at the boundaries or corners of the simulation space must be special-cased.

The entire purpose of a sophisticated domain decomposition technique such as described above is to minimize message passing requirements, the slowest component of a typical distributed cluster. With dynamic domain decomposition, processes need only communicate with other processes which are topological neighbors. All-to-all messaging (situations where data on one process is needed by all other processes) is slow. The dynamic domain decomposition method avoids this bottleneck, but it has an additional tacit requirement to ensure its success – a reasonably large number of nodes. A considerable amount of "overhead" is incurred with the distributed linked cell method and with load balancing routines. That is, a considerable amount of computation is done that is not directly related to the essence of molecular dynamics simulation, but is intended to facilitate and speed up the essential functions of the program by minimizing message passing. When the number of nodes in the distributed computing cluster is small, this overhead dominates, and renders the dynamic decomposition method ineffectual. With a small number of nodes, the simulation space cannot be efficiently subdivided.

The distributed computing cluster used in this study has only three nodes. This is far below the number in a typical cluster – most clusters used for high-performance computing have as a minimum 32 nodes, and typically have 128 or more. As a result, the conventional approaches, including the dynamic domain decomposition method, require rethinking.

In this study, a *static* domain decomposition technique was used. This technique is mentioned only briefly in the literature because it requires all-to-all communication, and is therefore a very poor choice for implementation on large clusters. On a small, three node cluster, it is the best choice. With only three nodes, the simulation space cannot be divided to make a process independent of other processes, and all-to-all communication cannot be avoided.

The static domain decomposition technique assigns a fixed set of atoms to each process, as opposed to assignment of a fixed region of space (as with dynamic techniques). Because atoms may be arranged in any unpredictable arrangement, atoms on a given process may bond with atoms on any other process. For this reason, all-to-all communication is necessary. However, with only three nodes in the cluster, each node is sufficiently busy that the nodes, typically, will not be "starved for data", a typical situation with all-to-all communication in large clusters.

The static domain decomposition technique requires that each node integrates equations of motion, compute bond lists, potentials, forces, screening factors, etc., for only its own set of atoms. Bonds to any atom in the system may occur – therefore, each process needs to know the current coordinates of and electron densities at each atom in the system.

Figures 7.2 and 7.3 show, in a coarse sense, process flow of the root and slave algorithms. The master (or root) process performs most of the initialization, starts processes on the slave nodes, and propagates all necessary data to the slaves. After the initialization stage, all nodes (root and slaves) operate more or less on an equal basis, each doing all necessary computation for its own subset of atoms, and exchanging coordinates and densities when necessary. Throughout the simulation, the root process performs all console and file I/O. While MPI and NFS will route any slave-based console or file I/O back to the root process, this is accomplished at the cost of additional network activity – therefore, this is avoided.

- Initialization
  - Initialize MPI, start processes on slave nodes
  - Process command line parameters
  - Process specified input file
  - Allocate memory
  - Create workpiece and tool
  - Load MEAM parameters from input file
  - Propagate parameters and atom coordinates to slave processes
  - Determine an even division of atoms and assign a subset to each proceses
- Relaxation (conjugate gradient)
  - Write pre-relax atom coordinates to a file
  - Determine bond list (intra-workpiece bonds only, local atoms only)
  - Determine screening factors (local atoms only)
  - Compute potential (local atoms only)
  - Assemble total system potential at root
  - Set initial search direction for local atoms to the negative of the gradient
    - Locate an interval containing a minimum (using parallel potential computation as above)
    - Reduce the interval, converging to the minimum to within a specified tolerance
    - If the new minimum is sufficiently close to the previous value, relaxation is complete
    - Determine new search direction for local atoms , resetting to the local gradient after every five iterations
  - Write post-relax atom coordinates to a file
  - End relax
- Main Simulation
  - Insert thermal energy into lattice (local atoms only)
  - Determine bond list (all workpiece and tool atoms, local atoms only)
  - Determine screening factors (local atoms only)
    - Integrate equations of motion (local atoms only)
    - Propagate new atom coordinates to all processes
    - Determine bond list (all workpiece and tool atoms, local atoms only)
    - Determine screening factors (local atoms only)
    - (Optional) Compute kinetic and potential energies of system to check energy conservation
    - Write atom coordinates at user-specified intervals
    - Reset local peripheral atom momentums if necessary
    - If local process has tool atoms, move them according to specified indentation speed
    - Repeat until simulation is complete
  - Finalize MPI, kill slave processes, free memory
- End

Figure 7.2 – Process Flow of the Root Algorithm

- Initialization
  - Initialize MPI
  - Allocate memory
  - Receive parameters and atom coordinates from root
  - Receive atom assignments from root
- Relaxation (conjugate gradient)
  - Determine bond list (intra-workpiece bonds only, local atoms only)
  - Determine screening factors (local atoms only)
  - Compute potential (local atoms only)
  - Report local potential to root
  - Set initial search direction for local atoms to the negative of the gradient
    - Locate an interval containing a minimum (using parallel potential computation as above)
    - Reduce the interval, converging to the minimum to within a specified tolerance
    - If the new minimum is sufficiently close to the previous value, relaxation is complete
    - Determine new search direction, resetting to the local gradient after every five iterations
  - End relax
- Main Simulation
  - Insert thermal energy into lattice (local atoms only)
  - Determine bond list (all workpiece and tool atoms, local atoms only)
  - Determine screening factors (local atoms only)
    - Integrate equations of motion (local atoms only)
    - Propagate new atom coordinates to all processes
    - Determine bond list (all workpiece and tool atoms, local atoms only)
    - Determine screening factors (local atoms only)
    - Reset local peripheral atom momentums if necessary
    - If local process has tool atoms, move them according to specified indentation speed
    - Repeat until simulation is complete
  - Finalize MPI, free memory
- End

Figure 7.3 – Process Flow of the Slave Algorithm

Parallel programming for a distributed computing cluster is tedious. There is no integrated developing environment or debuggers for message-passing style parallel programming. In addition to the normal potential for bugs that any complex program has, dividing the problem among separate processes each with mutual dependence has many additional potential pitfalls. It is difficult to view data which is stored in a slave process for debugging purposes. When a problem is encountered, the programmer is frequently forced to manually log and review program output and intermediate calculation results. Computer scientist researchers in the field of high performance cluster computing are working to develop new and better tools to facilitate parallel application development, but for now, things must be done the hard way.

Figure 7.4 illustrates the total elapsed time required for the sequential and parallel algorithms for various system sizes. Measured times are for systems of iron atoms, and simulation time are 500 times steps with an integration resolution of 0.1 time steps. The sequential algorithm is run on the master node (with no spawning of slave processes) to allow meaningful comparison.

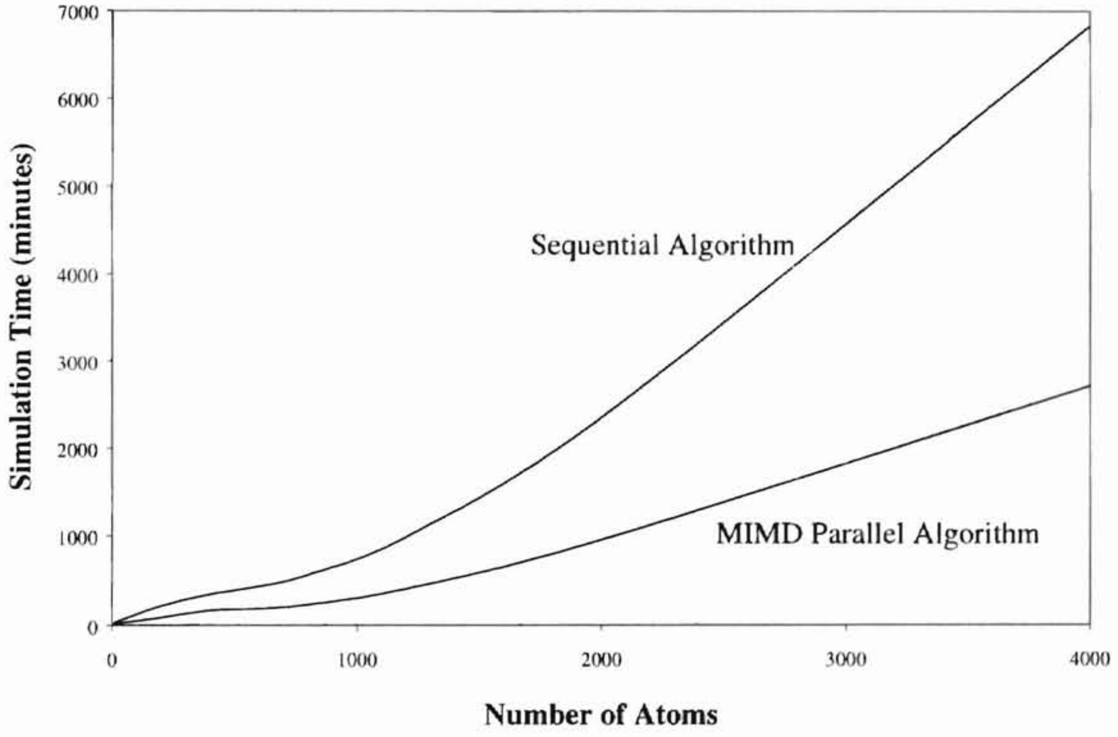

Figure 7.4 – Comparison of Run Times for Parallel and Sequential Algorithms

The time savings accomplished with the parallel algorithm are significant. For a system of 4000 atoms, the sequential algorithm takes almost 4.5 days – with the parallel algorithm, just over 2 days, a speed increase factor of roughly 2.25. With three nodes of equal capacity doing the work, a factor of 3 is the theoretical maximum performance gain. This is not realizable because of the additional time required to pass messages among the processes. However, this theoretical maximum can be approached, as in this case, with efficient and intelligent algorithm design.

CHAPTER 8

MD MODELING OF NANOMETRIC INDENTATION WITH THE MEAM

8.1 Introduction

The purpose of the work summarized in the preceding chapters was to make it possible to apply the Modified Embedded Atom Method to molecular dynamics modeling of nanoindentation, and achieve results in a reasonable amount of time. In the absence of these developments, namely the use of the conjugate gradient method and parallel processing, a typical molecular dynamics simulation involving a few to several thousand atoms would take over a week of computer time with the MEAM, using existing computing resources. With these new techniques, less than two days are required.

While this is significant, it is of little use if the resulting simulations do not model reality. While this judgement is frequently not clear-cut due the difficulty involved with experimental investigations on the nanometric scale, there are validation techniques available. The use of some of these techniques is summarized in this chapter in order to assess the usefulness of the MEAM for molecular dynamics simulation of nanoindentation. While the indentation process is definitely of interest in and of itself, the successes and/or failures of the MEAM for indentation can serve as a test case, and guide application of the MEAM to other manufacturing processes.

The first method for judging the validity of the MD simulations conducted in this study is viewing of animations of the simulations. General trends and behavior may be observed in this way and evaluated using common sense, materials science expertise, and by comparing to results in the literature. This method is used here, and snapshots of animations of the MD simulations are included to illustrate the points made.

Another method used is measurement of the theoretical strength of the indented crystals. By measuring the average force on the indenter from initial penetration until immediately before retraction, and dividing by either the contact or projected area of the indenter, the strength of the crystal can be computed. These values can then be compared to the theoretical strength of dislocation-free crystals, given by Hertzberg [22] as:

$$strength = {}^{G}\!/_{2\pi} \tag{41}$$

where G is the shear modulus of the material.

8.2 Indentation with the MEAM at the Interface

In his extension of the MEAM [39], Baskes formulates two models, one for pure crystals and one for alloys. With his parametric formulation for binary systems, interactions between many different materials was described. Because one of the deficiencies of MD modeling of manufacturing processes is the inability to accurately model interactions between unlike materials, it was hoped that the MEAM could accomplish this. This section describes the extension of the MEAM model of alloys to interactions between physically separate pure crystals of different materials.

Many combinations of tool and workpiece materials were tried and the majority were found unsuitable. To illustrate why, refer to Figure 8.1 which shows the indentation of iron with diamond, and Figure 8.2 which shows the indentation of iron with tungsten.

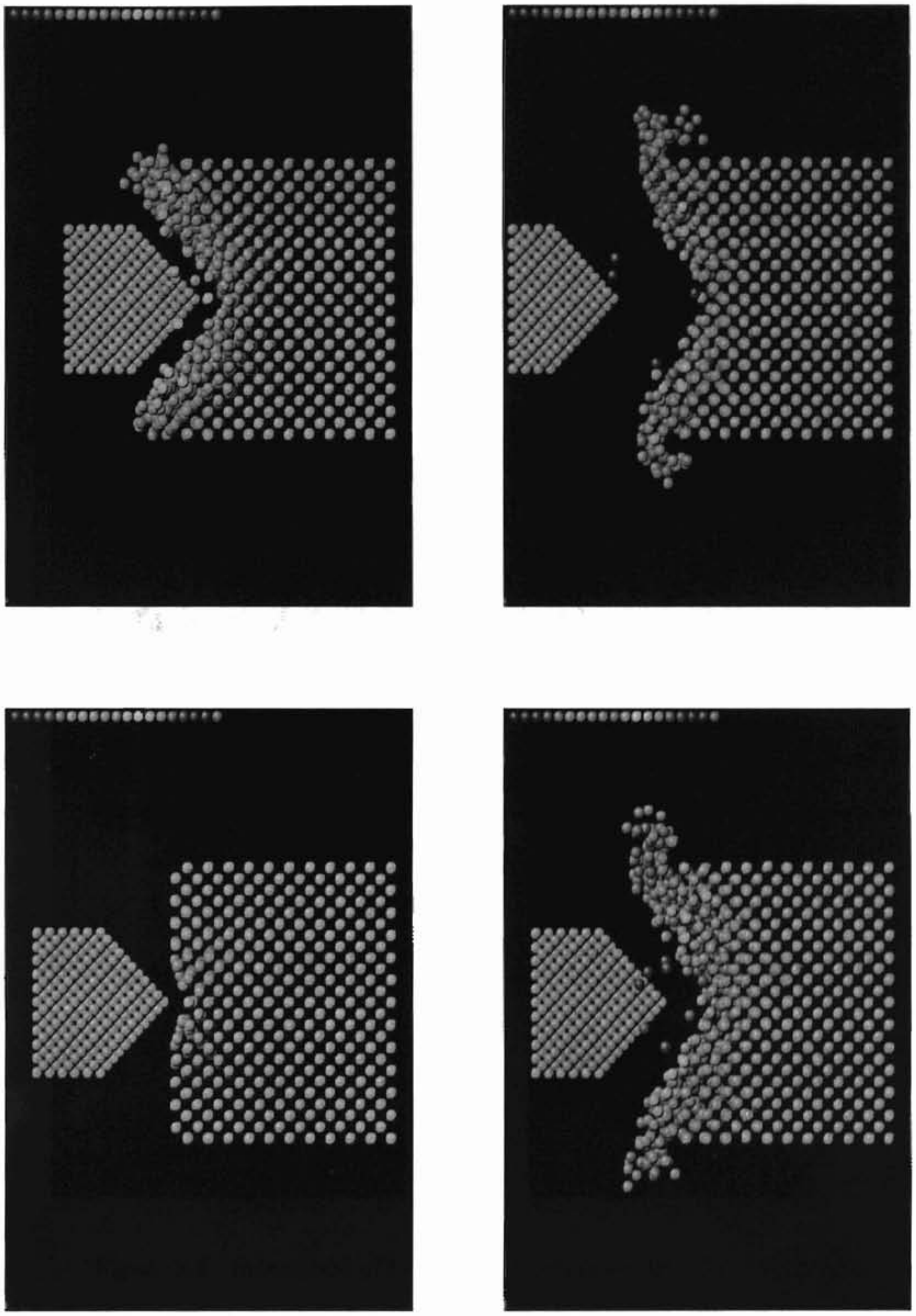

Figure 8.1 - Indentation of Fe with Diamond using MEAM at the Interface

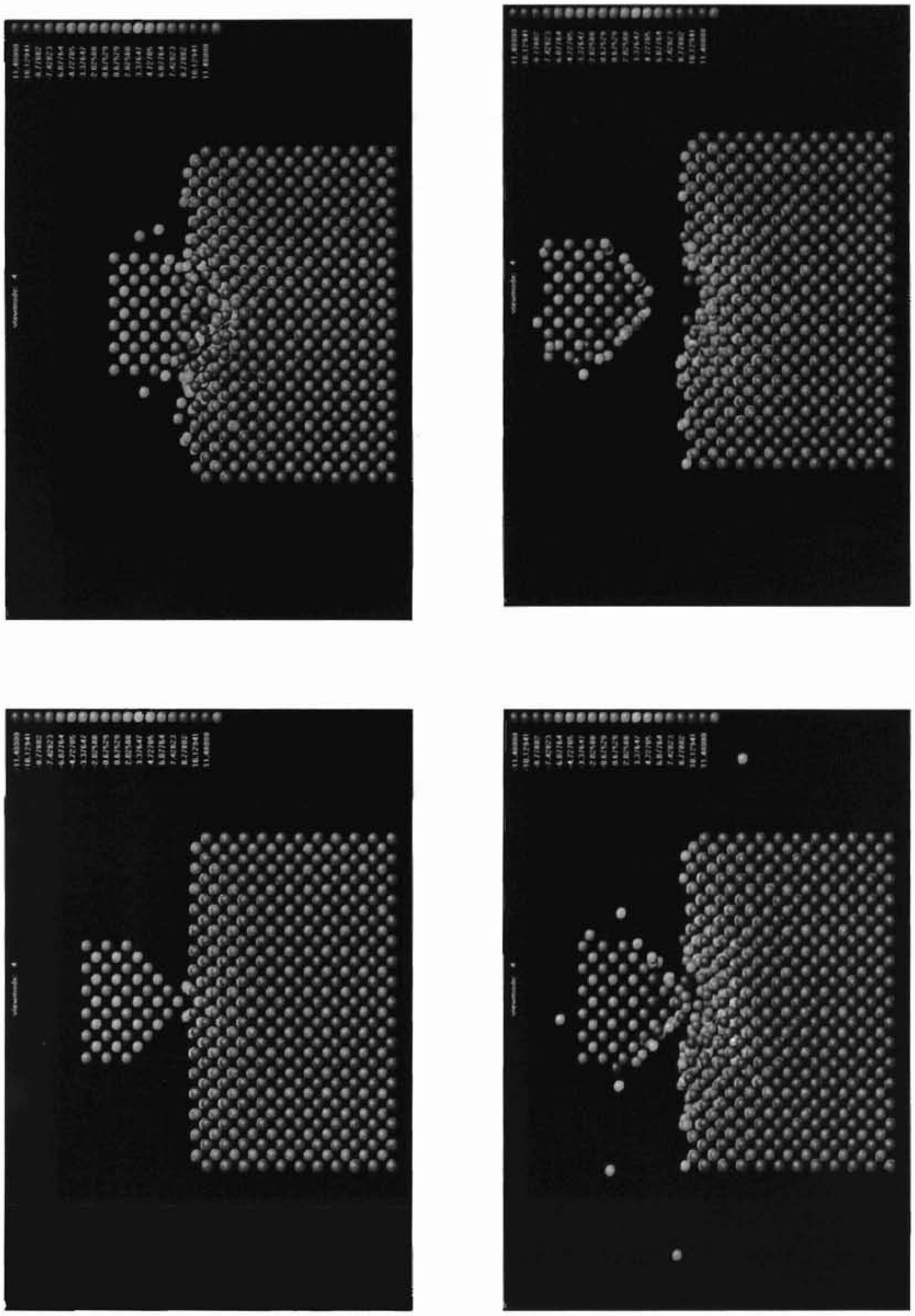

Figure 8.2 - Indentation of Fe with Tungsten using MEAM at the Interface

Near room temperature, these combinations of materials are chemically incompatible – that is, there should be little or no interaction. Indentation of the same workpiece by an infinitely hard indenter (this is the case here, since the tool is not allowed to deform), should give the same result regardless of the material of the indenter, as long as the materials used are chemically non-reactive at low temperatures.

When indenting iron with diamond, a strong long-distance repulsion exists which results in a crater larger than the tool. Interaction occurs at a considerable distance. Due to this strong repulsion, the calculated hardness of the work materials is two orders of magnitude too great.

When indenting with tungsten, the opposite situation occurs. A strong attraction occurs which results in too low of a calculated hardness. The work atoms jump into contact with the tool, and adhere to the tool after retraction. Negligible pile up occurs and a small crater is formed due to the fact that the attraction between tool and workpiece pulls the work atoms back into place as the tool is retracted.

These two extreme cases, either strong repulsion or strong attraction, occurred with the majority of combinations of materials used. While these interactions may or may not be accurate, it can be concluded that if workpiece effects are of primary interest, the binary formulation of the MEAM for work/tool interaction is inappropriate. However, application to alloys may present opportunities not available with other potential models, and work in this direction is recommended.

8.3 Indentation with Morse at the Interface

By using the MEAM at the interface between the indenting tool and the workpiece, it became clear that a chemically neutral potential interaction between the tool and workpiece is desirable. In order to focus on the behavior of the workpiece as modeled by the MEAM, the Morse potential was implemented at the interface. As described in previous chapters, this has been the traditional approach when considering effects between unlike materials. Morse parameters were used to give an insignificant attraction between atoms, and only a short-range repulsion. In essence, this Morse model causes atoms to interact in a fashion similar to how macro-scale bodies interact – there is no interaction beyond a simple "shoving" action when bodies contact each other.

This modification to the program was straightforward, due to the simplicity of the Morse potential. Also, a slight speed increase results. While intra-workpiece bonds dominate, the work-tool bonds which do develop take an insignificant amount of processing time with the Morse potential as compared with the MEAM.

Table 8.1 summarizes computed strengths of some pure crystals and shows how they compare to theoretical values.

Table 8.1 – Comparison of MD and Theoretical Strengths for Various Materials

Material	Theoretical Strength (GPa)	Strength from MD (GPa)
FCC		
Nickel	*33.4*	*28.2*
Copper	*19.1*	*20.7*
Silver	*12.6*	*11.6*
BCC		
Iron	*31.8*	*12.3*
Molybdenum	*54.1*	*15.0*
Chromium	*18.4*	*4.5*

The strength values from MD experiments are based on contact area. For the FCC materials, the MD values are reasonable, but for the BCC materials, the values are considerably too low. This trend holds for other FCC and BCC materials.

Figures 8.3 and 8.4 show snapshots of the simulations for copper and iron. A large amount of elastic deformation and recovery is apparent with iron, and little pile up occurs. With copper, much less elastic deformation is apparent, and a larger crater and more pile up results. Also with copper, an excessive amount of subsurface deformation occurs. Deformation to this extent does not seem reasonable. These trends are found with other FCC and BCC materials as well.

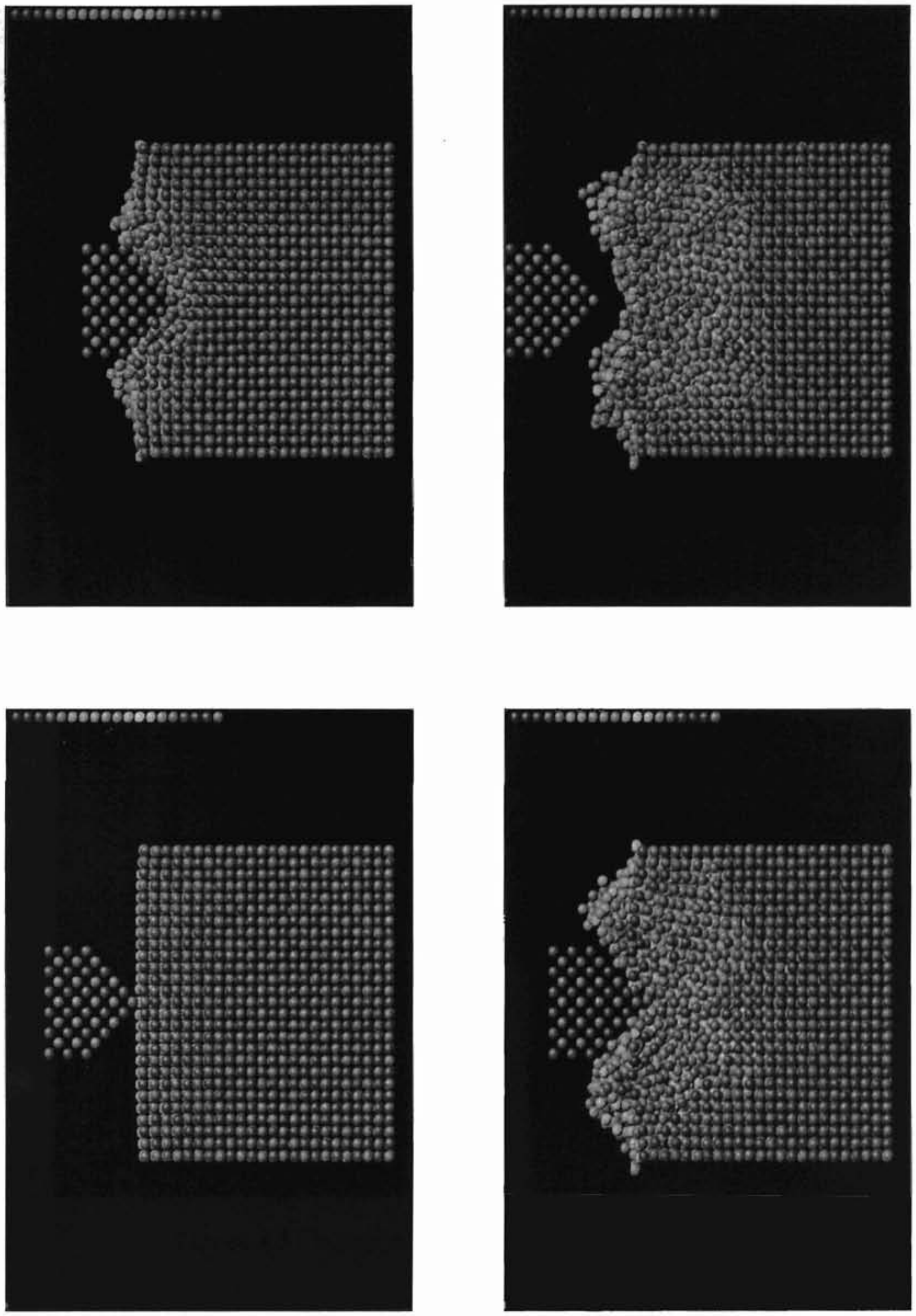

Figure 8.3 - Indentation of Cu using Morse at the Interface

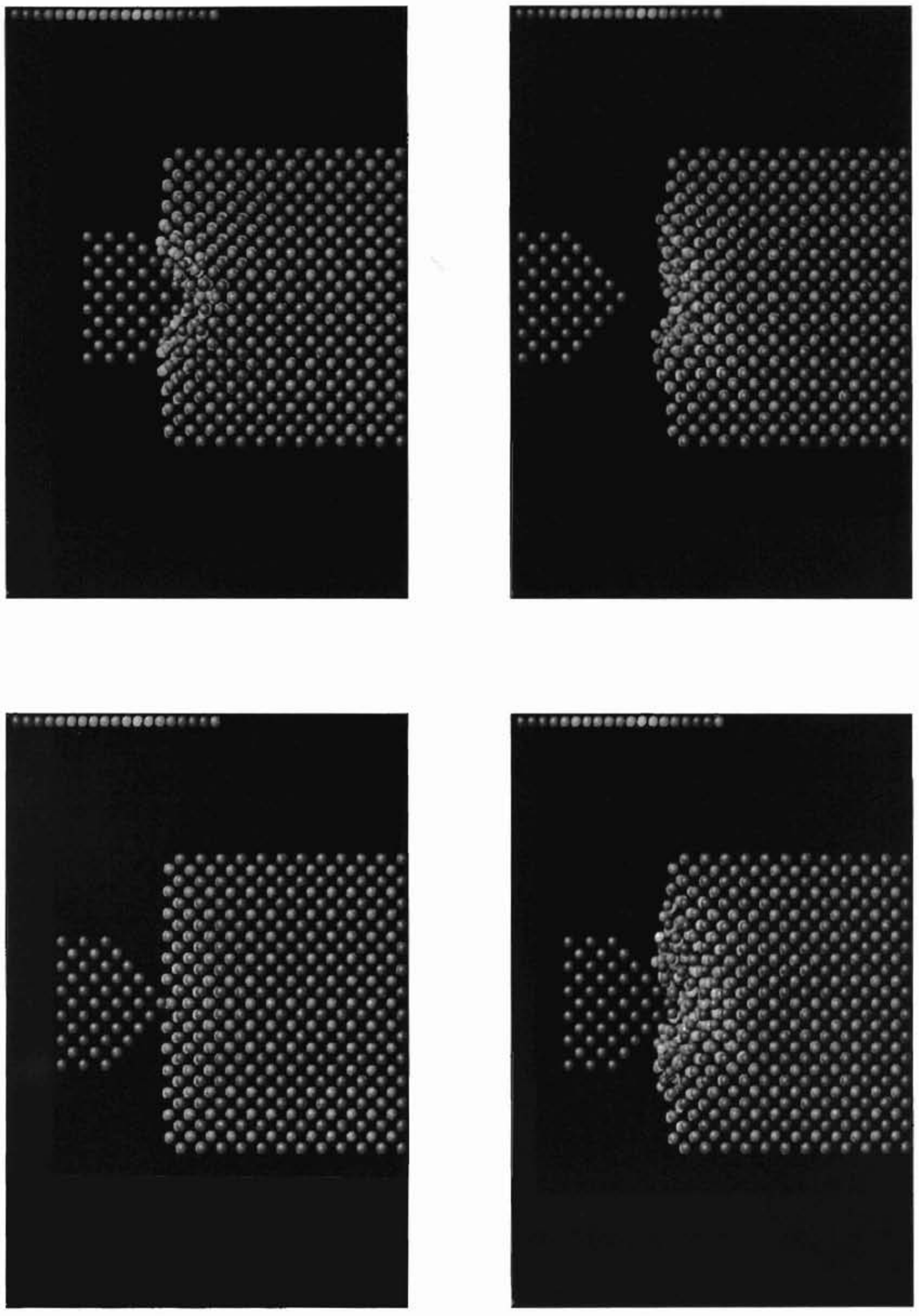

Figure 8.4 - Indentation of Fe using Morse at the Interface

CHAPTER 9

CONCLUSIONS

9.1 Summary of Results

Understanding of the mechanisms of material removal at small depths of cut, as is typical in ultra precision machining and grinding, is limited by technological difficulties in measuring forces and observing the chip formation process. Molecular dynamics (MD) simulation has the potential to sidestep these difficulties and provide insight into these issues in an inexpensive, reliable, and repeatable manner. However, even with advanced computing resources, MD simulation requires a great deal of computational time, especially when sophisticated potential models are used, such as the Modified Embedded Atom Method (MEAM).

The Morse potential, while easily implemented and relatively fast, has a number of deficiencies, primarily the inability to accurately model BCC materials. The MEAM, as developed by Baskes [39], offers the potential to avoid these deficiencies, and allow the modeling of a variety of FCC, BCC, HCP, and diamond cubic materials. The MEAM also accounts for interactions between unlike materials.

To reduce the processing time required by molecular dynamics simulation, a multiple-instruction-multiple-data (MIMD) style parallel processing algorithm has been designed and implemented. A static domain decomposition technique was used, which assigned a fixed set of atoms to each process in the cluster. All software development was done in such a manner as to allow easy extension to any potential model. Since application was on a three node cluster, all-to-all message passing was used – the standard methods of dynamic domain decomposition using process topology and load balancing is not effective with such a small cluster.

To test this new algorithm, computer programs to implement the MEAM have been developed and validated. Two versions have been developed – a sequential version and a parallel version. Both give identical results, confirming the validity of the parallel algorithm. The parallel version offers roughly a 2.25 times speed increase when applied to the MEAM using a three node distributed computing cluster consisting of Digital Alpha 500au workstations as compared to the sequential version running on a single workstation of the same type.

For the relaxation process, the conjugate gradient method has been found to operate much more quickly than an optimized damped trajectory method with the MEAM. This is due to the fact that the complexity of the MEAM causes computation of interatomic forces to be much more time consuming than computation of the total potential of the system. The conjugate gradient method requires infrequent evaluation of the local gradient of the potential hypersurface at the expense of addition determinations of system potential. This minimizes the need to call the forces function in the MEAM code, which is the most time consuming function.

In conjunction, these two time-saving techniques reduced simulation time from one week to less than two days for a typical MD experiment using the MEAM. With this usable simulation code, the MEAM was applied to indentation of pure crystalline materials at the nanometric level. Use of the MEAM formulation for interactions between unlike materials at the interface between the tool and workpiece resulted in unpredictable behavior which dominated the action in the workpiece. Regardless of the accuracy or inaccuracy of these interactions between unlike materials, it became immediately obvious that continued used of a chemically neutral form of the Morse potential at the interface was best for the study of nanoindentation and, most likely, manufacturing processes simulations in general.

Using Morse at the interface with the MEAM used within the workpiece, reasonable micro-hardness values were obtained for FCC materials. For BCC materials, hardness values were too low. Significant crater formation and pile up were observed with FCC metals, as was extensive and unrealistic subsurface deformation. With BCC materials, deformation was largely elastic, and elastic recovery prevented significant crater formation or pile up.

The superiority of the MEAM over the Morse potential has not been demonstrated. It seems that the deficiencies of the Morse potential and the strengths of the MEAM are not an issue for situations of interest to the manufacturing processes engineer. Indentation, cutting, grinding, and other material removal processes result in extreme deviation from the equilibrium lattice configuration and involve the input of huge amounts of thermal energy into the system of atoms. The embedded atom method family of molecular dynamics potential models are built on, in part, pure curve fitting to equilibrium conditions. Testing and validation methods of EAM methods in the literature do not involve such catastrophic deformation of the simulated material and energy input such as occurs during the typical material removal process. It is suggested that these are the reasons for the less than stellar performance of the MEAM.

9.2 Future Work

Application of the parallel processing techniques developed in this study to other potential models is recommended. With the construction of a larger distributed computing cluster, parallel implementation of the Morse potential would allow simulation of very large systems of atoms. This would allow investigation into a whole new class of problems, and could be accomplished with a minimum of effort. The MIMD parallel algorithm developed in this study has been written in a modular fashion, to allow easy extension to other potential models and other manufacturing processes, such as cutting or grinding.

Use of the conjugate gradient technique for relaxation is also recommended for any mathematically complex potential model. For any potential where computation of forces is significantly slower than computation of potential, the conjugate gradient method offers the potential for very large savings in computer time.

More work is recommended to reach a firm conclusion on the suitability of the MEAM for MD simulation of indentation and manufacturing processes in general. The mixed results found in this study make accurate commentary difficult. While it has been demonstrated that the binary formulation is not suitable for interactions between physically separate crystals, it may be useful for alloys.

REFERENCES

[1] Chadrupatla, T. R., Belegundu, A. D., Introduction to Finite Elements in Engineering,1997, Prentice Hall, New Jersey.

[2] Chandrasekaran, N., Length Restricted Molecular Dynamics (LRMD) Simulation of Nanometric Cutting, 1997.

[3] Komanduri, R., Chandrasekaran, N., Raff, L. M., "Molecular Dynamics (MD) Simulation of Uniaxial Tension".

[4] Daw, M.S., Baskes, M. I., "Embedded-Atom Method: Derivation and Application to Impurities, Surfaces, and other Defects in Metals", 1983, Phys. Rev. B., 29 12, pp 6443.

[5] Adams, J. B., Foiles, S. M., "Development of an Embedded-Atom Potential for a BCC Metal: Vanadium", 1989, Phys. Rev. B., 41 6 pp 3316.

[6] Baskes, M. I., "Modified Embedded-Atom Potentials for Cubic Materials and Impurities", 1991, Phys. Rev. B., 46 5, pp 2727.

[7] Slaets, F. W., Travieso, G., "Parallel Computing: A Case Study", 1989, Comp. Phys. Comm., 56, pp 63.

[8] Woods, M. B., Alford, C. O., "Scaleable Parallel Model Development and Performance Analysis for MIMD Molecular Dynamics Simulations", 1995, Comp. Elect. Engng., 21, 3, pp 159.

[9] Heyes, D. M., Smith, W., Inf. Q. Comput. Simulation Condensed Phases, 1987, 26, 68.

[10] Rapaport, D. C., Comp. Phys. Comm., 1988, 9, 1.

[11] Everaers, R., Kremer, K., "A Fast Grid Search Algorithm for Molecular Dynamcics Simulations with Short-Range Interactions", 1994, Comp. Phys. Comm., 81, pp. 19.

[12] Liem, S. Y., Brown, D., Clarke, J. H. R., "Molecular Dynamics Simulations on Distributed Memory Machines", 1991, Comp. Phys. Comm, 67, pp 261.

[13] Srinivasan, S. G., Ashok, I., Jonsson, H., Kalonji, G., Zahorjan, J., "Parallel Short-Range Molecular Dynamics Using the Adhara Runtime System", 1997, Comp. Phys. Comm., 102, pp 28.

[14] Srinivasan, S. G., Ashok, I., Jonsson, H., Kalonji, G., Zahorjan, J., "Dynamic Domain Decomposition Parallel Molecular Dynamics", 1997, Comp. Phys. Comm., 102, pp 44.

[15] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., Sunderam, V., <u>PVM – A User's Guide and Tutorial for Networked Parallel Computing</u>, 1997, Fourth Edition, MIT Press.

[16] Gropp, W., Lusk, E, Skjellum, A., <u>Using MPI: Portable Parallel Programming with the Message Passing Interface</u>, 1999, Second Edition, MIT Press.

[17] Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J., <u>MPI – The Complete Reference: Volume 1, The MPI Core</u>, 1999, Second Edition, MIT Press.

[18] Gropp, W., Huss-Lederman, S., Lumsdaine, A., Lusk, E., Nitzberg, B., Saphir, W., Snir, M. <u>MPI – The Complete Reference: Volume 2, The MPI Extensions</u>, 1999, First Edition, MIT Press.7

[19] Alder, B., Wainwright, T., "Studies in Molecular Dynamics I. General Method", 1959, J. Chem Phys., 31, pp 459.

[20] Alder, B., Wainwright, T., "Studies in Molecular Dynamics II. Behavior of a Small Number of Elastic Spheres", 1960, J. Chem Phys., 33, pp 1439.

[21] Hoover, W. G., De Groot, A. J. Hoover, C. G., Stowers, I. F., Kawai, T., Holian, B. L., Boku, T., Ihara, S., Belak, J., "Large Scale Elastic-Plastic Indentation Simulations Via Non-Equilibrium Molecular Dynamics", 1990, Phys. Rev. A, 42, pp 5844.

[22] Hertzberg, R. W., <u>Deformation and Fracture Mechanics of Engineering Materials</u>, 1996, Fourth Edition, John Wiley & Sons, Inc.

[23] Belak, J., Boercker, D. B., Stowers, I. F., "Simulation of Nanometer Scale Deformation of Metallic and Ceramic Surfaces", 1993, MRS Bull., 18, pp 55.

[24] Landman, U., Luedtke, W. D., Burnham, N. A., and Colton, R. J., "Atomistic Mechanisms and Dynamics of Adhesion, Nanoindentation, and Fracture", 1990, Science, 24, pp 454.

[25] Landman, U., Luedtke, W. D., and Ringer, E. M., "Atomistic Mechanisms of Adhesive Contact Formation and Interfacial Processes", 1991, Wear, 153, pp 3.

[26] Rentsch, R., Inasaki, I., "Molecular Dynamics Simulation for Abrasive Processes", 1994, Annals CIRP, 43, pp 327.

[27] Brenner, D. W., Sinnott, S. B., Harrison, J. A., Shenderova, O. A., "Simulated Engineering of Nanostructures", 1996, Nanotechnology, 7, pp 161.

[28] Yan, W., Komvopoulos, K., "Three-Dimensional Molecular Dynamics Analysis of Atomic-Scale Indentation", 1998, J. of Trib., 120, pp 385.

[29] Komanduri, R., Chandrasekaran, N., Raff, L. M., "MD Simulation of Indentation and Scratching of Single Crystal Aluminum", 2000, Wear, 240, pp 113.

[30] Johnson, R. A., "Empirical Potentials and Their Use in the Calculation of Energies of Point Defects in Metals", 1973, J. Phys. F., 3, pp 295.

[31] Stott, M. J., Zaremba, E., "Quasiatoms: An Approach to Atoms in Nonuniform Electronic Systems", 1980, Phys. Rev. B., 22, pp 1564.

[32] Puska, M. J., Nieminen, R. M., Manninen, M., "Atoms Embedded in an Electron Gas: Immersion Energies", 1981, Phys. Rev. B., 24, pp 3037.

[33] Daw, M. S., Baskes, M. I., "Embedded Atom Method: Derivation and Application to Impurities, Surfaces, and Other Defects in Metals", 1984, Phys. Rev. B., 29, pp 6443.

[34] Foiles, S. M., Baskes, M. I., Daw, M. S., "Embedded Atom Method Functions for the FCC Metals Cu, Ag, Au, Ni, Pd, Pt, and Their Alloys", 1986, Phys. Rev. B., 33, pp7983.

[35] Baskes, M. I., "Application of the Embedded Atom Method to Covalent Materials: a Semiempirical Potential for Silicon", 1987, Phys. Rev. Let., 59, 23, pp 2666.

[36] Oh, D. J., Johnson, R. A., "Simple Embedded Atom Method Model for FCC and HCP Metals", 1988, J. Mater. Res., 3, 3, pp 471.

[37] Johnson, R. A., Oh, D. J., "Analytic Embedded Atom Method Model for BCC Metals"" J. Mater. Res., 4, 5, pp 1195.

[38] Baskes, M. I., Nelson, J. S., Wright, A. F., "Semiempirical Modified Embedded Atom Potentials for Silicon and Germanium", 1989, Phys. Rev. B., 40, pp 6085.

[39] Baskes, M. I., "Modified Embedded Atom Potentials for Cubic Materials and Impurities", 1991, Phys. Rev. B., 46, pp 2727.

[40] Raff, L. M., <u>Molecular Dynamics Modeling</u>, Lecture Notes.

[41] Riley, M. E., Coltrin, M. E., Diestler, D. J., 1988, J. of Chem. Phys., 88, pp 5943.

[42] Baskes, M. I., "Determination of Modified Embedded Atom Method Parameters for Nickel", 1997, Mat. Chem. Phys., 50, pp 152.

[43] Hagan, M. T., Demuth, H. B., Neural Network Design, 1996, First Edition, PWS Publishing Company.

VITA

David L. Stokes

Candidate for the Degree of

Master of Science

Thesis: PARALLEL PROCESSING OF MOLECULAR DYNAMICS SIMULATION IN A DISTRIBUTED COMPUTING ENVIRONMENT AND APPLICATION TO THE MODIFIED EMBEDDED ATOM METHOD AND NANOINDENTATION

Major Field: Mechanical Engineering

Biographical:

Education: Graduated from Booker T. Washington High School in 1992; received Bachelor of Science degree in Mechanical Engineering from Oklahoma State University in December 1998; completed the requirements for the Master of Science degree in Mechanical Engineering at Oklahoma State University in December 2000.

Experience: Research Assistant for Dr. Ranga Komanduri at Oklahoma State University from May 1993 – December 2000.