MULTIRESOLUTION MOLECULAR MECHANICS: DYNAMICS, ADAPTIVITY, AND IMPLEMENTATION

by

Emre Biyikli

B.S. in Mechanical Engineering, Koc University, 2007

M.S. in Mechanical Engineering, Koc University, 2009

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2015

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Emre Biyikli

It was defended on

October 16, 2015

and approved by

Tevis D. Jacobs, PhD, Assistant Professor, Department of Mechanical Engineering and Materials Science

Jeen-Shang Lin, PhD, Associate Professor, Department of Civil and Environmental Engineering

Guofeng Wang, PhD, Associate Professor, Department of Mechanical Engineering and Materials Science

Dissertation Director: Albert C. To, PhD, Associate Professor, Department of Mechanical Engineering and Materials Science

MULTIRESOLUTION MOLECULAR MECHANICS: DYNAMICS, ADAPTIVITY, AND IMPLEMENTATION

Emre Biyikli, PhD

University of Pittsburgh, 2015

Full atomistic Molecular Dynamics (MD) simulations are very accurate but too costly; however, atomistic resolution is not actually required everywhere in many problems. For this reason, a concurrent atomistic/continuum coupling method called Multiresolution Molecular Mechanics (MMM) has been developed. The method employs atomistic resolution in the localized regions of interest and coarser continuum description elsewhere. A number of such multiscale methods have been developed but they fail to demonstrate consistency, accuracy, adaptivity, flexibility, and efficiency all in one. The goal of this research is thus to develop a multiscale method that possesses these properties to outperform the MD method by 1) formulating new dynamics equations under the MMM framework, 2) developing an adaptivity scheme, and 3) implementing efficient algorithms for the method. First, the derivation of the governing MMM equations from a Hamiltonian that approximates the energy of the original system is presented. Second, the adaptivity analysis of the MMM method is presented. Refinement and coarsening mechanisms of the adaptivity scheme are described in detail and the step-by-step procedures are outlined. Third, the implementation and efficiency of the MMM software is presented. The structure of the software along with the associated technologies is introduced. Many improvements that contribute to the efficiency of the MMM software are described and demonstrated through benchmark tests. The efficiency of the software is found to be as good as one of the best state-of-the-art MD codes, i.e., LAMMPS. The speed-up of the code in proportion to reduction in the rep atom ratio is demonstrated. The scalability of the software is demonstrated and competing effects of multiscale modeling and parallelization is discussed. The dynamics, adaptivity, and efficiency of the method are demonstrated by numerical examples including wave and crack propagation, dislocation glide, nanoindentation, and modal analysis in 1/2/3 dimensions. All results agree well with the true full atomistic solutions. Ultimately, the MMM method demonstrates an improvement of 6.3 – 8.3 times in efficiency over MD method by means of a combined reduction in simulation time and number of processors. In conclusion, this dissertation shows that the MMM method is consistent, accurate, flexible, and efficient.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I am very grateful to my advisor Dr. Albert C. To for his endless support over the past five years. I deeply appreciate his precious efforts with me and my work at various stages of my academic life. He has always been very understanding, which in turn provided a peaceful environment that let me focus on my research. I especially thank him for his care to motivate me for my research; and for being indulgent when I had different research interests. I also thank him being so nice and friendly. All in all, I sincerely appreciate his supervision and guidance.

I am also indebted to my committee members, Dr. Tevis D. Jacobs, Dr. Jeen-Shang Lin, and Dr. Guofeng Wang, for their valuable time and insightful suggestions. It has been a privilege for me to have these scholars all in my committee.

I would like to thank to all members of the Computational Nanomechanics Group, especially to those I spent many years with: Qingcheng Yang, Dr. Pu Zhang, and Dr. Yao Fu. I thank them for the inspiring discussions we had about our work, careers, and lives. I feel lucky for having such good friends.

Finally, I want to thank to my wife Senem Güler-Bıyıklı; my father İffet Bıyıklı, my mother Fatma Bıyıklı, and my brother Eray Bıyıklı in Konya; my sister Ebru Bıyıklı in Ankara; and my father-in-law Ahmet Güler, my mother-in-law Samiye Güler, and my brother-in-law Rasim Güler in İstanbul. Their love was an important driving force to accomplish my degree. I am so much indebted to them and I appreciate their endless support.

1.0 INTRODUCTION

A class of multiscale methods assemble atomistic and continuum scales together in order to take advantage of both approaches. Continuum is the most traditional and common scale that is analytically and computationally investigated in depth. It assumes that matter completely fills the space it occupies, is continuously distributed, and is infinitely divisible into very small pieces. Physical principles governing the continuum are conservation laws of mass, momentum, and energy, as well as a constitutive law to govern the material behavior. Owing to the extensive research, continuum theories are well established and thus robust, highly efficient, and accurate; however, they usually fail at the atomistic scale for several reasons. To start with, they ignore the inherent discrete structure of the material, particles such as atoms and electrons, so that they leave the physical underpinning weak [1]. Another reason is the fact that different physical phenomena are in play at each scale; for example, quantum effects are not considered at larger scales but they must be accounted for at the nanoscale [2]. A third reason is because the continuum does not allow the material length to vanish to zero; such as at a crack opening [3]. In rare cases where continuum methods are able to handle the physical phenomena at the atomistic scale, they lack the precision that the atomistic methods are ready to offer.

The advent of micro/nanoscale technologies such as Microelectromechanical Systems (MEMS) and Nanoelectromechanical Systems (NEMS) devices provide the scientist with the tools to investigate matter at nanoscale. For example, high-resolution microscopes are able to detect atoms individually [4]. In addition, computational tools such as faster processors and supercomputers are made available to scientists. The recent and rapid progresses present nanotechnology as the new trend and triggers vast research activity in development and application at the smaller scales [5]. A considerable amount of these activities are devoted to computational methods or, more precisely, atomistic methods. Compared to continuum, atomistic methods are relatively new and require further exploration since their inherent discrete structure does not fit in the traditional understanding. The most attractive feature of atomistic methods is their high accuracy. For example, the behavior of individual atoms in a friction event at the nanoscale can be tracked in profound detail [6]. In addition, atomistic approaches naturally achieve formation of cracks and other types of defects by breaking and rearrangement of bonds without any additional effort. In spite of these advantages, atomistic methods are computationally very expensive hence limited to small systems. For example, a typical aluminum grain consists of approximately 10^{13} atoms, which is much larger than the quantity of atoms a typical computer can handle [7]. For another example, Gracie et al. states that a representative volume element should have a volume of 1,000 μm^3, which could be resolved by $8.6x10^{13}$ atoms assuming a lattice constant of 3.6 Å [8]. One of the heaviest simulations is conducted by Abraham et al. in 2002 that includes one billion atoms corresponding to a cube of only one micron side length [9]. The limitations of computational power is not only restricted to length scale but it is also true for the time scale. For instance, MD is available to only very high experimental strain rates, i.e., $10^3 – 10^6$ s^{-1} [10]. Length and time scales commonly studied in computational mechanics are categorized in Table 1 by Qian et al. [11].

Table 1. Categorization of length and time scales commonly studied in computational mechanics.

Method	**Length Scale (m)**	**Time Scale (s)**
Quantum mechanics	$< 10^{-8}$	$10^{-15} – 10^{-12}$
Molecular mechanics	$10^{-10} – 10^{-6}$	$10^{-12} – 10^{-9}$
Micromechanics	$10^{-6} – 10^{-4}$	$10^{-9} – 10^{-3}$
Continuum mechanics	$> 10^{-3}$	$> 10^{-3}$

In atomistic/continuum multiscale methods, the main motivation is to systematically reduce the total number of degrees of freedom while preserving accuracy. This way, simulations can be run in less time with less computational power. Furthermore, larger length and time scales, which are otherwise unfeasible, can be attained. Reduction in the total number of degrees of freedom is best guided with the following insight: only a small portion of atoms are actually undergoing localized deformation, such as dislocation and stacking fault, and the rest of the atoms are undergoing non-localized deformation, such as tension and compression [12]. The regions with the localized deformation are physically more relevant and therefore of greater interest. Figure 1 shows a snapshot from a nanoindentation example where only 8% of the atoms that undergoes localized deformation are made visible. This particular figure best explains the motivation behind the multiscale methods and how the multiscale model should be constructed. While the regions with localized deformation can be well investigated by highly accurate atomistic methods, much larger rest of the domain with non-localized deformation can be handled by efficient continuum methods. In that, the premise is that the deformation is free from abrupt changes in the continuum region. This way the entire domain is spatially decomposed into atomistic and continuum regions.

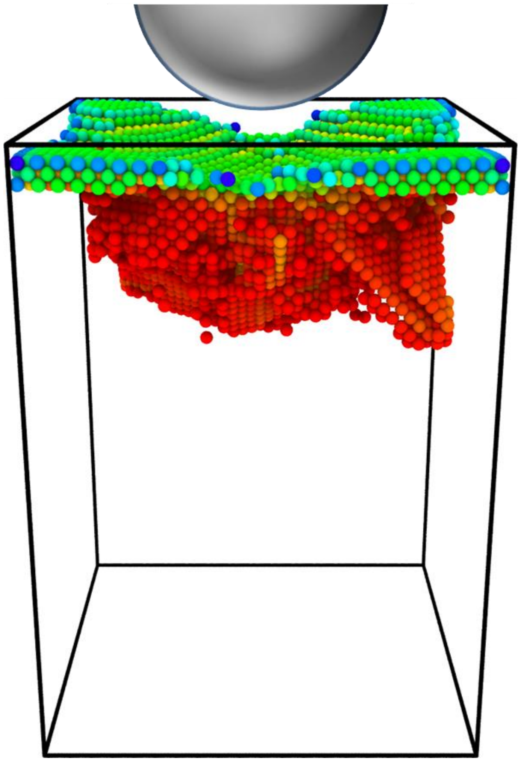

Figure 1. Interesting atoms take only 8% of the entire system in a nanoindentation example.

The multiscale approach is further motivated by the nature of matter itself. The structure of matter is dual in nature: continuous at a larger scale and discrete when viewed at an atomic scale [13]. In addition, the deformation and failure of many engineering materials are inherently multiscale such that the observed phenomena occur at many different length and time scales [14]. The macroscopic events are actually results of microscopic events happening at much smaller length scales. The disparity in length scales are as large as 10^{10} so that it is not feasible to include the effects of both scales without a multiscale model [15]. Another role of the continuum region is to keep the real boundaries far away and impose proper boundary conditions on the atomistic region; in this way, producing an improved boundary value problem replacing the necessity of running a full atomistic simulation [14].

In the atomistic scale, the atoms are treated as discrete particles with individual masses. The interactions between atoms are governed by empirical potentials that return interatomic forces and energies in reply to input positional vectors. Statics minimizes the potential energy of the entire system while dynamics solves Newton's second law of motion. In the context of multiscale modeling, continuum scale usually spans the larger rest of the entire scale, as categorized in Table 1. Except in the case of special handling of material defects, such as Extended Finite Element Method (XFEM) in Bridging Domain (BD) method and Dislocation Dynamics (DD) in Coupled Atomistic and Discrete Dislocation (CADD) methods, the homogeneous deformation of the continuum prevails under a linear elastic description, such as Cauchy-Born rule or averaging of atom clusters [16]. The continuum model is expected to reproduce the same material properties (i.e., elastic constants) with the atomistic model. In these continuum models, well-established Finite Element (FE) method techniques are often employed.

1.1 CLASSIFICATION OF MULTISCALE METHODS

Multiscale methods are classified in many directions:

(i) Formulation type: energy-based and force-based.

(ii) Coupling type: strong and weak.

(iii)Handshake region

(iv)Continuum model

(v) Hierarchical/Concurrent

Multiscale methods are classified into two groups with respect to the method of search of equilibrium structures by Miller et al. [16]: energy-based and force-based methods. Energy-based methods present an approximated energy of the system, and then they try to minimize the energy by exploiting the differential of the energy (which is equal to force) and zero out forces. On the other hand, main motivation of force-based multiscale methods is to realize that the ultimate purpose of energy minimization is to reach a configuration where forces are equal to zero. For this reason, they directly establish an approximate expression of forces instead of a unified energy functional for the entire system. Force-based methods have the following advantages: (i) energy-based formulation cannot eliminate ghost-forces [16] (the issue of ghost forces will be described later in detail) and (ii) formulating an energy functional for the entire system may not be feasible in some cases such as irreversible processes [15]. Force-based methods have the following disadvantages: (i) they have spurious solutions [12], (ii) they show slower convergence [12], (iii) they converge to unstable states [16], (iv) they are non-conservative [16], (v) they are numerically unstable [16], (vi) stiffness matrix derived from the force function is non-symmetric [17], and (vii) they cannot be as easily analyzed as energy-based methods [18]. Zhang et al. suggest that the force-based formulation, when derived from its energy-based counterpart, gives the same result with energy-based formulation at a substantially lower cost [19]. Energy-based formulations can be minimized by the conjugate gradient method applied on energy or the Newton method whereas force-based formulations without an energy functional can be minimized by the conjugate gradient method applied on force or the quasi-

Newton method [14]. It is noteworthy that some force-based formulations might be constructed by modifying the energy-functional such that the summation of forces is equal to zero.

A second classification is given due to the type of coupling by Miller et al. [16]. According to the given definitions, strong coupling requires continuum mesh size to reduce down to atomistic resolution and use of pad atoms in the continuum region for full coordination of atoms in the atomistic region [20]. Both requirements are shown in Figure 2 where the mesh is refined down to the atomic lattice spacing and pad atoms are located in continuum region. The positions of pad atoms are interpolated from positions of mesh nodes. The methods that do not conform to these requirements are regarded as to have weak coupling [21]. Coupling is argued to be much easier if the FE mesh is not required to be refined down to the atomic lattice spacing in order mesh nodes to match the atoms. This matching requirement is highly undesired in terms of adaptivity due to the need of re-meshing after each adaptation step. Luan et al. [21] and Nie et al. [22] implement a weak coupling over a region where the atomistic region provides boundary conditions for the continuum region and the continuum region does so for the atomistic region. The former is achieved by determining nodal positions from weighted average of atom positions within a radius. The latter is achieved by interpolation of atoms positions from nodal positions. These two layers of boundary conditions are kept far away from each other by unconstrained layers of atoms in order to prevent conditions of one to effect the other.

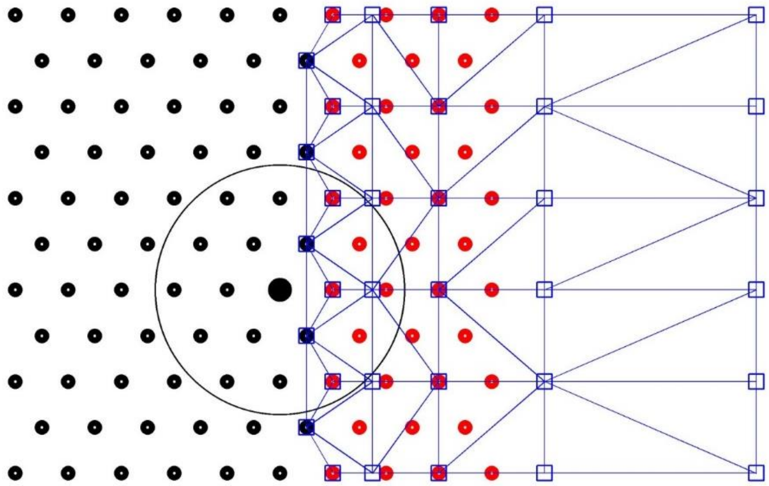

Figure 2. Atomistic/Continuum coupling. Black circles are atoms of atomistic region, blue squares and lines are nodes and mesh of continuum region, red circles are pad atoms, and big black open circle represents the cut-off of big black solid atom.

Third classification is given by Miller et al. [16] and Aubertin et al. [1] and it judges the multiscale methods whether they implement a handshake region or not. A handshake region is defined to be a region that is neither fully atomistic nor continuum but rather to serve for the integration of the two. In this region, a transition between the two main regions is accomplished with each method's own terms. The other option is an interfacial coupling in that the two formulations meet at a point in 1-D, edge in 2-D, or face in 3-D without any overlap. Handshake regions are employed by many multiscale methods [15, 23]. Due to this attention, Parks et al. analyzed the class of multiscale methods that provide solutions of two different methods back-and-forth to each other in an alternating Schwarz framework; and, conclude that the accuracy of the solution depends on both the size and position of the overlapping region [18]. Another analysis of blending functions is given by Badia et al. and will be discussed in later under

Atomistic-to-Continuum (AtC) method [24]. It is argued by Jebahi et al. that edge-to-edge coupling is only good for statics or dynamics with low temperature whereas a damping, hence a region to impose it, may be required for dynamics with finite temperature [2].

Fourth classification is due to continuum model and given by Miller et al. [16]. The continuum model specifies how the continuum region of the domain is modeled. A common continuum model is the Cauchy-Born rule, which displaces a group of atoms with respect to continuum deformation gradient and extracts relevant information. For example, the deformation gradient at FE quadrature points is applied to a group of virtual atoms from which the energy density, and in turn, stress is calculated. The validity of Cauchy-Born rule is limited to uniform deformation that can be justified by assuming small strains. This assumption, however, agrees with the expectation of most multiscale methods from the continuum approach. On the downside, Cauchy-Born rule restricts a change in lattice constant and poses difficulties for composite lattices such as graphene [25]. Another continuum model is the Virtual Atom Cluster (VAC) model developed by Qian et al. and will be detailed later under Bridging Scale Method (BSM) [11].

A fifth classification is given by many others [3, 26-28] and defines that (i) hierarchical methods (serial coupling or information passing) use the output of one or more scales in the simulation of the other scale(s) and that (ii) concurrent methods consider both scales at the same time in the same system. In concurrent methods, different length scales exist together and they continuously exchange information. Hierarchical methods are useful as demonstrated for viscosity of water by Abraham et al. [3]. In this example, viscosity of water obtained at a finer scale is informed to a coarser scale in two steps: from quantum mechanics to atomistic to continuum. Another useful application of hierarchical methods is to gather continuum constitutive inputs from separate MD simulations. As a matter of fact, passing of information obtained from atomistic methods to continuum methods has an older history. However, we will keep our focus on the trend of multiscale modeling emerging today: concurrent coupling [14]. It is also noteworthy that Jebahi et al. proposed a third classification as hybrid that combines features of hierarchical and concurrent approaches [2].

1.2 GHOST FORCES

Ghost forces are defined to be any residual forces that arise upon applying the multiscale method to a configuration that is already in equilibrium with forces equal to zero. Ghost forces arise due to local/nonlocal mismatch at the atomistic/continuum interface since the former is local and the latter is nonlocal. That is, in case the interactions are not restricted to the first nearest neighbors, the degrees of freedom in the atomistic region interact with a few neighbor shells in their surroundings, hence nonlocal, as shown in Figure 1. In contrast, the degrees of freedom in the continuum region only interact with the nearest neighbors, hence local. When they meet at an interface, degrees of freedom of the nonlocal region see the degrees of freedom of the local region but not the opposite.

At the interface, boundary conditions for the continuum region are easy to implement. For instance, nodes of a FE mesh can be set to match positions of some atoms. The reverse is, however, much harder to establish due to nonlocal nature of atoms. In case FE mesh is not fully resolved to atomistic length scale, the atoms does not see anything on the other side. A solution is to introduce a region of pad atoms within the continuum region from which atomistic interactions can be computed as shown in Figure 1. As noted earlier, pad atoms follow the rules of continuum formulation such as their positions are interpolated from the nodal displacements and FE shape functions. Unfortunately, pad atoms cannot completely eliminate the ghost forces either. Nevertheless, they provide an alternative insight to look at the ghost forces: the pad atoms exert forces on the atoms in the atomistic region and displace them. In contrast, the forces exerted by the atoms in the atomistic region on the pad atoms do not displace them. This mismatch introduces the error that is attributed to the ghost forces [14].

Ghost forces can be eliminated in many ways. First of all, force-based methods can easily eliminate ghost forces by construction since they formulate the force equations. But, the forces may not be conservative and force-based methods have many other disadvantages as discussed earlier. As a second solution, corrective forces can be added to the formulations to even the ghost forces [29]. These corrective forces are calculated for once at the beginning and then used until the end of the simulation. The initial calculation is usually practiced at a defect/stress free lattice configuration. In some cases, the corrective forces require a periodic update during the simulation [14]. In a third setting, the corrective forces are calculated from the difference between a cheaper model (such as QC with normal cluster size) and a more expensive model (such as QC with a larger cluster size) for sake of gaining efficiency [17]. However, it is argued that corrective forces are not conservative since they are not derived from a corrective energy [30]. Shimokawa et al. presents a forth way to eliminate ghost forces, which is to define a new type of atom (named quasi nonlocal atom) that is located at a buffer region between the continuum and atomistic domains and act both locally and nonlocally with respect to the opposite interacting atom [30].

Although there are many suggestions to avoid ghost forces in general, it is still a difficult problem to deal with in a systematic way. That is why; the solutions are usually ad-hoc. It is argued that there is not yet an energy-based method that completely avoids the ghost forces without any correction. In the investigation of the existence of ghost forces, infinite crystals by means of periodic boundary conditions are good examples. However, one should be careful that ghost forces do not arise in case of first nearest neighbor interactions since these interactions are local for both atomistic and continuum models. As the definition suggests, the ghost forces are best observed after applying the investigated multiscale method on a system that is already equilibrated by full atomistic.

1.3 WAVE REFLECTIONS

Concurrent multiscale methods suffer from wave reflections; that is, high frequency components of waves that emanate from atomistic domain cannot pass through the interface to the continuum domain, thus are trapped in the atomistic domain. In particular, waves with frequencies higher than what the continuum domain can represent are reflected back. An analytical proof to this reflection phenomenon is given by Jebahi et al. [2]. Cut-off frequency of the continuum domain is inversely proportional to the resolution of the continuum discretization, such as mesh size. Waves are especially crucial for heat problems since thermal phonons, which share the same properties with waves, are the main constituents of such kinds of problems. Reflection of the high frequency waves causes an unrealistic energy growth in the atomistic domain.

An ideal coupling is proposed to suppress all high frequency wave reflections at the coupling and transmit all low frequency waves to the continuum domain [31]. In the ideal case, the energy of high frequency waves should be completely dissipated whereas the energy of low frequency waves should not be dissipated at all. The energy of high frequency waves are negligibly small compared to the energy of low frequency waves so that dissipation of the former should conserve the energy to a large extent [31].

The same wave reflection phenomenon is also observed in FE method between two domains of different mesh sizes [32]. As a typical solution, the wave reflection is argued to be reduced by refinement of FE mesh down to the atomistics resolution [31]. However, it is also typically argued that the mesh refinement is costly and precludes the use of coarse timesteps in the continuum domain in multiple timestep schemes. This expensive meshing scheme is particularly a bottleneck in adaptive simulations due to re-meshing after each adaptation step.

Many methods design a handshake region to systematically eliminate the wave reflections at the atomistic/continuum interface [23, 33, 34]. For an example application, To et al. used Perfectly Matched Layers (PML) at the handshake region in order to damp the high frequency waves that could not be represented by the continuum description [35, 36]. In spite of these efforts, Curtin et al. employed a thermostat damping around the atomistic region in order to mitigate wave reflection [37]. Since the primary concern of wave reflections is the resultant temperature increase in the atomistic region, their work proves that a common MD thermostat can naturally regulate the temperature without a cumbersome handshake region.

It is important to note that waves are not reflected only because of different spatial resolutions but they are also reflected because of (i) different wave velocities of the two mediums (different material properties) and (ii) different timesteps used with different methods (in multiple timestep methods) [2]. The former is rather rare because most of the multiscale methods are developed for one type of material where they try to achieve same material properties for models that they try to couple. The latter is either employed or claimed to be potentially viable by some multiscale methods but their wave reflection aspects have not been discussed.

1.4 EXISTING MULTISCALE METHODS

In the following, a review of the existing multiscale methods will be presented. Due to the large body of the current literature that includes a high number and variety of multiscale methods, the review is restricted to a limited number of multiscale methods. Some of these methods are investigated in detail and some others in more general terms depending on their popularity. Considerably, presented methods are chosen such that their variety and volume span the majority of the literature and provide the reader with a thorough understanding of the existing efforts and ideas. The detailed methods are fully presented in every aspect except adaptivity and implementation, parts of which are saved for the following Chapters that are dedicated to these subjects. Special attention is given to quasicontinuum (QC) method due to its similarity to the MMM method.

1.4.1 Quasicontinuum (QC) Method

Quasicontinuum (QC) method is first introduced by Tadmor et al. in 1996, and it is the most popular of multiscale methods today [38]. So that, it is the mostly studied method with more than 200 journal articles published by numerous researchers from institutions all around the world. The method marries FE with molecular mechanics (MM) without any handshake region.

Fundamentally, QC method makes two basic assumptions: it imposes kinematic constraints and approximates energy/force.

1.4.1.1 First assumption: Kinematic constraints

In the kinematic constraint, only a small number of atoms are appointed as degrees of freedom of the system, called as representative atoms (rep atoms), and positions of rep atoms are explicitly accounted for. The number of rep atoms is much less than the number of all atoms in the system. The domain is discretized by a FE mesh where rep atoms are the nodes of the mesh. Positions of the rest of the atoms are interpolated from nodal values via FE shape functions. The only unknowns of the problem are nodal values of displacement, which is solved via a FE system. While atomistic potential is directly imposed in the atomistic region, it is incorporated via Cauchy-Born rule into the continuum. More specifically, the constitutive behavior is grounded upon the atomistic potential in the continuum region. The energy of the system is expected to be well approximated by this setting. However, the computational savings from this constraint is not enough because all of the atoms are still need to be visited in order to calculate energy of the system. Therefore, the following assumption is introduced.

1.4.1.2 Second assumption: Approximation of energy/force

The approximation is carried out for the energy in an energy-based formulation and force in a force-based formulation. For simplicity, we will follow the description for energy-based formulation hereafter. The energy approximation is conducted in two ways. The first one is using local formulation [38], which is also called the element-based summation rule [39]. The second way is by sampling energy of the system at some particular atoms. The latter is introduced as cluster summation rule [40], which is also known as atom-based summation [39]. Either way, the approximation formulation can be generalized to the multiplication of energies of some particular atoms (named sampling atoms) by some specific weights. These weights can be thought as the number of atoms represented by the sampling atom. The weight can be determined from one of the three functions: Voronoi characteristic function, patch characteristic function, and FE shape function. Knap et al., however, showed that the three methods are indifferent [29]. Moreover, the same study showed that the weights of nonlocal atoms should be unity. Energy sampling is further advanced to Lobatto quadrature, in which energy of an element is account by $E_{atom}xV_{element}/V_{atom}$ where E is energy and V is volume [40]. In its exact form, the weights are supposed to sum the FE shape functions exactly, but in practice they are usually lumped, which introduces some error from the construction [41]. In conclusion, the approximation scheme corresponds to sampling of energy from few atoms at each element; it thus provides substantial computational savings. Computation of energy of the system now scales by number of elements rather than number of atoms, hence several orders smaller.

1.4.1.3 Local & Nonlocal formulations

In the first paper of QC, Tadmor et al. distinguish two formulations: local and nonlocal [38]. In the regions where deformation varies slowly, the first assumption dictates that atoms are part of an infinite crystal undergoing uniform deformation thus the deformation gradient is constant. An element is then called local and the atoms in the element are displaced with respect to the constant deformation gradient. Even further, lists of positions of these atoms do not need to be saved in the memory. Instead, their positions can be generated as needed from the crystalline references. The energy, force, and stiffness are calculated as functions of deformation gradient from the continuum formulations. That is, a unit cell with periodic boundary conditions is displaced for an input deformation gradient (with respect to the Cauchy-Born rule), and its energy is then computed. The local formulation lacks the ability to account for some structural non-homogeneous structural features, e.g., stacking fault. Besides, local QC formulation suffers from non-symmetric stiffness matrix due to local-nonlocal force mismatch [17].

In contrast to the local formulation, an atom in a nonlocal element displace with respect to the deformation gradient at its position. In addition, nonlocal formulation dictates energy, force, and stiffness to be directly calculated from the atomistic potential. As a result, atom positions are implicitly accounted in the local formulation while they are explicitly accounted in the nonlocal formulation. In order to get the final values, energy, force, and stiffness terms from the local and nonlocal formulations are superposed to the global matrices. To decide whether an element will be treated by local or nonlocal formulation, a criterion is introduced. This criterion can be based on the second invariant of the Lagrangian stress tensor or it can be an empirical constant. It was found that magnitude of deformation is a bad criterion while variation in magnitude is a good one [7]. Furthermore, the elements that are close to the nonlocal elements are also appointed as nonlocal elements.

1.4.1.4 Fully nonlocal formulation

Knap et al. introduces the fully nonlocal force-based QC formulation and elaborates energy-based and force-based formulations that completely eliminates the local formulation, hence approximation by the Cauchy-Born rule [40]. This formulation is coupled with the cluster-based QC, which will be detailed in the next section. Miller et al. argue that the fully nonlocal QC formulation suffers from an overestimation of the energy at surfaces when a rep atom is located at the corner of a specimen [7]. The authors thus insist that use of local and nonlocal formulations together is the best choice. The fully-nonlocal scheme is further extended to energy-based by Eidel et al [17]. The author reported that the energy-based QC formulation eliminates rank-deficiency problem of stiffness matrix (that manifests itself as zero-energy modes) even for node-based summation rule [17].

1.4.1.5 Cluster-based QC

Rep atoms are located inside the elements and surrounded by some other atoms – altogether called a crystallite in the introductory paper [38]; but they are subsequently corresponded to the FE mesh nodes [42]. Later, Knap et al. introduces cluster-based and node-based (i.e., cluster-based with a cluster size of 1 atom) summation rules [40]. They first argue that node-based summation rule is indeed rank-deficient, and then proposed cluster-based summation rules. That is, in energy-based cluster-based QC, energies of the atoms over the cluster are summed and then multiplied by the cluster weight, which are the number of atoms the cluster represents. In other words, energy of each rep atom is taken to be the average of energies of atoms in its cluster. In force-based cluster-based QC, force is computed for each atom in the cluster and then mapped to nodes via FE shape functions. Atoms in the clusters are not degrees of freedom except the rep atom itself. In a sense, the cluster resembles the very initially introduced crystallites except that they are now centered at FE nodes. It is proven that cluster-based summation rule outperforms node-based summation rule and eliminates rank-deficiency problem [40]. The cluster-size can be useful in adjusting between the accuracy and efficiency. In this perspective, the effect of cluster size on accuracy is investigated by Knap et al. [40]. In cluster-based QC, the atoms in the full-resolution have a weight of 1, and the ones in the continuum have a weight bigger than 1. Cluster-based QC does not employ any constitutive law but it utilizes continuum principles to interpolate for the positions of some atoms.

Luskin et al. showed that node-based cluster summation rules are inconsistent and inaccurate and increasing the cluster size does not rule out the error [41]. They suggest three fixes to be investigated:

1. Quadrature-rule type sampling [19, 43-45].

2. For force-based method, summation over element interfaces rather than the elements themselves as in FE.

3. Non-uniform weighting of atoms in the clusters as in node-based quadrature rules of FE.

In support, Miller et al. also show that node-based cluster summation rules show larger error compared to other versions of QC [16].

1.4.1.6 Quadrature-rule type QC

Gunzburger and Zhang introduce quasicontinuum method with quadrature-rule type summation (QC-QR) and also derive energy-based and force-based formulations [19, 43]. More specifically, quadrature-rule type summation introduces a summation over atoms that are regarded as quadrature points multiplied by some weights. Selection of quadrature points rely on the established techniques of FE. The quadrature points are required not to be coplanar, i.e., they do not lie on a (d-1)-dimensional plane. The weight of a quadrature atom can be thought of as the number of atoms it represents. Weights are constrained such that the summation is exact for linear polynomials. In fine-scale regions, the summations are suggested to be explicitly done whereas in coarse-scale regions Lobatto quadrature is suggested to be employed. All quadrature settings are sorted at the beginning of the simulation for one time for the rest of the simulation.

In their work, Gunzburger et al. argue that the QC method with only kinematic constraints (hence no sampling) has dN_r (where d is dimension and N_r is number of rep atoms) system of equations and energy calculation has $O(NN_b)$ complexity where N is number of atoms and N_b is number of atoms in a ball defined by the cut-off distance [43]. However, QC-QR reduces the energy calculation complexity to $O(N_r)$. The authors conduct error analysis with a 1-D atom chain for a quadratic function and conclude that QC-QR is superior to QC-CS (quasicontinuum method with cluster-based summation rule) in terms of both accuracy and efficiency. They also confirm the same result with a 1-D nanoindentation example. Another comparison is conducted between QC with no sampling and QC-QR. It is found that QC with no sampling has better accuracy and QC-QR has better efficiency. The same conclusions for the QC with no sampling and QC-QR are drawn by Zhang et al. for both short-range and long-range interactions [19].

As a similar method that performs energy sampling within elements, Lin sampled the energy in the middle of the elements with degrees of freedom still being the nodes of elements in a 2-D triangular mesh setting [44]. In that, the author assumed that the energy of any atom in an element is approximately equal to the energy of the atom in the middle of the element.

As another similar method that performs energy sampling within elements, Beex et al. sampled energy by one atom inside the elements, called the central summation [45]. Moreover, discrete sampling atoms sample their own energy in order to be utilized in small elements. It is noted that using discrete sampling atoms along the edges of the elements, to make sure that rest of the atoms in the element has all neighbors in the same element, results with exact summation of the energy of the system. The incenter of the triangular elements serves as the best position for the location of sampling atoms since it is the farthest point to the edges of the triangle. Energies of the atoms that lie on the boundaries of an element are proportionally (in some cases equally) split to the adjacent elements. The authors compare central summation to cluster summation and find that central summation results are much better. They also present a numerical example similar to the benchmark problem tested by Miller et al. for fourteen methods [16].

1.4.1.7 Ghost forces and solutions

QC method, like most of the other multiscale methods, inevitably suffers from ghost forces, which arise at the local/nonlocal interface. Ghost forces are not conservative and they lead to asymmetry, for instance, in stiffness matrix. Also, they are confined to the local/nonlocal interface. E et al. show that both element-based and cluster-based sampling rules introduce ghost (spurious) forces [39]. With respect to their own set of definitions, Eidel et al. classify spurious forces into two groups as ghost forces and residual forces [17]. Although these definitions are consistent in their own right, we use the term “ghost force” in place of the term “spurious forces” in this work in order to be consistent with the rest of the literature. Following the set of definitions of Eidel et al., they state that ghost forces are a result of modeling error because different types of atoms inconsistently interact with each other. Residual forces are different from ghost forces in three ways: they are (i) conservative, (ii) symmetric, and (iii) distributed to the entire domain. Fully nonlocal QC has some residual forces originating from the approximations of the QC method, i.e., numerical quadrature. Residual forces do not appear with uniform meshes or they can be reduced by increasing the number of sampling (quadrature) points. Another way to eliminate residual forces is to apply a correction force similar to what is suggested for ghost forces.

1.4.1.8 Inconsistencies

In general, the term inconsistency may imply ghost forces or it may imply non-conservative forces/energies. E et al. discuss two inconsistencies of the QC method [39]. The first one is at the local-nonlocal interface since local formulation only considers the nearest neighbor atoms whereas nonlocal formulation extends beyond the nearest neighbors. The second inconsistency is the atom-based versus element-based samplings employed in the local region. Atom-based formulation is favored for small elements while element-based formulation is favored for large elements. The authors also present accuracy analysis at the local-nonlocal interface and element-based summation rule.

Another inconsistency of the cluster-based QC method is shown by Yang et al. [46]. That is, the weighting factors are calculated by assuming that the energy is piecewise linear. Indeed, this assumption conflicts with the earlier assumption of linear shape functions since linear shape functions readily implies that the energy is constant in an element. This may be a reason for the fail of cluster summation rule as shown by Luskin et al. [41].

1.4.1.9 Implementation of local – nonlocal QC

At the implementation level of QC, the method (i) starts with selecting rep atoms; (ii) deforms atoms in the crystallites (or clusters) with respect to either local or nonlocal formulation using deformation gradient; (iii) computes energy and derivatives from the underlying atomistic potential and feeds these back to the FE system; (iv) solves the FE system; (iv) adapts to the new state by mesh refinement and incorporates nonlocal elements in over strained regions; and (v) goes back to step (ii). The authors use Delaunay triangulation with linear elastic elements, which confirms the earlier assumption of constant deformation gradient for the local formulation. Therefore, interpolations are performed in a linear piecewise manner utilizing FE shape functions, which also obey partition of unity. They utilize conjugate gradient and Newton-Raphson solvers. Further implementation aspects and improvements are detailed in Knap et al. [40].

1.4.1.10 Extensions and Modifications

Extension of QC to finite temperature dynamics is introduced by Shenoy [47]. However, high frequency waves are not able to pass from the atomistic to the continuum region. This could be solved by a specialized interface treatment in expense of implementation complexity and computational cost. Further, for a discussion on finite temperature QC, check out section 6.1 of Miller et al. [7]. Also, QC is extended to polycrystals [29], three-dimensions [40], complex lattices [48], and curved lattice structures such as CNT [49]. Some numerical examples the QC method solved include static simulations of dislocations [38], quasistatic simulations of fracture [42] and deformation processes including interfaces [29] in FCC and BCC [10] crystals.

QC method is modified in many ways. The new versions usually differ by the first or second assumption. Kochmann et al. present energy-based nonlocal meshless QC based on local maximum-entropy interpolation [50]. The local maximum-entropy interpolation scheme establishes a balance between the entropy and shape function support such that one end is local support (minimum entropy) and the other end is global support (maximum entropy). The balance is parameterized and can be adjusted accordingly. As a result, the meshfree setting yields better control and flexibility on adaptation and adjustment of accuracy. Shan et al. modified local QC method in two ways: they used (i) the Cauchy-Green tensor instead of the deformation gradient, and (ii) representative lattice instead of rep atoms [27]. They named their approach as “concurrent lattice homogenization”. Advantages and disadvantages of the approach are discussed as well.

1.4.1.11 Critique

It is difficult to distinguish every QC version and discuss their drawbacks. Nonetheless, a generic disadvantage of QC is due to its adaptation mechanism. The adaptation mechanism suggests introducing atomistic resolution wherever required in the domain. In case of a deformation that leads to many dislocations spread out around the domain, the adaptivity scheme requires most of the domain to be resolved by atomistics. In turn, this results with a nearly full atomistic setup and thus cost. In this sense, the methods that can represent deformation mechanisms in continuum are superior. These methods may show better efficiency than QC in general as well. In this respect, QC is more suitable for problems with localized defects and problems with minimal spread of defects.

1.4.2 Bridging Domain (BD) Method

Bridging Domain (BD) method couples two decomposed domains of different scales via another domain, called the bridging domain, hence the name [23, 32]. The scales vary from quantum to atomistic to continuum. In two dimensions, the coupling may be accomplished edge-to-edge or over an overlapping area, called the handshake region. In other words, two domains of different scales meet at an edge or region for coupling. The coupling imposes a displacement constraint in order to achieve compatibility between the two scales. In fact, the constraint dictates that the displacement fields of both scales conform to each other. The energy (or Hamiltonian) of the handshake region is a weighted sum of energies of fine and coarse scales in order to avoid double counting. Although a linear weighting is usually the case, a nonlinear weighting is also reported [32]. The scaling parameter that constitutes weighting is defined in material coordinates thus it is constant during the simulation [31]. The total energy of the system is a sum of energies of fine, coarse, and handshake regions. The constraint is imposed by using the augmented Lagrange multiplier method that includes the total energy of the system, the constraint, and a penalty term, which augments the constraint. Governing equations of the system can be derived by considering the stationary point of the constructed Lagrangian. Zhang et al. showed that these governing equations can be further simplified if the continuum domain is taken to be residing in the linear elastic regime in order to gain efficiency [51]. In that, BD method takes the Cauchy-Born rule as the continuum constitutive model in order continuum domain to be consistent with the atomistic domain [25]. The method also utilizes pad atoms (and even pad elements for the same reason) that are placed around the coupling region to prevent the force imbalance on atoms of the atomistic domain due to missing coordination [8, 25, 51].

The method is first introduced in statics by Belytschko et al. [23] and then extended to dynamics with multiple timesteps by Xiao et al. [32]. Zhang et al. presents some extensions and analysis of the method [51] and Xu et al. investigates conservation properties [31]. Gracie et al. and Moseley et al. add adaptivity and XFEM description to the method [8, 25]. Marenic et al. compares BD/Arlequin method to QC [52]. In some examples, the method is employed in solving numerical examples of bending [23] and fracture [23, 25, 51] of graphite sheet, bending [23] and fracture [23, 51] of carbon nanotube, wave propagation [31, 32], crack propagation [32], heat transfer [32], and dislocation nucleation from void and crack tip [8].

In the dynamics version, Xiao et al. utilizes a velocity Verlet time integration method along with a predictor-corrector scheme to determine the Lagrange multipliers [32]. As a separate note, Lagrange multipliers are approximated by interpolation via shape functions thus the coupling is not exact. The dynamics formulation damps out the high frequency waves in the handshake region, which is observed by the dissipation of molecular energy. The success of the damping of high frequency waves is proportional to size of the handshake region. In this regard, an edge-to-edge coupling is shown to be performing poorly.

Zhang et al. presented a method based on Moving Least Squares (MLS) to approximate the atomistic strain values properly [51]. The method is shown to be reproducing linear and quadratic displacement fields exactly. In the same work, calculation of stress is also achieved by the Cauchy-Born rule. In effect, stress is derived from the continuum energy, which is set equivalent to the atomistic energy. The authors also run a patch test that results with good accuracy but also with an indication of room for improvement. Results of numerical examples show that global error is indifferent for varying sizes of atomistic domain or mesh; however, local error decreases with increasing size of atomistic domain and increases with increasing size of mesh.

Xu et al. analytically prove that BD method conserves linear momentum, angular momentum, and energy by showing that the time derivatives of these properties are equal to zero [31]. They also confirmed the conservation properties by a numerical investigation. The authors compare coupling features of consistent and diagonalized constraint matrices and find that the diagonalization is cheaper, more effective in damping wave reflections, and dissipates energy of high frequency waves. The latter seem to violate the energy conservation property but the violation is negligible due to the small energy of high frequency waves.

Gracie et al. employed the method with XFEM to represent discontinuities such as dislocations in the continuum domain [8]. The efficiency of XFEM formulation is shown to be superior to that of classic FE formulation. The authors also presented an adaptivity scheme based on the energy of atoms to periodically refine/coarsen elements of FE mesh. Refinement is carried out for elements that contain high energy atoms. Coarsening is conducted for elements without any high energy atoms and the error calculated by a least squares fit is below a particular threshold value.

Moseley et al. further extends the BD method to handle a wider range of discontinuities including cracks, dislocations, and other defects [25]. It is noted in this work that the coupling of BD method is more difficult for complicated lattices such as graphene. The authors also advance the adaptivity of the BD method. In addition, they point out some downsides of the XFEM coarsening such as energy loss at surface representation and violation of energy conservation.

In the critique of the method, a handshake region is argued to be disadvantageous for the reasons that (i) Lagrange multipliers deteriorate the positive-definiteness of the system, which precludes use of many iterative methods [51], (ii) the implementation is rather complicated compared to other methods without a handshake region, and (iii) it loads an additional computational burden. In addition, the method is limited to two-dimensions and straight dislocations.

1.4.3 Bridging Scale Method (BSM)

In the Bridging Scale Method (BSM), the key idea is to divide displacement into orthogonal coarse and fine components. In that, the coarse scale components exist in the entire domain while the fine scale components exist only at the fine scale region. In accordance, the FE mesh (or meshfree [11]) discretization, hence the shape functions, is imposed over the entire domain. Then, the coarse scale components are only the FE displacements interpolated by the FE shape functions. The fine scale components are only required in the atomistic region where the total displacement is calculated from the MD simulation. Utilizing the shape functions, a set of displacements are fit (projected) to the total displacements. The fitness measure is the squared difference weighted by atomic masses. The fine scale components are then the total displacement minus the set of fit displacements, since the latter is obtained by some sort of a coarsening technique. Next, equations of motion are derived from the displacement that is disintegrated into coarse and fine components. In this context, there are two sets of equations of motion: coarse scale and fine scale. The coarse scale equations are solved everywhere in the domain while fine scale equations are only solved in the enriched region [33]. It should be noted that the coarse scale components could also be extracted from the MD displacements since the latter contains both fine and coarse scale components. However, the atomistic region is limited to a part of the domain, hence not general [33]. Another note is that the convergence of fine scale system is enhanced by utilizing the coarse scale solution as an initial estimation [11].

Although the forces of fine scale system are calculated as in the standard MD simulations, the forces of coarse scale system are calculated differently for the atomistic and continuum regions. In the atomistic region, the displacement that is expressed as a sum of coarse and fine components is plugged into the Newton's second law of motion. In the continuum region, the forces are calculated by a continuum technique such as the Cauchy-Born rule [33] or specialized techniques such as the Virtual Atom Cluster (VAC) technique [11]. Besides, a multiple timestep scheme is employed such that the continuum and atomistic routines cycle one after each other and dynamically exchange information [33].

In the coupling of coarse and fine scale systems, the fine scale is included in force calculation of the coarse scale. In reverse, pad atoms represent the effect of coarse scale on the fine scale. However, the pad atoms can only impose the coarse components of displacements by providing full coordination to the atoms in the atomistic region [11]. This will impose a constraint that does not originally exist. To overcome this problem, BSM introduces the impedance and random force as specialized boundary conditions on the atomistic region [33]. In that, an impendence force (that is derived from a time history kernel) and a random force term that represents the effects of fine scale displacement components of the removed degrees of freedom in the continuum region are employed. Both terms are attached on the right hand side, hence as an external force, of the fine scale equations. The boundary atoms are solved by this updated fine scale equations whereas the interior of atomistic region is still solved by the original fine scale equations. The impedance and random force terms imposes the linearized effect of fine scale components of continuum region on the atomistic region implicitly. The impedance force is calculated from the known quantities: coarse and fine scale components of displacements in the atomistic region. The time history kernel that is required for the calculation of impedance force is calculated analytically for simple cases and numerically for complicated cases [33]. Park et al. improves the calculation of time history kernel to a (i) numerical automated procedure, (ii) more compact size due to lattice symmetry, and (iii) less storage due to truncation in time history [53]. The impedance force term eliminates the wave reflection by means of dissipating fine scale energy (therefore the system is dissipative and not conservative [53]). Furthermore, it lets the low frequency wave to pass from the continuum region to the atomistic region. The random force term also imposes the temperature effects of continuum region on atomistic region. In case temperature effects to be ignored, this term is removed from the equation. The random force term conducts energy exchange between continuum and atomistic regions due to temperature difference [33]. The reader is referred to Figure 1 of Liu et al. for a representation of how impedance force takes place in governing equations [54].

The method is extended to finite temperature [55], quantum/continuum coupling [56], different resolutions of continuum/continuum coupling [57], and 3-D [58]. Also, Farrell et al.

focuses on the interface between continuum and atomistic scales and introduces implementation details and efficient algorithms to calculate the time history kernel, lattice stiffness matrices, and random force [59, 60]. In some examples, the method is applied in solving dynamic 1-D wave example with harmonic and anharmonic potentials [33], quasi-static twisting and bending of carbon nanotubes [11], 2-D wave and crack propagation [53], 3-D crack propagation [58], and 2-D intersonic crack propagation [59].

1.4.4 Other Methods

The literature includes many other multiscale methods. Some of these methods will be summarized in the following.

1.4.4.1 Coupled Atomistic and Discrete Dislocation (CADD)

Coupled Atomistic and Discrete Dislocation (CADD) plasticity couples atomistics with linear elastic continuum [20]. The specific feature of CADD is to incorporate dislocations in continuum as well. This way, much larger length scales can be attained. These dislocations are solved by Discrete Dislocation (DD) method in the continuum and are coupled to each other. Enabling the passing of dislocations between atomistic and continuum regions, dislocations in the atomistic and continuum regions are coupled too. The developers of the method present specialized algorithms for the detection and passing of dislocations between the atomistic and continuum regions. The atomistic region is treated by the standard molecular mechanics. On the interface, pad atoms are utilized to make the atoms of the atomistic region act like bulk atoms in price of having a stiffer region due to double counting of the energies of pad atoms. The method is later extended to finite-temperature dynamics using a Langevin thermostat [37]. In that, the waves emitted from the atomistic region are absorbed in a “stadium” that is damped by tweaking the damping part of the Langevin thermostat. The thermostat is only employed in the stadium region; yet, it is shown to be able to produce the desired temperature and thermal fluctuations in the atomistic region by exchanging kinetic energy with the interior atoms. They admit that the presented method is worse than other techniques developed for zero-temperature and linear material behavior, but they claim that it is better for finite temperature and non-linear behavior. Overall, the method is currently limited to 2-D and straight dislocations.

1.4.4.2 Concurrent Atomistic-Continuum (CAC)

Concurrent Atomistic-Continuum (CAC) method combines full atomistics with a FE framework [61]. In the method, crystalline materials are modeled as continuous collection of lattice cells with a group of discrete atoms inside. Governing equations are derived from balance laws that are reformulated based on the multiscale model. Expectedly, full atomistic resolution is utilized in regions of interest and a coarsening approach is utilized elsewhere. The most prominent feature of the method is its ability to represent dislocations slip planes in the coarse regions. In that, the authors realize that, owing to the formulation, finite elements do not need to be connected. As a result, Burgers vector can be represented along the interelement boundaries with the help of shape functions. In other words, the coarse setup lets finite elements to slip along each other in order to represent dislocations. Moreover, dislocations are able to pass from the atomistic to the coarse region and vice versa. In the former, the dislocations are able to pass even when they hit across the element boundaries. However, the deformation representations in the coarse region are limited to only dislocations and it is shown that stacking faults and twinning cannot be represented. Later, Xiong et al. introduced adaptivity to CAC by making it possible to split the finite elements into two in order to let otherwise suppressed deformation mechanisms such as dislocation migrations to develop [62]. This way, the requirement to employ atomistic resolution to capture certain deformation mechanisms is eliminated. Furthermore, the restriction on deformation mechanisms in the coarse region is relaxed enabling stacking faults, albeit leaving twinning out of limits.

1.4.4.3 Atomistic-to-Continuum (AtC)

Atomistic-to-Continuum (AtC) is a force-based concurrent multiscale method [15]. In the method, the authors model particular regions of the domain with atomistic and continuum methods. The equilibrium is achieved by blending stresses of the two methods on an overlapping subregion. The atomistic stress is defined by means of the Virial stress theorem. The blending of stresses is actually equivalent to blending forces. The compatibility, on the other hand, is ensured on an average sense by the displacements of the two methods. Fish et al. compares three different blending functions: piecewise constant, piecewise linear, and piecewise cubic for each finite element in the overlapping domain; and, find that they are indifferent [15]. Badia et al. analyze blending methods, constraints, and imposition of these constraints [24]. In their investigation, they analyze four different blending methods and propose a consistent one in conclusion. In addition, they discuss three different constraints: (i) atomistic displacement is slaved to continuum displacement, (ii) continuum displacement is slaved to atomistic displacement, and (iii) average atomistic and continuum displacements are related. These constraints are further discussed to be imposed by Lagrange multipliers or restricting mathematical spaces in order to satisfy constraints. Two ways of imposing constraints produce the same results but differ in terms of implementation. They also necessitate that the blending should provide continuity and anticipated material properties in the effective region. Consistency [24] and patch tests [15, 24] of the method are also given.

1.4.4.4 Adaptive Multiscale Method (AMM)

Adaptive Multiscale Method (AMM) is developed by Budarapu et al. [63]. The method is distinguished from the Extended Bridging Domain Method (XBDM) in two aspects: using (i) continuum description everywhere instead of only continuum region and (ii) bridging two domains by only pad atoms instead of a bridging domain. Also, the authors employ Virtual Atom Cluster (VAC) approach instead of the Cauchy-Born rule. They partition the displacements into coarse and fine components and have them exist everywhere and atomistic regions, respectively. The authors employ phantom node method in describing elements with discontinuities instead of the Extended Finite Element Method (XFEM). In a following and similar method, called the Meshfree Adaptive Multiscale Method (MAMMF), Yang et al. resolve the continuum region by a meshless approach and they treat the discontinuities by an enriched Differential Reproducing Kernel Particle (DRKP) approximation [64].

1.4.4.5 Macroscopic Atomistic Ab initio Dynamics (MAAD)

In their method, known as the Macroscopic Atomistic Ab initio Dynamics (MAAD), Abraham et al. seamlessly unifies the descriptions of continuum, atomistic, and quantum mechanics [3]. Specifically, they couple the FE, MD, and tight-binding methods, respectively. The method is formulated for dynamics, run at low temperatures, and make use of a multiple timestep algorithm. In a crack propagation example, for instance, the bond rupture at the crack tip is captured by tight-binding, the nonlinear atom behavior along the crack is captured by MD, and the rest is captured by FE. The authors introduce special treatments at the FE/MD an MD/tight-binding interfaces. In the coupling of FE to MD, mesh size is refined down to the atomistic resolution in order to eliminate the wave reflection.

1.4.4.6 Smoothed Molecular Dynamics (SMD)

Smoothed Molecular Dynamics (SMD) is developed by Liu et al. [65]. The method aims to increase the efficiency in coarsened regions by using a larger timestep. A larger critical timestep is achieved by solving the momentum equations on a regular background grid. As a result, the behaviors of atoms are smoothed by eliminating the high frequency waves. The interpolations are performed by FE shape functions and the background grid is kept regular by restricting its deformation. The SMD region naturally converts to FA when the grid size is refined down to atomic scale. A multi timestep scheme is implemented in order to facilitate different domains with different timesteps. Adaptive features of the method are later developed and presented as Adaptive Smoothed Molecular Dynamics (ASMD) [66].

1.4.4.7 Coarse-grained

Coarse-grained multiscale methods with adaptive resolutions are presented [67, 68]. In this method, selected groups of atoms (e.g., molecules) are treated as single particles in the coarsened region while their substructures (i.e., atoms) are explicitly account for in the atomistic region. The atomistic region is identical to an MD simulation as it is governed by a standard potential. The coarsened region, however, is governed by an effective potential that is derived from the potential used in the atomistic region. This way, the method is introduced as MD with different resolutions. The adaptivity is facilitated by converting molecules as they travel between atomistic and coarsened regions. In order to obtain a smooth transition, the conversion is gradually performed by means of hybrid molecules interacting with both regions in a weighted manner [67]. In addition, conservation properties of these methods are investigated and novel conservative algorithms are introduced [68].

1.4.4.8 Atom Collocation Method (ACM)

Yang et al. proposes force-based Atom Collocation Method (ACM), which is truly meshfree [69]. Implementing a compatible atomistic/continuum interface, the method is proven to be free of ghost forces and provides enhanced adaptation capability by easily turning collocation atoms on and off.

1.4.4.9 Surface Cauchy Born (SCB)

Park et al. introduces the Surface Cauchy Born (SCB), which is an energy-based continuum method that efficiently captures the prominent surface effects at the nanoscale by utilizing a carefully modified version of the well-known Cauchy-Born constitutive model around the surfaces [70].

1.4.4.10 Review

For further information about multiscale methods, interested readers can see excellent review articles of the literature [2, 5, 12-14, 16, 71-75].

1.5 NUMERICAL EXAMPLES

At the end of each Chapter, numerical examples are introduced to demonstrate the validity and accuracy of MMM, which is achieved by comparing MMM results to that of full atomistics where the latter served as a datum. When the trajectories are compared, it should be noted that the energy functional is highly non-convex and has many metastable configurations at local minima. Due to this characteristic property, many deformation paths are possible and hence deformation behavior of the crystals lacks uniqueness [40, 61]. Another matter is that results of dynamic simulations oscillate about some mean value due to inherent vibration of atoms. For the sake of clarity, forces in the presented results are averaged from a large group of atoms over an interval of time. The iterations are integrated by the Velocity-Verlet scheme. All simulations are run with our MMM software, which is presented in Chapter 4.0 . Results are post processed by C++ and MATLAB programs. Visualizations are rendered by MATLAB and by the fast and user-friendly visualization software OVITO [76].

1.6 RESEARCH OBJECTIVES

The field of multiscale modeling and simulation is not yet fully explored. In particular, concurrent atomistic/continuum coupling multiscale methods need to be studied further in order to contribute to the understanding and development of the field. That for, in this dissertation, the following objects are to be achieved:

- Extension of the MMM method from its current statics stand to zero temperature dynamics. Derivation of the equations of motion from the MMM energy functional, incorporation of the temperature effects, and mitigation of the wave reflections.
- Incorporation of adaptivity into the MMM method. Development of robust, efficient, accurate, and conservative refinement and coarsening procedures associated with a sound adaptivity scheme and effective adaptivity criteria.
- Implementation of an efficient, modular, readable, robust, flexible, and contemporary MMM software. Demonstration and discussion of the efficiency and scalability of the software.

All objectives are to be accompanied with comprehensive numerical examples to demonstrate the effectiveness and application of the described concepts.

1.7 OUTLINE

In the following Chapter, formulation and rationale of the MMM method with an emphasis on dynamics will be presented. In Chapter 3, adaptivity scheme of MMM method will be detailed. In Chapter 4, implementation aspects of the MMM method and the MMM software will be presented along with a discussion on efficiency in detail. All Chapters are accompanied with numerical examples for further demonstration of the described concepts.

2.0 DYNAMICS

The Multiresolution Molecular Mechanics (MMM) method is a concurrent energy-based atomistic/continuum coupling multiscale method [46]. The method does not require implementation of a cumbersome treatment over a handshake region. In a nutshell, the coarsening is achieved by means of a FE mesh that is utilized to impose (i) kinematic constraints and (ii) energy approximation. The first one is imposed by using shape functions to interpolate positions of groups of atoms from the nodal positions of a mesh. The second one is imposed by sampling the energies of groups of atoms by only a few selected atoms. The method introduces an atomistic description everywhere in the domain. The most prominent advantage of this is having a truly seamless connection between atomistic and continuum regions. It does not differentiate between atomistic and continuum regions since the continuum is indeed an atomistic region coarsened by continuum concepts. Also, atomistic description preserves the essential advantages of full atomistic simulations. Other advantages include (i) eliminating the requirement of the mesh to be refined to the atomistic resolution, (ii) eliminating the requirement to use pad atoms, (iii) accommodating the continuum to be naturally modeled by atoms without a special procedure, and (iv) mitigating ghost forces to a great extent. The price paid in using the atomistic description is the computational cost of storing and computing for the crowded number of atoms. The number of ghost atoms can be potentially reduced by excluding latent atoms from storage and calculations (the calculations are indeed eliminated later in the Implementation

Chapter). MMM is introduced for statics [46] and dynamics [77] followed by convergence and error structure analysis [78] and introduction of a unified and consistent framework for general FE shape functions [79].

2.1 RATIONALE

MMM method extends molecular mechanics to larger scales by employing continuum methods. More specifically, molecular mechanics is employed in regions of interest and a continuum approach by means of finite elements is utilized elsewhere. Owing to the atomistic description, the atoms are regarded as degrees of freedom in the atomistic region whereas they are represented by some special atoms in the continuum region. These special atoms and atoms in the atomistic region are assigned a type and called “rep atoms”. More specifically, the special rep atoms in the continuum region are called interpolating rep atoms (red atoms in Figure 3). Setting the interpolating rep atoms as nodes of a mesh, the continuum region is discretized by finite elements. The mesh, coupled with FE shape functions, represents the collective behavior of atoms in the continuum region. The atoms that are represented by interpolating rep atoms are altogether called ghost atoms and further assigned three types as will be detailed later.

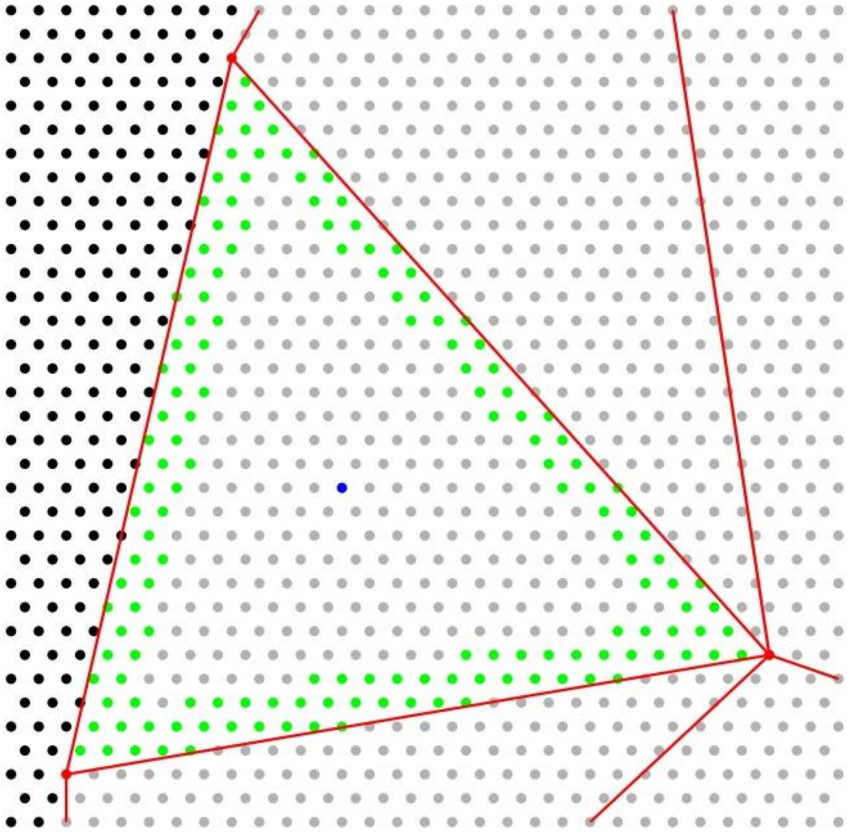

Figure 3. Types of atoms: interpolating rep atom (red), non-interpolating rep atom (black), primary sampling atom (blue), secondary sampling atom (green), non-sampling atom (gray).

A key point in the MMM framework lies in an assumption it makes about the continuum region. MMM assumes that deformation of the continuum region is linear; therefore, it can be represented by linear elements. As noted earlier, MMM is originally introduced using linear elements [46] and then extended to higher order elements [79]; but we are considering linear elements in the current work. In correspondence to linear elements, the strain (or the deformation gradient) in each element is constant. The assumption is further supported by the fact that the difference between true and homogeneous deformation is minimized within the cut-off radius of the potential. As a result of this assumption, the energy distribution in an element is constant for the interior atoms. That is, the atoms that do not have neighbors in other elements have the same energy. This useful property still holds when the element is deformed. The uniformity of energy distribution motivates energy sampling, which is a shortcut to compute the energies of atoms in an element. Energy of a preselected atom is sampled and the value of the energy is assigned to other atoms in the element. The atom that is used to sample the energy is called the “primary sampling atom” (blue atom in Figure 3) and the atoms whose energies are sampled by the primary sampling atom are called the “non-sampling atoms” (gray atoms in Figure 3). The energy sampling scheme saves the computational cost of calculating energies of the majority of atoms in the continuum region.

In accordance with the earlier assumption, MMM employs kinematic constraints in the continuum region. That is, the positions of atoms that are not appointed as degrees of freedom are interpolated from the positions of interpolating rep atoms located at the nodes of the mesh. In order to be consistent with the linear elements, MMM employs linear shape functions in the continuum region [46]. The shape functions are utilized not only for interpolating the positions of atoms but also for interpolation of the masses, velocities, forces, and other state variables since the rep atoms are meant to fully represent other atoms. The shape functions utilized in MMM are built and operated in the same way as in FE. For instance, an interpolating rep atom is locally supported by the corresponding shape functions of the elements where it is a node by construction.

Of importance is that the uniformity of energy distribution in an element is only valid in the interior of the element. This is because the energies of the atoms along the edges of an element depend on the atoms in neighbor atoms located in adjacent elements. This asymmetry implies that the edge atoms have different environment and hence different energy than the interior atoms. In order to solve this issue, the energies of some of the atoms should be accounted for individually instead of employing the earlier one-for-all sampling relationship for the primary sampling atoms. These atoms are called the “secondary sampling atoms” (green atoms in Figure 3). They are utilized near the edges for the aforementioned reasons and around the rep atoms for better accuracy. Consequently, MMM method has five types of atoms as listed in Table 2.

Table 2. Descriptions of five types of atoms of MMM method.

General name	**Specific name**	**Degree of freedom**	**Energy**	**Description**
Rep atom	Interpolating rep atom	Representative	Individual	Nodes of the mesh, interpolates for the positions of ghost atoms
	Non-interpolating rep atom	Individual	Individual	Same as atoms of full atomistic
Ghost atom	Primary sampling atom	Passive	Representative	Samples for the energy of non-sampling atoms
	Secondary sampling atom	Passive	Individual	Samples its own energy
	Non-sampling atom	Passive	Passive	Contributes to atomistic description

Four out of five types of atoms are described above. The fifth type of atom is called the "non-interpolating rep atom" (black atoms in Figure 3), which is identical to an atom in full atomistic. These atoms are employed in atomistic regions of the domain in order to obtain high accuracy. In total, MMM has five types of atoms each for a specific purpose as listed in the last column of Table 2. Nonetheless, atoms can be assigned to any of these types, thus providing the method with a high level of flexibility. For instance, primary sampling atoms and secondary sampling atoms can be chosen in any number and location in an element. Suggestions on this matter have been recently proposed [79]. This flexibility provides the user with an opportunity to tune the atom type selection for the optimum balance between accuracy and efficiency.

2.2 FORMULATION

In the following, the concepts described above will be cast into the mathematical formulation of the MMM method on dynamics, namely MMD. For an isolated full atomistic model, let E_α represent the site internal potential energy and T_α the kinetic energy of each atom. Also, let E^{int} represent the total internal potential energy, E^{ext} the total external potential energy, Π the total potential energy, and T the total kinetic energy. Then, we have

$$\Pi = E^{int} + E^{ext} = \sum_{\alpha=1}^{N_A} E_\alpha(\mathbf{r}^{N_A}) - \sum_{\alpha=1}^{N_A} \mathbf{f}_\alpha^{ext}\mathbf{u}_\alpha \tag{1}$$

$$T_\alpha = \frac{1}{2m_\alpha}\mathbf{p}_\alpha(\dot{\mathbf{r}}_\alpha)\cdot\mathbf{p}_\alpha(\dot{\mathbf{r}}_\alpha) \tag{2}$$

$$T = \sum_{\alpha=1}^{N_A} T_\alpha(\dot{\mathbf{r}}^{N_A}) \tag{3}$$

where N_A is the total number of atoms and $\mathbf{r}^{\mathrm{N_A}} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_\alpha, \dots, \mathbf{r}_{\mathrm{N_A}})$ is the set of atomic coordinates. For atom α ϵ N_A, m_α is the atomic mass, $\mathbf{f}_\alpha^{\mathrm{ext}}$ the external force vector, $\mathbf{u}_\alpha$ is the displacement vector, and $\mathbf{p}_\alpha$ is the momentum. The dot notation "**.**" denotes the derivative with respect to time. $\mathbf{p}_\alpha$ is defined as

$$\mathbf{p}_\alpha = m_\alpha \dot{\mathbf{r}}_\alpha = m_\alpha \dot{\mathbf{u}}_\alpha \tag{4}$$

The total Hamiltonian of the system is then given by

$$H\left(\mathbf{r}^{\mathrm{N_A}}(t), \mathbf{p}(t)\right) = T + \Pi$$

$$= \sum_{\alpha=1}^{N_A} \frac{1}{2m_\alpha} \mathbf{p}_\alpha(\dot{\mathbf{r}}_\alpha) \cdot \mathbf{p}_\alpha(\dot{\mathbf{r}}_\alpha) + \sum_{\alpha=1}^{N_A} E_\alpha(\mathbf{r}^{N_A}) - \sum_{\alpha=1}^{N_A} \mathbf{f}_\alpha^{ext} \mathbf{u}_\alpha \tag{5}$$

The well-known Hamiltonian canonical equations of motion are

$$\dot{\mathbf{p}}_\alpha = -\frac{\partial H}{\partial \mathbf{r}_\alpha} \tag{6}$$

$$\dot{\mathbf{r}}_\alpha = -\frac{\partial H}{\partial \mathbf{p}_\alpha} \tag{7}$$

Equations (6) and (7) can be combined to yield

$$m_\alpha \ddot{\mathbf{r}}_\alpha = -\frac{\partial \Pi}{\partial \mathbf{r}_\alpha} = \mathbf{f}_\alpha^{int} + \mathbf{f}_\alpha^{ext}, \forall \alpha \in N_A \tag{8}$$

where $\mathbf{f}_{\alpha}^{int} = -\frac{\partial E^{int}}{\partial \mathbf{r}_{\alpha}}$ is the interatomic force vector. Equation (8) is typically employed in full atomistic simulations.

For the MMD models, we first define the linear shape functions $\aleph$ used in standard FE methodology in order to accommodate reduced degrees of freedom of the original system. Note the consistency of linear shape functions with our earlier assumptions regarding linearity of the continuum region. Notations required to establish the mathematical formulation of MMD are in order. N_r, N_g, N_{psa}, N_{ssa} are the set of rep atoms, ghost atoms, primary sampling atoms, and secondary sampling atoms, respectively. Rep atoms are further divided into two types: interpolating rep atoms (N_{irep}) and non-interpolating rep atoms (N_{nirep}), hence $N_r = N_{irep} + N_{nirep}$. Let $\mathbf{r}^{N_r} = \left(\mathbf{r}_{r1}, \mathbf{r}_{r2}, \dots, \mathbf{r}_{N_r}\right)$ denote the set of coordinates of rep atoms, $\mathbf{r}^{N_g} = \left(\mathbf{r}_{g1}, \mathbf{r}_{g2}, \dots, \mathbf{r}_{N_g}\right)$ ghost atoms, $\mathbf{r}^{N_{irep}} = \left(\mathbf{r}_{irep1}, \mathbf{r}_{irep2}, \dots, \mathbf{r}_{N_{irep}}\right)$ interpolating atoms, and $\mathbf{r}^{N_{nirep}} = \left(\mathbf{r}_{nirep1}, \mathbf{r}_{nirep2}, \dots, \mathbf{r}_{N_{nirep}}\right)$ non-interpolating rep atoms. Thus, we have $\mathbf{r}^{N_A} = \left(\mathbf{r}^{N_r}, \mathbf{r}^{N_g}\right)$, $\mathbf{r}^{N_r} = \left(\mathbf{r}^{N_{irep}}, \mathbf{r}^{N_{nirep}}\right)$, and $\mathbf{r}_{\alpha} = \mathbf{r}_{\alpha 0} + \mathbf{u}_{\alpha}$, $\forall \alpha \in N_A$ where $\mathbf{r}_{\alpha 0}$ denote the initial coordinates of atom α. In addition, the shape function $\aleph_{\gamma}$ of a rep atom γ is defined as follows: if γ represents an interpolating rep atom, then $\aleph_{\gamma}$ is the standard interpolation shape function; if γ represents a non-interpolating rep atom, then $\aleph_{\gamma}$ is unity at itself and is zero at all other atoms. Note that $\aleph_{\gamma}$ is evaluated at the atom positions. Then, coordinates and velocities of the ghost atoms can be interpolated through the following expression as

$$\mathbf{r}_{\beta} = \mathbf{r}_{\beta 0} + \mathbf{u}_{\beta} = \mathbf{r}_{\beta 0} + \sum_{\gamma \in N_r}^{N_r} \aleph_{\gamma}\left(\mathbf{X} = \mathbf{r}_{\beta 0}\right) \mathbf{u}_{\gamma}\ , \forall \beta \in N_g \tag{9}$$

$$\dot{\mathbf{r}}_{\beta} = \sum_{\gamma \in N_r}^{N_r} \aleph_{\gamma}\left(\mathbf{X} = \mathbf{r}_{\beta 0}\right)\dot{\mathbf{r}}_{\gamma}\,, \forall \beta \in N_g \tag{10}$$

Coordinates and velocities of the ghost atoms depend only on the coordinates and velocities of interpolating rep atoms, therefore $\mathbf{r}^{N_g}$ and $\dot{\mathbf{r}}^{N_g}$ can be expressed as a function of $\mathbf{r}^{N_r}$ as follows

$$\mathbf{r}^{N_g} = \Psi(\mathbf{r}^{N_r}) \tag{11}$$

$$\dot{\mathbf{r}}^{N_g} = \Phi(\dot{\mathbf{r}}^{N_r}) \tag{12}$$

Then, total potential energy of the original system is approximated as follows [46]

$$\tilde{\Pi}(\mathbf{r}^{N_r}) = \sum\nolimits_{i}^{N_r} E_i\left(\mathbf{r}^{N_r}, \Psi(\mathbf{r}^{N_r})\right) + \sum\nolimits_{j}^{N_{psa}} w_j E_j\left(\mathbf{r}^{N_r}, \Psi(\mathbf{r}^{N_r})\right) + \sum\nolimits_{k}^{N_{ssa}} E_k(\mathbf{r}^{N_r}, \Psi(\mathbf{r}^{N_r}) - \sum_{\alpha=1}^{N_A} \mathbf{f}_{\alpha}^{ext}\mathbf{u}_{\alpha}(\mathbf{r}^{N_r}) \tag{13}$$

where w_j is the weight associated with the primary sampling atom in element j. Let $N_g^{e_j}$ and $N_{ssa}^{e_j}$ denote the set of ghost atoms and secondary sampling atoms in element e_j, respectively. Assuming one primary sampling atom is employed in e_j, which is the case in the current work, we have

$$w_j = |N_g^{e_j} \backslash N_{ssa}^{e_j}| \tag{14}$$

which is the number of ghost atoms that are not secondary sampling atoms in element j. In other words, w_j is the sum of numbers of primary sampling and non-sampling atoms in element j.

Let $\mathbf{p}^{N_r} = \left(\mathbf{p}_{r1}, \mathbf{p}_{r2}, \dots, \mathbf{p}_{N_r}\right)$ denote the set of momenta of rep atoms. The momentum of an atom α of the original system can then be approximated as follows

$$\hat{\mathbf{p}}_\alpha(\mathbf{p}^{N_r}) = m_\alpha . \sum_{\gamma\in N_r}^{N_r} \aleph_\gamma \left(\mathbf{X}=\mathbf{r}_{\beta 0}\right)\dot{\mathbf{r}}_\gamma = m_\alpha . \sum_{\gamma\in N_r}^{N_r} \aleph_\gamma \left(\mathbf{X}=\mathbf{r}_{\alpha 0}\right)\frac{\mathbf{p}_\gamma}{m_\gamma} \tag{15}$$

Then, the total kinetic energy of the original system can be approximated as a function of $\mathbf{p}^{N_r}$

$$\hat{T}(\mathbf{p}^{N_r}) = \sum_{\alpha=1}^{N_A} \hat{T}_\alpha(\dot{\mathbf{r}}^{N_r}) = \sum_{\alpha=1}^{N_A} \frac{1}{2m_\alpha}\hat{\mathbf{p}}_\alpha(\mathbf{p}^{N_r}) \cdot \hat{\mathbf{p}}_\alpha(\mathbf{p}^{N_r}) \tag{16}$$

With the above expression in hand, total Hamiltonian of the MMD system can be written as

$$\hat{H}\left(\mathbf{r}^{N_r}(t), \mathbf{p}^{N_r}(t)\right) = \hat{T}(\mathbf{p}^{N_r}) + \hat{\Pi}(\mathbf{r}^{N_r}) \tag{17}$$

then the approximated Hamiltonian canonical equations of motion are

$$\dot{\mathbf{p}}_\gamma = -\frac{\partial \hat{H}}{\partial \hat{\mathbf{r}}_\gamma} \tag{18}$$

$$\dot{\mathbf{r}}_\gamma = -\frac{\partial \hat{H}}{\partial \mathbf{p}_\gamma} \tag{19}$$

for $\forall\gamma \in N_r$. Equations (18) and (19) can be combined to yield

$$\mathbf{M}\ddot{\mathbf{r}}^{N_r} = -\frac{\partial \hat{\Pi}(\mathbf{r}^{N_r})}{\partial \mathbf{r}^{N_r}} = \mathbf{f}^{int} + \mathbf{f}^{ext} \tag{20}$$

where the mass matrix **M** and interatomic force vector $\mathbf{f}^{\text{int}}$ are given as follows, respectively

$$\mathbf{M} = \sum_{\alpha=1}^{N_A} m_\alpha (\aleph^{N_r})^{\mathbf{T}}\aleph^{N_r} \tag{21}$$

$$= \begin{bmatrix} \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{r1} \aleph_{r1} & \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{r2} \aleph_{r1} & \cdots & \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{N_r} \aleph_{r1} \\ \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{r1} \aleph_{r2} & \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{r2} \aleph_{r2} & \cdots & \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{N_r} \aleph_{r2} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{r1} \aleph_{N_r} & \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{r2} \aleph_{N_r} & \cdots & \sum_{\alpha=1}^{N_A} m_\alpha \aleph_{N_r} \aleph_{N_r} \end{bmatrix}$$

$$\mathbf{f}^{int} = -\sum\nolimits_{i}^{N_r} \frac{\partial E_i\left(\mathbf{r}^{N_r}, \Psi\left(\mathbf{r}^{N_r}\right)\right)}{\partial \mathbf{r}^{N_r}} - \sum\nolimits_{j}^{N_{psa}} w_j \frac{\partial E_j\left(\mathbf{r}^{N_r}, \Psi\left(\mathbf{r}^{N_r}\right)\right)}{\partial \mathbf{r}^{N_r}} - \sum\nolimits_{k}^{N_{ssa}} \frac{\partial E_k\left(\mathbf{r}^{N_r}, \Psi\left(\mathbf{r}^{N_r}\right)\right)}{\partial \mathbf{r}^{N_r}} \quad (22)$$

where $\aleph^{N_r} = \left(\aleph_{r1}, \aleph_{r2}, \dots, \aleph_{N_r}\right)$ is the set of interpolation shape functions.

2.3 DIAGONALLY LUMPED MASS MATRIX

The mass matrix given by Equation (21) is called the Consistent Mass Matrix (CMM). Another widely-used type of mass matrix is the Diagonally Lumped Mass Matrix (DLMM). DLMM is preferred over CMM due to the smaller computational and storage costs in general, especially in explicit time integration. Further, direct lumping naturally covers the case of concentrated (point) mass being a natural part of model building. The diagonal entries of DLMM are

$$m_\gamma^{diag} = m_\gamma + \sum\nolimits_{e}^{T_\gamma^e} \frac{\sum_i^{N_g^e} m_i}{N_{node}^e}, \forall \gamma \in N_r \quad (23)$$

where T_γ^e denotes the number of elements sharing rep atom γ, and N_{node}^e denotes the number of nodes of element *e*. Then the DLMM $\mathbf{M}_L$ in the MMD formulation can be expressed as

$$\mathbf{M}_L = \begin{bmatrix} m_{r1}^{diag} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & m_{N_r}^{diag} \end{bmatrix} \tag{24}$$

Then, Equation (20) can be rewritten as

$$\mathbf{M}_L \ddot{\mathbf{r}}^{N_r} = \mathbf{f}^{int} + \mathbf{f}^{ext} \tag{25}$$

Equation (25) will be employed in the numerical tests and examples in this work.

2.4 THERMOSTAT

Since MMD is an energy-based method, finite temperature MMD simulations can be enabled by directly employing any MD thermostats such as the Nosé-Hoover and Berendsen thermostats. In this paper, the Berendsen thermostat [80] is implemented within the MMD framework as follows

$$\mathbf{M}_L \ddot{\mathbf{r}}^{N_r} = \mathbf{f}^{int} + \mathbf{f}^{ext} - \gamma \left(1 - \frac{T_{target}}{T_{current}}\right) \mathbf{M}_L \dot{\mathbf{r}}^{N_r} \tag{26}$$

where γ is dissipation coefficient; T_{target} and $T_{current}$ are the target and current temperatures of the system, respectively. The extra term on the right hand side of Equation (26) that is not present in Equation (25) is employed to control temperature of the system. It should be noted that MMD is currently a zero temperature dynamics method and it is left as a future work to extend it to finite temperature as discussed in the conclusion.

The thermostat is applied on every degree of freedom of the system regardless of where the degree of freedom lies, atomistic or continuum region. In this way, the overall temperature of the entire system is kept under control. As an alternative, the thermostat could be applied to a group of atoms in the atomistic or continuum region. However, this approach is likely to be incapable of regulating the overall temperature of the entire system since it requires transmission of waves between atomistic and continuum. As discussed earlier in the section of *wave reflections* of previous Chapter, multiscale methods try to damp out the high frequency waves since they cannot be represented in the coarse region. As a result, thermal phonons cannot travel between the two descriptions, hence are unable to transmit the temperature effects. As will be discussed next in the section of *wave reflections* of this Chapter, MMM cannot represent high frequency waves in the coarse description too. Therefore, the thermostat is applied everywhere in the entire region in order to ensure that every subregion has the same temperature. This is particularly important in case of an adaptive refinement, where the temperature of the refined subregion is interpolated from the temperature of the previously coarse region.

2.5 GHOST FORCES

As discussed earlier, none of the energy-based multiscale methods is able to completely eliminate the ghost forces. This claim also holds for the MMM method. However, thanks to the atomistic description, MMM is able to mitigate ghost forces to a great extent, especially when compared to other multiscale methods.

Ghost forces in an MMM model can be observed by applying it to a relaxed full atomistic model. Let’s consider a single triangular element full of atoms with zero forces; and, let’s apply an MMM model with one primary sampling atom at the center and three interpolating rep atoms at the corners of the element. The force between an interpolating rep atom and a non-interpolating ghost atom in its range is now multiplied by a weight as a result of the MMM model. This weighted force of the non-interpolating rep atom is then extrapolated to the interpolating rep atoms with respect to its shape function. Obviously, the resultant forces of interpolating rep atoms will be different than before due to the extrapolated contributions, not only due to the considered pair interaction but due to all other pair interactions. This difference in resultant forces between the relaxed full atomistic and the MMM model are the ghost forces.

2.6 WAVE REFLECTIONS

As discussed earlier, wave reflection is an important phenomenon for multiscale methods as it may affect the results adversely. In particular, a method should allow low frequency waves to pass from continuum region to atomistic region; and, should not allow high frequency waves reflecting from atomistic/continuum interface. MMM mitigates the effects of wave reflection by using a thermostat in the atomistic region as employed by Curtin et al. [37] as discussed earlier. In this context, problems are categorized into two classes: (i) the wave is the primary concern of a problem such as shock impact and (ii) the wave is not the primary concern of a problem such as nanoindentation. In the first class of problems, the wave front is required to be captured closely by the multiscale method. In reference to this requirement, the capacity of MMM to capture the wave front is demonstrated by solving a 1-D and 2-D wave problems where the wave is concerted with a full atomistic region. In the second class of problems, the adopted approach is shown to be effective by several numerical examples, such as crack propagation and nanoindentation. Of course, the theory suggests that there is room to improve the accuracy of the method by special treatment of wave reflection. However, demonstrated accuracy suggests that such a special treatment is not indispensible for the MMM method.

2.7 MMM VS. QC

MMM method is very similar to QC method and this similarity is often questioned. Since there are many variations to the original QC method, MMM method should be compared to the version with the most common grounds: fully nonlocal QC with the central summation rule. The fully nonlocal approach employs an atomistic description in the entire domain and central summation rule samples energy within the elements. The atomistic description and the sampling scheme are the features that the MMM method shares with this particular QC version. However, the sampling schemes are not identical for the two methods for all cases. If the comparison is restricted to linear elements, as both methods are first presented with, the sampling schemes are identical; and, therefore the methods are identical too. On the contrary, it is rather difficult to compare the methods for higher order elements because it takes another effort to establish the sampling schemes for the higher order elements. Yang et al. presented the analysis on different sampling schemes that, in turn, concluded with suggestions for the most optimum schemes [79]. To the best of my knowledge, there has not been a unifying comprehensive presentation on sampling schemes of QC method. As such, QC method offers many potential sampling schemes; but, the optimum choice is not obvious.

2.8 PRESCRIBED ADAPTIVITY

In case where the main focus of a problem is propagating waves, a prescribed adaptive scheme is proposed to capture the wave front. The current scheme is different than the actual adaptivity scheme presented later in the Adaptivity Chapter. The current scheme is developed because the actual adaptivity of the MMM software was not developed back then. However, the results should not differ considerably since both schemes successfully capture waves with very high accuracy. In the prescribed adaptivity scheme, an initial mesh that covers the entire domain is built in the beginning. The elements are turned on and off adaptively during the simulation. In particular, elements around the wave front are turned off and the region is then refined to full atomistic resolution by switching the ghost atoms into rep atoms. The rest of the domain is kept coarse by the remaining elements. After the wave has moved away from a full atomistic subregion, the elements are turned back on and the subregion is coarsened by switching the rep atoms into ghost atoms. Thus, the only criterion for switching an atom type is that if the atom is just included in or excluded from the full atomistic region. The full atomistic resolution is implemented only at a certain distance before and after the wave front. In other words, the full atomistic region is carried along with the wave front. The switching of full atomistic region is achieved manually in the current work, hence not automated as in the actual adaptivity of the MMM software. The full atomistic region is prescribed to follow a path that tracks the wavefront as observed from the full atomistic simulation.

An adaptivity scheme must conserve certain fundamental physical quantities such as mass. In order to ensure conservation of mass and momentum, mass and velocity are mapped from ghost atoms to rep atoms and vice versa. It is important to note that ghost atoms do not have velocities or masses. When an element is to be turned on, the mass of the element, which is defined as the sum of masses of all atoms in the element, is equally distributed to the interpolating rep atoms of the element. When the element is turned back off, the procedure is reversed. Thus, the mass is always conserved. On the other hand, if an element is to be turned off, then the velocities of interpolating rep atoms are linearly mapped to the ghost atoms. This way, the momentum is conserved.

2.9 NUMERICAL EXAMPLES

The MMD method is tested for four numerical examples to demonstrate dynamics features: (i) one-dimensional (1-D) wave propagation; (ii) two-dimensional (2-D) wave propagation; (iii) 2-D crack propagation; and (iv) 2-D modal analysis. In all examples, the common settings include Lennard-Jones (LJ) potential with parameters $\sigma = 1$ and $\varepsilon = 1$ representing the interatomic interaction and 1 g/mole as the mass. The interactions are truncated beyond the second nearest neighbors. Initial spacing (r_0) is set to $2^{1/6}$ Å, the equilibrium spacing between two atoms for the LJ potential. Following numerical examples concentrate on the accuracy of the method and the efficiency of the method will be discussed in the last Chapter.

2.9.1 1-D Wave Propagation

The first numerical example is a 1-D wave propagation that is simulated to demonstrate the capability of the method to capture propagating waves. The model is consisted of an atom chain of 1,201 atoms. The atoms are initially relaxed by static energy minimization. The models are fixed at two ends by two atoms in every dimension (Figure 4). The MMD model is coarsened by line elements of size $4r_0$, which consists of one primary sampling atom and two secondary sampling atoms. An initial Gaussian wave as given by Equation (27) is imposed by means of displacement in the middle of the chain [33]. See right half of the symmetric MMD model in Figure 4.

$$u(x,t=0)=\begin{cases} A\dfrac{e^{-(x/\sigma)^2}-u_c}{1-u_c}\left[1+b\cos\left(\dfrac{2\pi x}{H}\right)\right], & |x|\le L_c \\ 0, & |x|>L_c \end{cases} \tag{27}$$

where the following values for the various parameters in the equation are employed in the simulation: σ = 20, H = σ/4, A = 0.01, b = 0.2, L_c = 4σ, and u_c = 0.

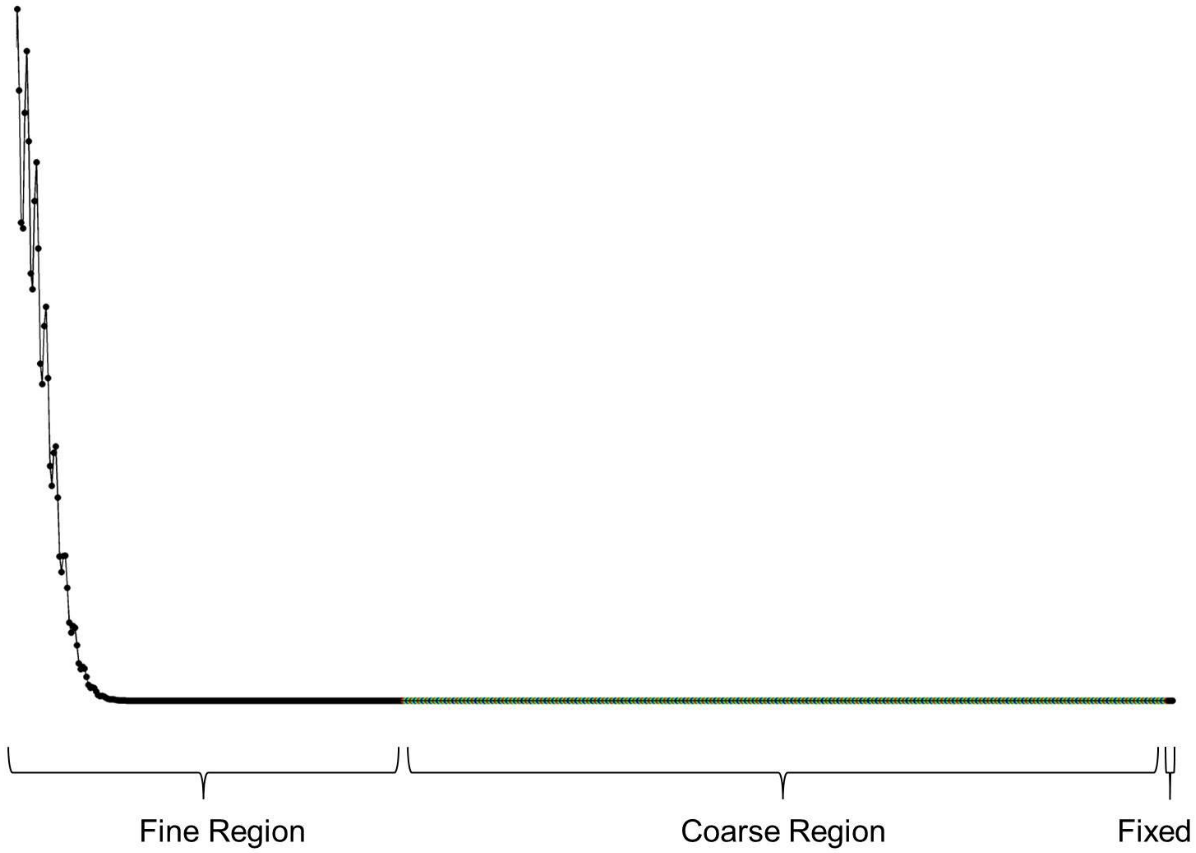

Figure 4. Right half of the symmetric 1-D wave model that includes the full atomistic region where the wave is initiated, coarsened region, and the fixed end.

The simulations are run for 280 fs at a timestep of 0.1 fs. Full atomistic region is initially applied in the middle of the domain where the wave is introduced; then, travels to both ends step-by-step along with the wave front. As the number of turned-on elements is altered during the simulation, the portions of full atomistic and coarse region are varied as well. The portion of full atomistic region is varied between 33% and 58%. Figure 5 shows several snapshots from the simulations of full atomistic (a, c, e) and MMD (b, d, f). The waves of full atomistic and MMD simulations match perfectly throughout the simulation.

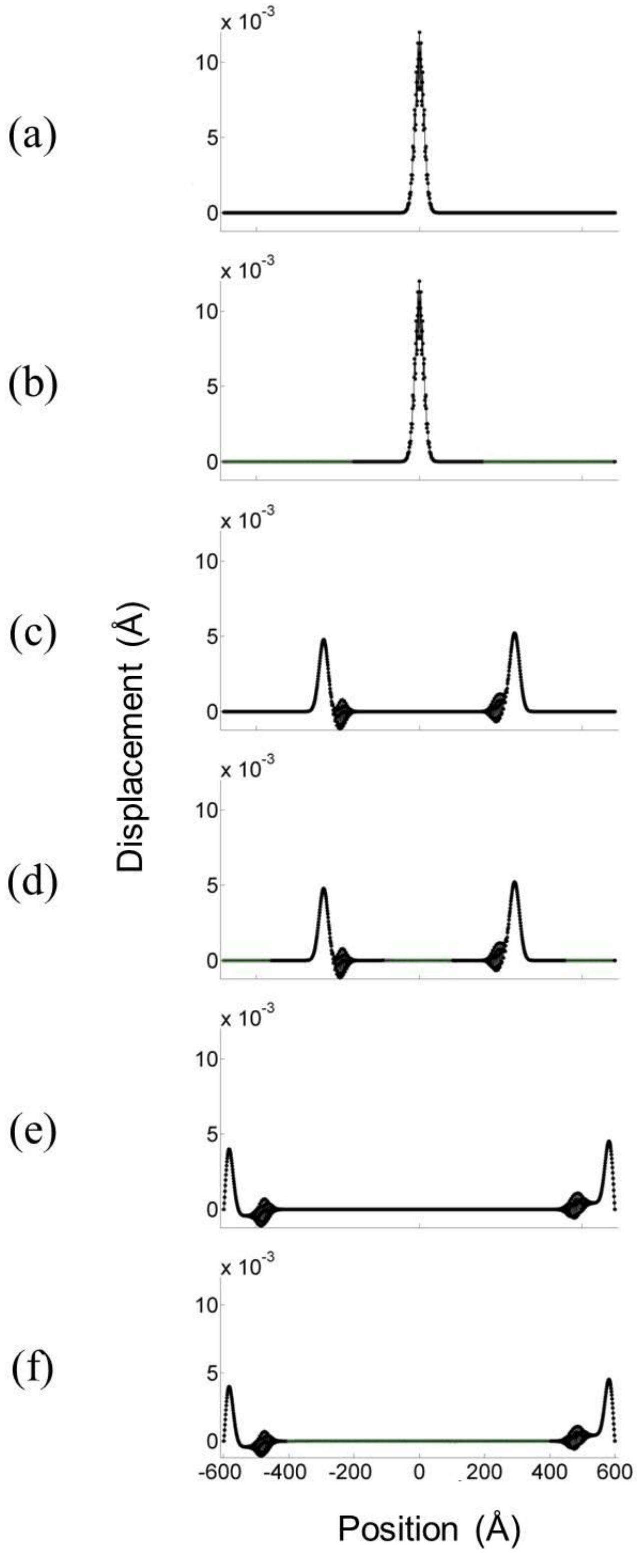

Figure 5. 1-D wave propagation of full atomistic and MMD models at their initial (a, b), intermediate (c, d) and final (e, f) states, respectively.

The results were quantified by monitoring the kinetic energy of a subregion in the middle of the domain during the simulation (Figure 6). Some fluctuations occurred in the beginning due to introduction of the wave, and stabilized later. Then, the kinetic energy gradually decreased as the wave left the monitored region. The curves of full atomistic and MMD are again indistinguishable.

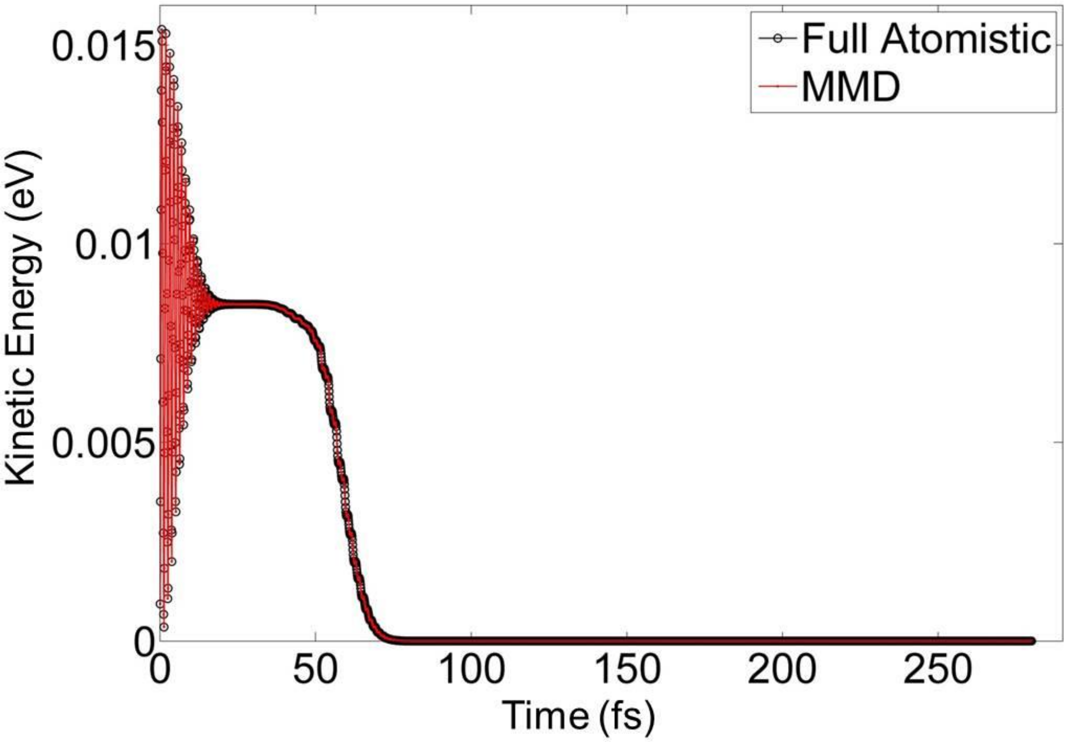

Figure 6. Kinetic energy transfer in 1-D wave propagation of full atomistic and MMD model during the simulation.

2.9.2 2-D Wave Propagation

The second numerical example is a 2-D wave propagation that is simulated to demonstrate the capability of the method to capture propagating wave in a higher dimension. The model is consisted of a square of 16,577 atoms in hexagonal configuration. The atoms are initially relaxed by static energy minimization. The models are fixed in *x* and *y* dimensions at the left-bottom corner and in *x* dimension at the upper-left corner (Figure 7). The MMD model is coarsened by 8,192 elements of size $2r_0$, which consists of one primary sampling atom and the rest of the atoms in the element are set as secondary sampling atoms. As in the 1-D wave propagation example, an adaptive scheme is employed to capture the wave front. See Figure 7 for the initial MMD model where the center region employs full atomistic resolution and the rest of the domain is coarsened with 7,931 elements. A circular wave with respect to Equation (28) is introduced by means of displacement at the center of the model [53].

$$u_r(r,t=0)=\begin{cases}\frac{A}{A-u_c}\left[1+b\cos\left(\frac{2\pi r}{H}\right)\right]\left(Ae^{-(r/\sigma)^2}-u_c\right)\hat{e}_r,\ r\le r_c\\ 0,\ r>r_c\end{cases} \tag{28}$$

where *r* is the radial distance to the center of the domain and the different parameters are set to be following values in the simulation: σ = 15, $H = \sigma/4$, A = 0.015, b = 0.1, $r_c = 5\sigma$, and $u_c = Aexp(-r_c/\sigma)^2$.

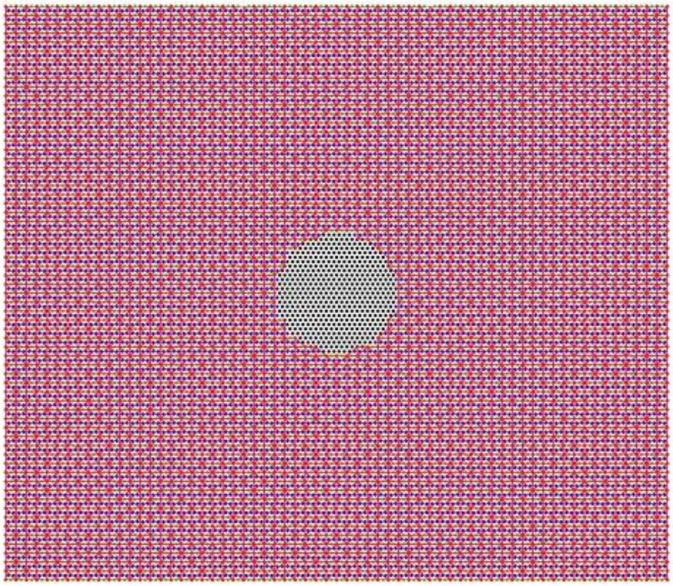

Figure 7. 2-D MMD model with the center in full atomistic resolution and the rest coarsened by the mesh.

The circular wave can be visualized by the initial state of full atomistic and MMD simulations (Figure 8). Each simulation is run for 80 fs at a timestep of 0.001 fs. The portion of full atomistic region is increased from 2% to 69% during the adaptive MMD simulation since the wave front is spread from the center to a larger area. Final states of the wave for full atomistic and MMD simulations are shown Figure 8. In addition, wave trajectories are compared along a line from the center to the left edge of the domain (Figure 9) at several timesteps for full atomistic and MMD. The comparisons suggest that the wave front is captured successfully by the MMD model, but there are some small deviations elsewhere. The wave front is captured well because it is always resolved with full atomistic resolution during the simulation. The deviation, on the other hand, is probably caused by the coarsening applied to regions where the wave front has left. The deviation is expected because coarsening is applied to a lattice structure that is not as perfect as at the initial state anymore. There is still the high frequency content in the full atomistic region when it is switched to coarse region again. However, the important part of wave is the front where the critical physical phenomena occur. Therefore, capturing the front of a wave should be sufficient to capture the governing physical phenomena for the problem.

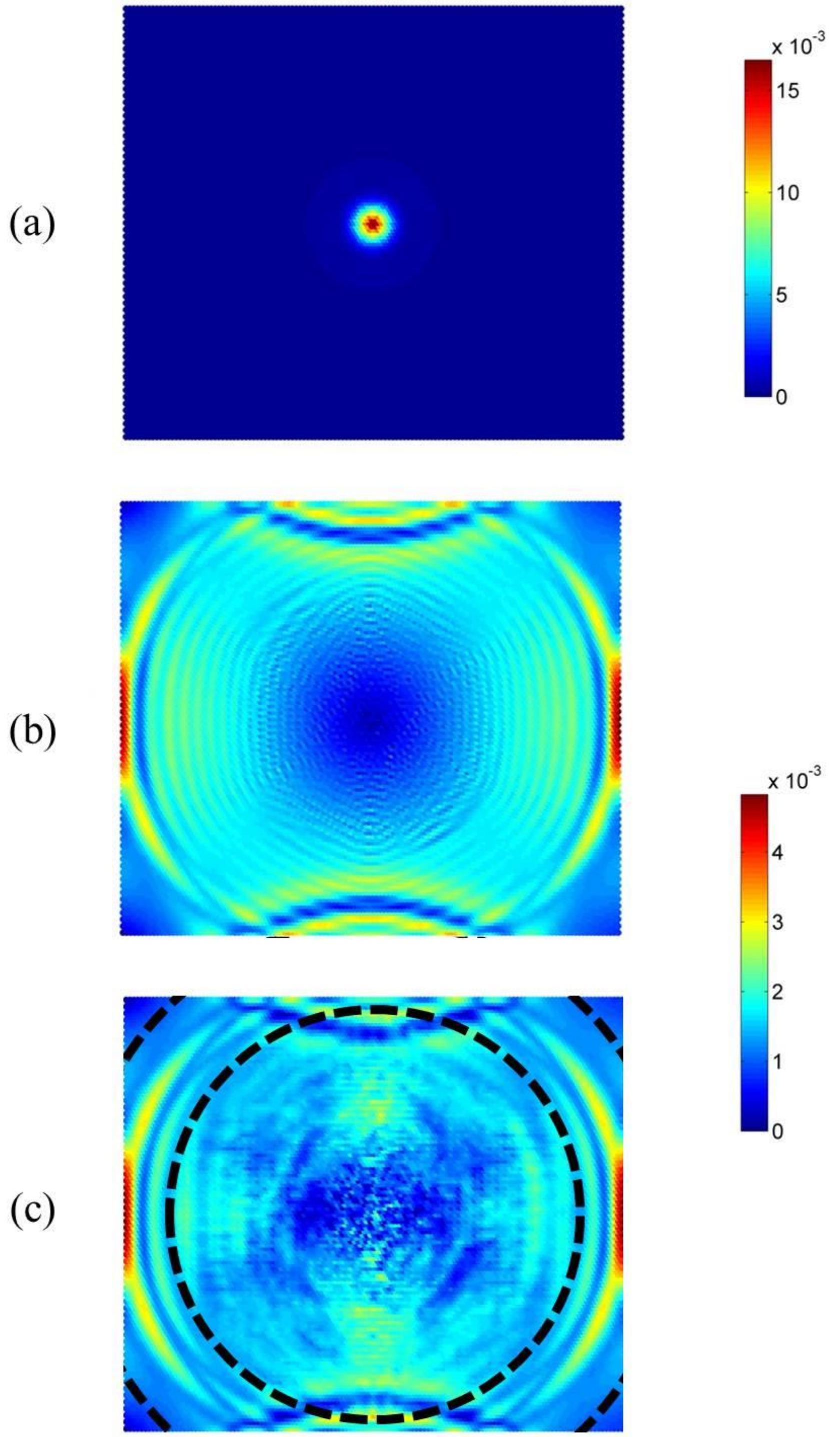

Figure 8. 2-D wave propagation of full atomistic and MMD models at their initial state (a) and final states (b, c), respectively. Dashed lines in the MMD final state indicate borders of the full atomistic region.

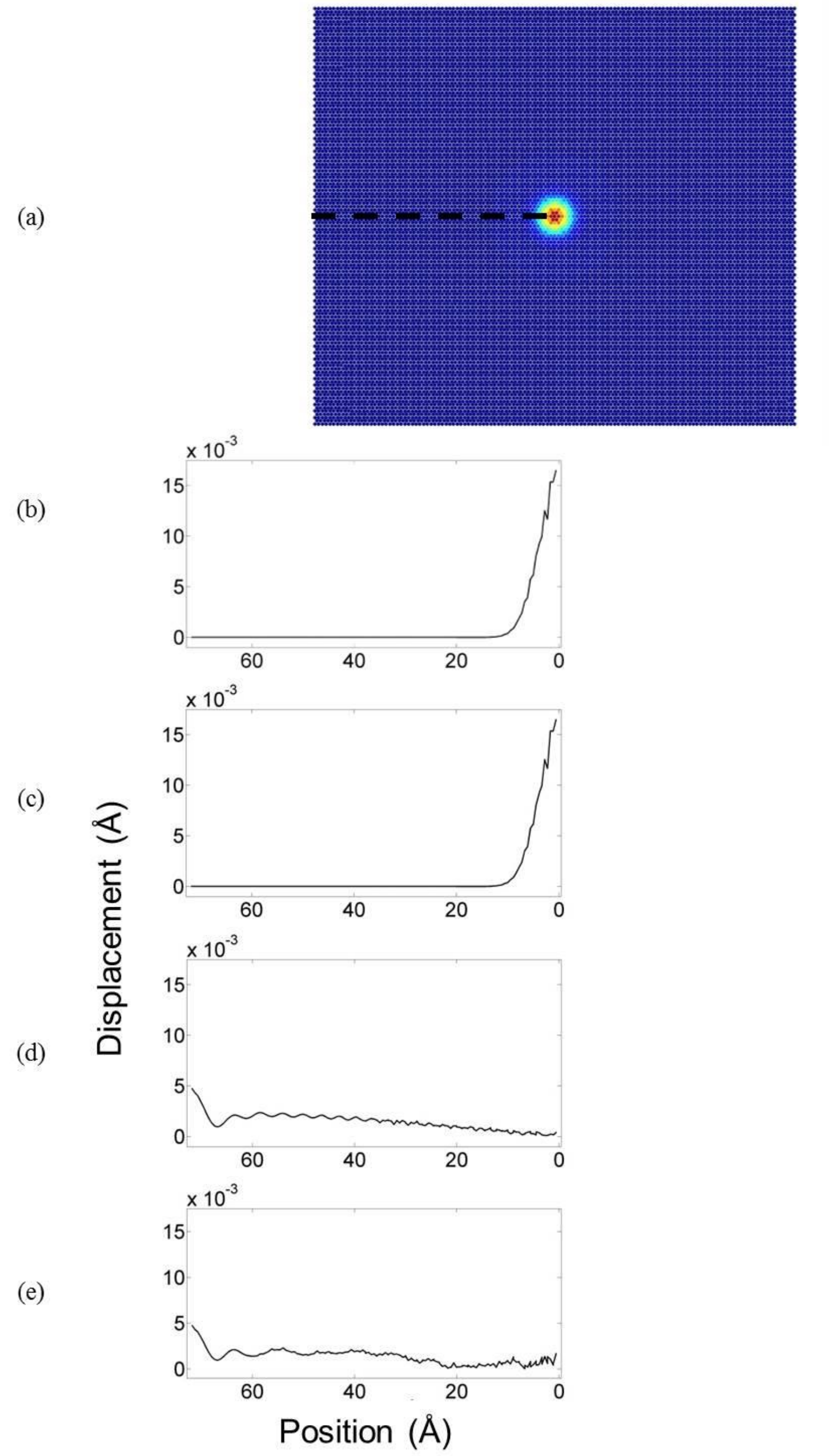

Figure 9. 2-D wave propagation of full atomistic and MMD models compared along the dashed line in (a) at their initial (b, c) and final (d, e) states, respectively.

As in the 1-D case, the results are quantified by monitoring the kinetic energy of subregion at the center of the domain (Figure 10). At first, the kinetic energy increased when the wave started to move, fluctuated for some time, and then stabilized. Later, the kinetic energy decreased as the wave left the monitored region. A discrepancy between the curves of full atomistic and MMD is observed after some time. This might be due to the fact that monitoring is performed over a fixed region that includes some elements in part, not in whole. In support to this argument, time of start of coarsening corresponds to 1,200 fs where the discrepancy first appears at ~1,350 fs. Despite the discrepancy, MMD followed the full atomistic results quite well.

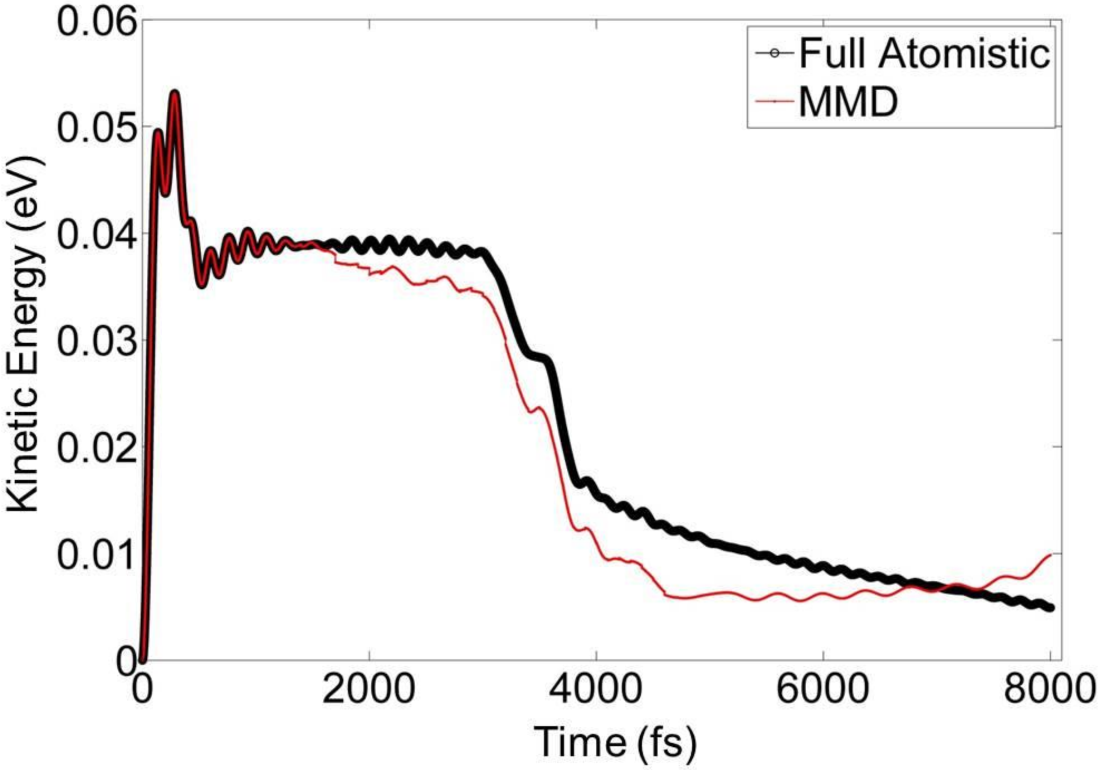

Figure 10. Kinetic energy transfer in 2-D wave propagation of full atomistic and MMD model during the simulation.

2.9.3 2-D Crack Propagation

The third numerical example is a 2-D crack propagation that is simulated to demonstrate the capability of the method to capture propagation of cracks. The model is consisted of a rectangular prism of 10,701 atoms in a hexagonal configuration with two notches in the middle. The atoms are initially relaxed by static energy minimization. The models are fixed in all dimensions at two layers of atoms from the left and right ends (Figure 11). The MMD model is coarsened by 16 triangular elements of size $29r_0$, which consists of one primary sampling atom in the middle. Correspondingly, 69% of atoms are ghost atoms in the MMD model, in which 4 layers from the surfaces and 12 layers from the notches are covered with non-interpolating rep atoms. The MMD model is shown at the top of the right panel in Figure 11.

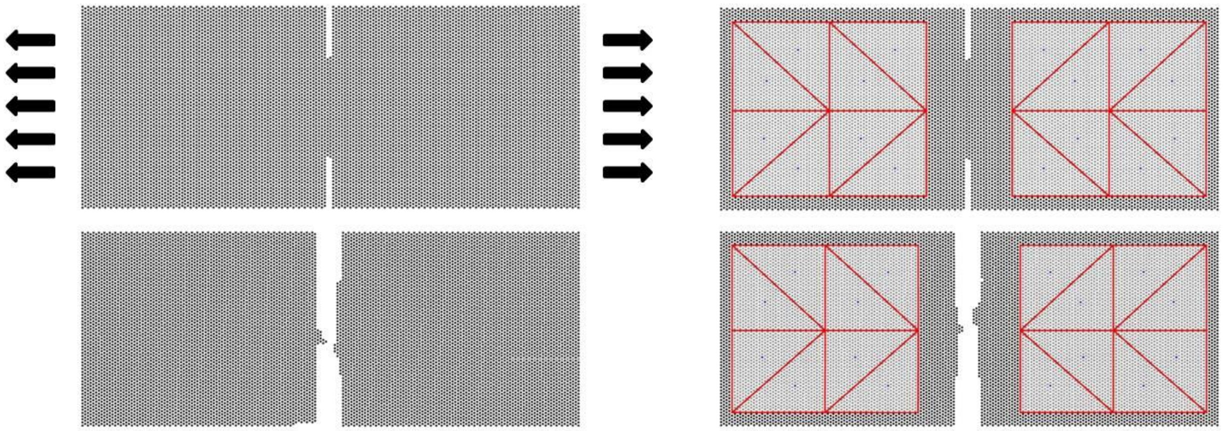

Figure 11. Snapshots from simulations of full atomistic MD (left) and MMD (right) at the beginning (top) and end (bottom).

Temperature of the system is kept constant at 10 K by the Berendsen thermostat with damping-to-timestep ratio of 1. The simulations are run for 405 ps at a timestep of 0.5 fs. At each timestep, the fixed atoms at the left and right ends are displaced in the opposite *x*-directions in tension at an amount corresponding to a strain rate of 10^8 s^{-1}.

Stress is measured as the average of absolute forces in the *x*-direction of the atoms at the fixed ends, and engineering strain is measured in the conventional manner. The stress-strain curves of full atomistic and MMD solutions are compared in Figure 12. As apparent from the figure, the stress-strain curves agree very well. Note that the elastic modulus, ultimate strength, and ultimate strain can be estimated successfully by the MMD method.

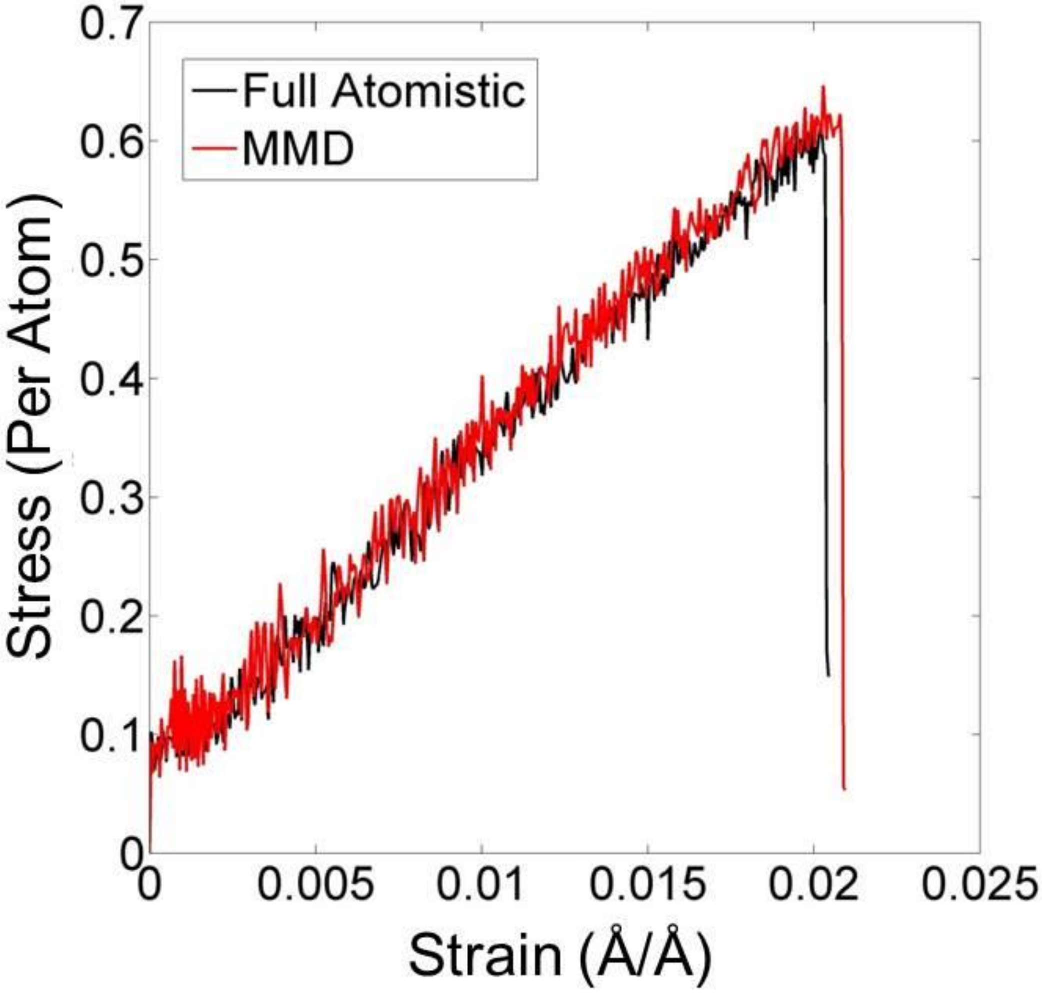

Figure 12. Stress–strain curves of 2-D crack propagation for full atomistic MD and MMD.

2.9.4 2-D Modal Analysis

The fourth numerical example is a 2-D modal analysis that is simulated to demonstrate the capability of the method to perform modal analysis. For this purpose, the linearized equations of motion without damping are given by Equation (29) as follows

$$\mathbf{M}_L\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = 0 \tag{29}$$

where $\mathbf{M_L}$ is the DLMM and $\mathbf{u}$ is the displacement vector; $\mathbf{K}$ is the stiffness matrix populated with stiffness values between pairs of neighbor atoms calculated with respect to the second derivative of the given potential function at the equilibrium position. Fourier transform of Equation (29) is taken so that Equation (30) is obtained

$$(\mathbf{K} - \omega^2\mathbf{M}_L)\tilde{\mathbf{u}} = 0 \tag{30}$$

where $\tilde{\mathbf{u}}$ is the Fourier conjugate of $\mathbf{u}$, such that $\mathbf{u}(\mathrm{t}) \Leftrightarrow \tilde{\mathbf{u}}(\mathrm{w})$. When multiplied by $\mathbf{M}_L{}^{-1}$, Equation (30) represents an eigenvalue problem where $\tilde{\mathbf{u}}$ is the mode shape (eigenvector) and ω is the natural frequency (eigenvalue). Eigen-analysis is conducted to obtain the three non-zero smallest natural frequencies. The model is consisted of a rectangular beam of 33,313 atoms in hexagonal configuration. One full atomistic model and four MMD models of uniform meshes with different element sizes are compared (Figure 13). Each element included one primary-sampling atom and the rest of the atoms in the element are set as non-sampling atoms. Element sizes, arrangement of elements, and number of elements of MMD models are listed in Table 3. Relative errors of MMD models with respect to the full atomistic solution are given by Equation (31).

$$\varepsilon_{rel} = \frac{|\omega_{MMM} - \omega_{FA}|}{|\omega_{FA}|} x100 \tag{31}$$

Table 3. Models, element sizes, arrangements and number of elements of MMD models in the convergence study.

Model	Element size (r_0)	Arrangement of elements	Number of elements
b	4.01	8x8	4,096
c	8.02	4x4	1,024
d	16.04	2x2	256
e	32.07	1x1	64

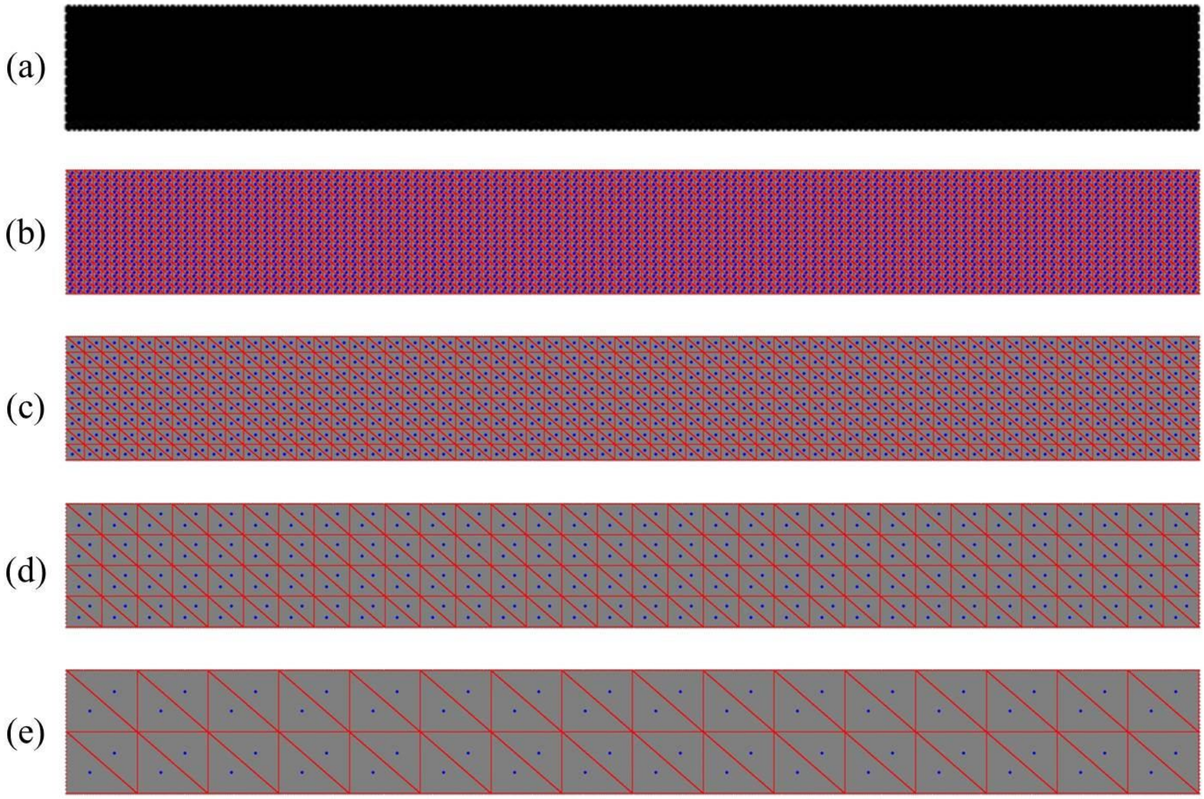

Figure 13. The beam model employed to perform model analysis and convergence study of the MMD method: (a) full atomistic model. (b–e) MMD models with element sizes from fine to coarse.

The three smallest natural frequencies obtained with each model are listed along with relative errors in Table 4. The relative errors suggest monotonic convergence for all three natural frequencies as the element size is decreased.

Table 4. Smallest three natural frequencies and relative errors of full atomistic (a) and MMD models with element sizes varying from fine to coarse (b-e).

Model	**ω_1**	**Rel. error (%)**	**ω_2**	**Rel. error (%)**	**ω_3**	**Rel. error (%)**
a	0.00119399	-	0.00282537	-	0.00519850	-
b	0.00119605	0.17	0.00283294	0.27	0.00521413	0.30
c	0.00120636	1.04	0.00286477	1.39	0.00528371	1.64
d	0.00120981	1.32	0.00292323	3.46	0.00546742	5.17
e	0.00121571	1.82	0.00312044	10.44	0.00604410	16.27

3.0 ADAPTIVITY

Adaptivity is essential to any concurrent atomistic/continuum coupling method. Considering a moving region of interest such as wave or dislocation on a path, the method can either model the entire path *a priori* at fine resolution or adaptively refine and coarsen the path. Multiscale methods that lack adaptive features are limited to the first option; they can only be employed to solve problems where the path can be guessed *a priori* and model the entire path in fine resolution, which severely limits problem size. Even so, efficiency of the implementation will not be as good as an adaptive method due to overuse of fine resolution. Although many multiscale methods are introduced, only a few of them feature adaptivity. Shenoy et al. [29], Park et al. [49], Kwon et al. [81], Shimokawa et al. [82], and Shan et al. [27] introduced adaptivity to the QC method. Moseley et al. [25], Moseley et al. [26], and Gracie et al. [8] presented adaptive features of the BD method. In addition, Marenic et al. presented review of adaptive methods with a focus on the QC and BD methods [52]. Xiong et al. added adaptivity to the CAC method [62]. Another multiscale method that presented adaptive features is the CG method by Praprotnik [67] and Heyden [68]. All these adaptive methods including the current work are empirical in a sense that they are not based on theoretical estimation of errors. In their work, Oden et al. presents rigorous mathematical calculations to serve as a sound basis to estimate the modeling error so that they can control this error adaptively [28]. Another important note about adaptive multiscale methods is that the theoretical framework of a concurrent coupling method would dictate how that method performs adaptive analysis, and therefore obtained results are still characterized by the theoretical framework rather than the adaptivity scheme.

Adaptivity means automatic tracking of moving regions of interest in order to reduce the number of degrees of freedom that needs to be simulated. The moving region of interest can be a wave front in a shock impact problem or a dislocation in a tensile test. Adaptivity automates the tracking of the moving regions of interest that are not known *a priori*, and thus the simulation is easily setup without much preprocessing. More importantly, adaptivity is crucial for efficiency because it keeps the computationally-expensive fine-resolution regions limited to the regions of interest and prevents overuse. One of the two main functions in adaptive analysis is refinement, which converts coarse regions into fine regions wherever required. Refinement is needed to open the front of the region of interest for it to be able to continue its progress. The second function is coarsening, which converts the fine regions back into coarse regions wherever fine resolution is no longer required. Coarsening is needed to close the moving region of interest so that the target efficiency is retained. Coarsening is especially crucial to keep the method efficient without compromising the accuracy since it would otherwise be too expensive to enlarge the fine regions gradually to everywhere the defects visited. Of course, refinement and coarsening are expected to be activated automatically by the method itself without a need to give *a priori* instructions.

3.1 LITERATURE REVIEW

The core issue of adaptive analysis lies in defining a criterion to determine whether a region shall be modeled with fine or coarse resolution. One way or another, all criteria try to measure the severity of deformation, which characterizes the regions of interest. These criteria call for refinement of the region when they are satisfied and call for coarsening when they are not. These criteria may depend on potential energy [8, 25, 63], centrosymmetry parameter [64], difference between local and global deformations [10], deformation gradient [27], spatial variation in displacement [40], local shear strain [62], interatomic bonds [25], strain energy density between adjacent nodes and consecutive timesteps [83], or difference between smoothed and exact forces [66]. The refinement criterion is mostly imposed in a way that the atoms or nodes are activated for refinement when their values fit into the prescribed range. In addition, atoms that are too close to the activated atoms are also activated for refinement in some implementations [8, 25, 63]. On the other hand, the criterion for coarsening has the opposite relationship such that it is performed when an atom is not activated for refinement any longer. In that, some implementations impose additional requirements such as they allow coarsening only if the candidate coarse description is accurate enough not to bring in an error more than a prescribed value [8, 25, 29, 52]. Similarly, the history of an atom or bond is taken into account such that recently refined entities are excluded from coarsening [25]. Some adaptive techniques perform ad-hoc adjustments of the frequency of calling [8] or tolerance [10] of adaptivity procedures so that they are run frequently enough to capture the relevant physical phenomena.

The adaptive analysis procedures include interpolation and extrapolation of state variables between nodes and atoms. These interpolations are conveniently realized by the FE shape functions [26, 66]. The refinement procedure interpolates for the positions, velocities, and in some cases accelerations of the new atoms that are activated for refinement. Furthermore, thermal vibrations could be added to preserve temperature of the system in a temperature critical application. In the case that atoms in the coarse regions are turned off, i.e., not stored in the memory, lattice sites of these atoms can be approximately reproduced [50]. The coarsening can be realized by shape functions or by least squares fit [8].

Adaptive analysis by means of either refinement or coarsening introduces a disturbance to the system, which may result in intolerable error. By a numerical example, Moseley et al. show that the problem arises from instantaneous conversion of fine regions to coarse regions or vice versa [26]. That is why; the authors carry out the adaptation procedures in several timesteps instead of only one timestep. Gracie et al. staggers the execution of refinement and coarsening procedures in time to alleviate the error [8]. Miller et al. [42] and Shenoy et al. [29] re-establish equilibrium after each adaptation step since the system is considered to be no longer in equilibrium. In a similar way, Miller et al. limits the adaptive analysis to focus on a particular region or limits the counts of adaptation steps in order to restrict adaptation from becoming unduly costly [42]. It should, however, be noted that coarsening will always introduce some sort of an indispensible error since it is essentially an approximation of the full atomistic by construction [8].

The coarse regions of multiscale models can be meshed by either a uniform or a graded mesh. The uniform mesh dictates a constant mesh size all over the region while the graded mesh allows variation in mesh size. The graded mesh is particularly important to multiscale methods since it can be utilized to dissipate part of the high frequency waves in case that the mesh size is reduced down to atomistic resolution near the atomistic/continuum interface. Another benefit of the graded mesh is its capability of reaching very coarse resolution by means of very large elements in the continuum region in order to enhance the efficiency. Despite its benefits, the graded mesh has several significant drawbacks. First of all, generation of a graded mesh is more expensive than generation of a uniform mesh. Second, adaptivity requires a graded mesh to be re-generated every time it executes refinement or coarsening since places of fine and coarse regions are updated. Third, generation of a mesh is followed by building shape functions and atom-element relationships, e.g., designating atoms inside an element. The second and third tasks may become very expensive because the costs of these computations are already high and may become even extremely expensive with frequent updates. Moreover, some adaptivity procedures are recursively run until equilibrium is achieved [40]. As a result, this iterative procure may be prohibitively expensive. Last, graded mesh prevents taking advantage of using multiple timesteps since the timestep of the coarse region is limited by the mesh size [84]. Finally, the uniform mesh is also favored for its simplicity in implementation and generation.

3.2 ADAPTIVE ANALYSIS

In the current work, adaptive analysis of MMM is introduced. In contrast to previous methods that require special atomistic/continuum interface treatment, the underlying MMM framework facilitates easy implementation of the adaptive analysis. The presented adaptivity scheme is simple, effective, and accurate as demonstrated by the results of three numerical examples including 1-D wave propagation, 2-D dislocation, and 3-D nanoindentation. The scheme employs a uniform mesh over the entire region for its aforementioned benefits. Owing to its consistency and robustness, MMM does not need to employ any special technique to smooth the adverse effects of adaptivity.

The adaptivity criteria of MMM are based on potential energy, proximity, and element integrity. First, potential energy is a good indicator of distortion and it is already available [8, 25, 63]. A threshold value is input to the simulation such that atoms with higher potential energies are activated for refinement. According to this criterion, an atom *i* is activated when

$$E_i \geq E_{threshold} \tag{32}$$

where E_i is the potential energy of atom *i*, $E_{threshold}$ is the input threshold value. Second, atoms within an input radius of activated atoms are also activated for refinement in order to include the region that may possibly be influenced by the distortion [8, 25, 63]. This way, it is made sure to refine the regions that may be directly or indirectly affected by the ongoing physical phenomena. Third, all of the atoms inside the elements that include activated atoms are also activated for refinement for the sake of element integrity. In other words, an element is either in the coarse or fine region completely. Of course, only the ghost atoms among the activated atoms are selected and then converted to non-interpolating rep atoms. Activated atoms are required to be assigned with only velocities since their positions are already available thanks to the atomic description. The velocities are interpolated from the corresponding interpolating rep atoms with respect to shape functions. In order to conserve mass, the masses of activated atoms take on their original values and those of the corresponding interpolating rep atoms are decreased in proportion to reduction of the number of ghost atoms they represent. The adaptivity function is called periodically with respect to an input number of iterations. In total, adaptivity is controlled by three parameters: (i) threshold of the potential energy, (ii) radius of influence, and (iii) frequency of update.

3.2.1 Refinement

The refinement procedure is outlined in Figure 14.

Refinement Procedure

1. Return if it is not time to query adaptivity.
2. Activate ghost atoms with potential energies higher than the threshold value, return if none.
3. Activate atoms within a radius of influence of the activated atoms.
4. Activate atoms inside elements that include activated atoms.
5. Convert activated atom types to non-interpolating rep atoms.
6. Interpolate new velocities of activated atoms from corresponding interpolating rep atoms.
7. Default masses of activated atoms and reduce masses of corresponding interpolating rep atoms accordingly.
8. Recalculate energies and forces of all atoms for the current iteration.

Figure 14. Refinement procedure of MMM adaptivity.

3.2.2 Coarsening

The coarsening procedure is the opposite of the refinement procedure. In order for an element that is currently full of non-interpolating rep atoms to be coarsened, none of the rep atoms must be activated for refinement. Only then is the coarsening activated and all atoms inside the element are assigned with the types that the MMM scheme specifies. The positions of atoms are left as is and new shape functions are constructed. This way, the element is able to preserve its distorted configuration in order to keep peace with its environment. Otherwise, the original shape functions would dictate the original configuration that may not fit to the current environment and cause instability. The velocities and masses of the ghost atoms are extrapolated to the corresponding interpolating rep atoms where the former extrapolation is performed by shape functions and the latter extrapolation by arithmetic averaging. These extrapolations are carried out in order to conserve mass and momentum of the element and, in turn, the system. The coarsening procedure is outlined in Figure 15.

Coarsening Procedure

1. Return if it is not time to query adaptivity.
2. Select rep atoms with potential energies lower than the threshold value, return if none.
3. Deactivate elements full of the selected atoms.
4. Deactivate all atoms inside deactivated elements.
5. Convert deactivated atom types to scheme types.
6. Extrapolate velocities and masses of deactivated atoms to corresponding interpolating rep atoms.
7. Recalculate energies and forces of all atoms for the current iteration.

Figure 15. Coarsening procedure of MMM adaptivity.

3.2.3 Conservation Properties

It is a well-known fact that forces derived from a global energy functional conserve linear momentum, angular momentum, and total energy. In addition, the conservation of mass is assured by construction. Analytical proofs to these conservation properties of the MMM method are beyond the scope of this work. After the analytical construction of a method, there are numerical considerations that violate the conservation laws. First is the introduction of a cut-off radius that truncates the interactions beyond a few neighbor shells [75, 85, 86]. Neighbor atoms travelling in and out of this borderline may have their properties accounted for in an on-and-off manner, which would violate the conservation laws. Second, MMM employs a velocity Verlet time integration scheme that is proven to be non-conservative but symplectic, which means showing nearly conservative behavior [75]. The third property is the use of a thermostat to regulate the temperature of the system [21]. A thermostat achieves this regulation by including additional forces on atoms to impose an external force on the system. In spite of these negligible inaccuracies, refinement and coarsening procedures are carefully tailored to conserve mass and momentums of the system. However, the energy cannot be exactly conserved due to inclusion and exclusion of energy approximation in refinement and coarsening, respectively [25]. After all, our own numerical investigations on simple models conclude that the method shows nearly conservative behavior.

3.3 NUMERICAL EXAMPLES

The MMD method is tested for three numerical examples to demonstrate adaptivity features: (i) one-dimensional (1-D) wave propagation; (ii) two-dimensional (2-D) dislocation; and (iii) three-dimensional (3-D) nanoindentation. The interactions are truncated beyond the second nearest neighbor. In the adaptivity scheme, all the considered models start as a fully coarse model but they are automatically refined at the very beginning of the simulation, for instance, due to a wave or surface effects. Following numerical examples concentrate on the accuracy of the method and the efficiency of the method will be discussed in the last Chapter.

3.3.1 1-D Wave Propagation

The first numerical example is a 1-D wave propagation that is simulated to demonstrate the capability of the method to capture wave motion. The model considered here is consisted of a chain of 2,405 atoms. The atoms are separated by 1 Å in order to speed-up propagation of the wave. The model is fixed by two atoms at both ends (Figure 16). The MMD model is coarsened by a mesh of elements of size 100r_0, which included one primary sampling atom in the center and four secondary sampling atoms around the interpolating rep atoms. The interactions are modeled with the Lennard-Jones (LJ) potential with parameters σ = 1 Å and ε = 1 eV and mass is set to 1 g/mole. The adaptivity criteria are set to -0.0555 eV (93% of the initial energy value) potential energy threshold, 150 Å radius of influence, and 50 iterations of update frequency. As discussed later, the first criterion (i.e., potential energy threshold) is adjusted by the user depending on which measure is prioritized: accuracy or efficiency. The latter criteria can be adjusted in a similar fashion so that larger radius of influence or smaller update frequency results with higher accuracy but lower efficiency and vice versa. All three adaptivity criteria in this and other examples are adjusted in an ad-hoc fashion in order to produce good results.

A Gaussian wave including both high and low frequency components is introduced in the center of the model by Equation (33), see right half of it in Figure 17.

$$u(x,t=0)=\begin{cases} A\dfrac{e^{-(x/\sigma)^2}-u_c}{1-u_c}\left[1+b\cos\left(\dfrac{2\pi x}{H}\right)\right], & |x|\le L_c \\ 0, \ |x|>L_c \end{cases} \tag{33}$$

where σ = 20, H = σ/4, A = 0.01, b = 0.2, L_c = 4σ, and u_c = 0.

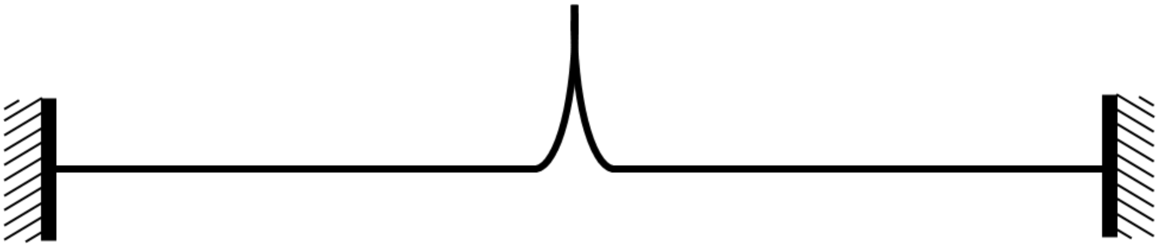

Figure 16. A Gaussian wave is introduced in the center and atoms are fixed at the ends in the 1-D wave propagation example.

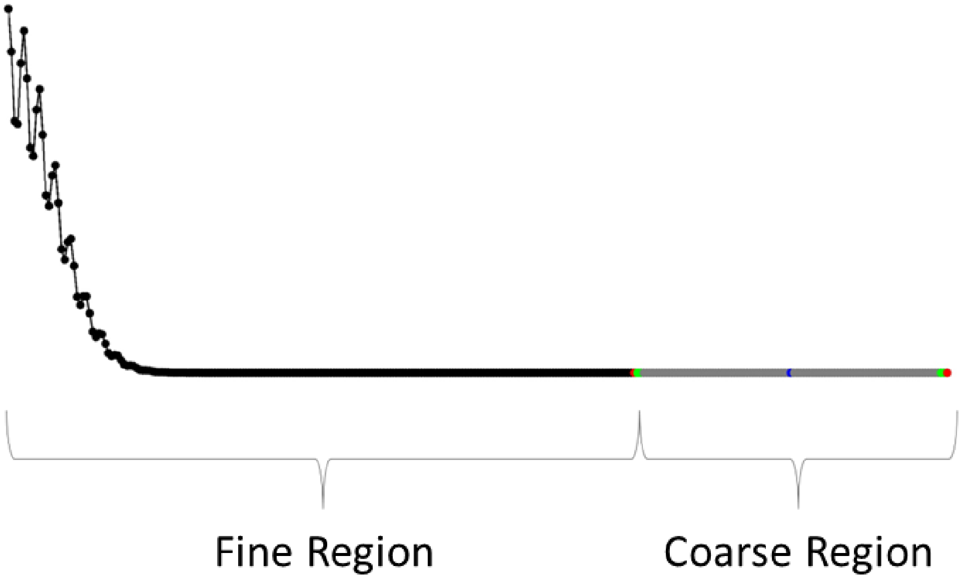

Figure 17. Part of the 1-D wave model includes right half of the wave in full atomistic region and an element in the coarsened region.

The simulations are run for 500 fs at a timestep of 0.1 fs. At the beginning of the simulations, the wave splits into two parts and then they travel opposite to each other towards the ends. As the adaptivity adds and removes rep atoms, the overall ratio of rep atoms starts at 34%, increases to 67%, and ends at 51%. Figure 18 shows trajectories of wave propagation for full atomistic (a, c, e) and MMD (b, d, f). The results are indistinguishable at any step of the simulation. In addition, the kinetic energy of a region as marked in Figure 19 is monitored during the course of the simulations. The region consists of an element with 101 atoms and the kinetic energy is computed as sum of the kinetic energies of these atoms (Figure 20). The kinetic energy slightly increases as low frequency component of the wave enters into the region and then decreases as it starts to leave. At this point, the kinetic energy sharply increases as high frequency components of the wave enter and then sharply decreases upon their leave. The kinetic energies of the full atomistic and MMD being monitored are again indistinguishable.

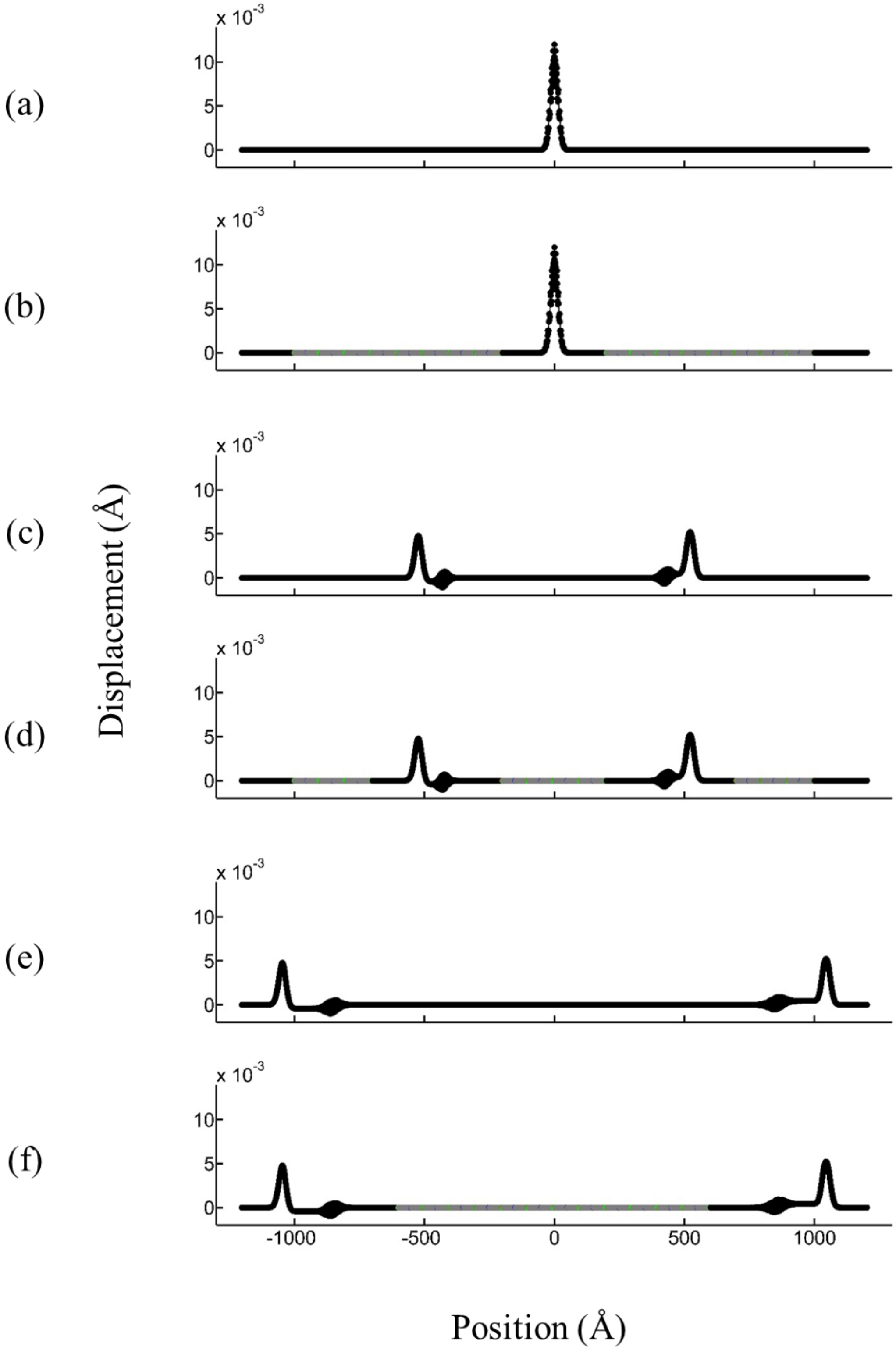

Figure 18. Comparison of trajectories of full atomistic and MMD model at the beginning (a, b), intermediate (c, d) and end (e, f) of the 1-D wave propagation.

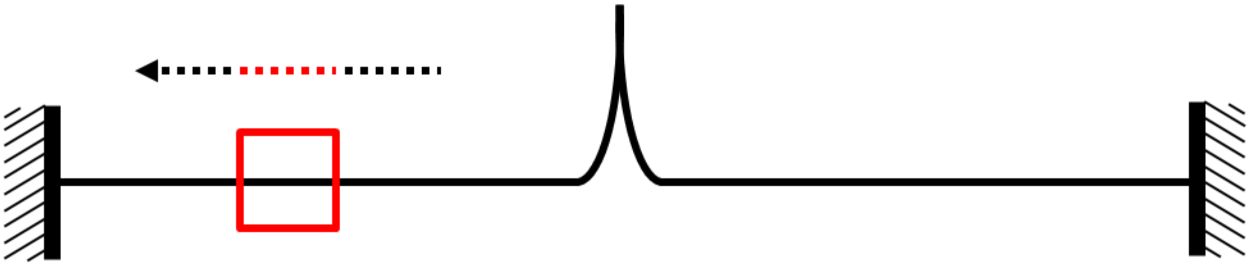

Figure 19. Kinetic energy of the element in the red box is monitored in the 1-D wave propagation example.

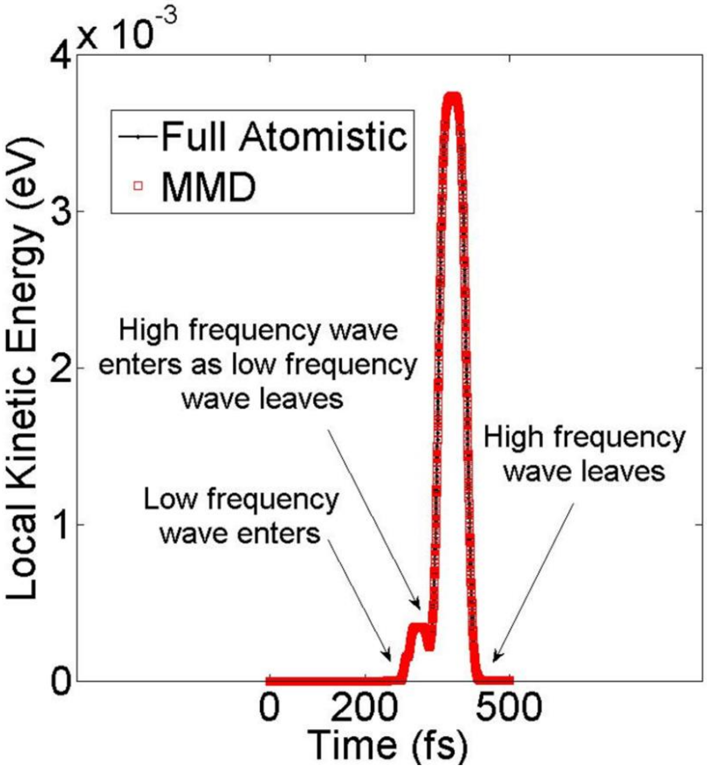

Figure 20. Comparison of monitored kinetic energies of full atomistic and MMD in the 1-D wave propagation example.

3.3.2 2-D Dislocation

The second numerical example is a 2-D dislocation that is simulated to demonstrate the capability of the adaptivity to consistently refine and coarsen as a defect migrates. The 2D model is consisted of a rectangular plate of 21,709 atoms in hexagonal configuration. The atoms are initially relaxed by static energy minimization. The models are fixed by two layers of atoms at the bottom end (Figure 21 - left). The MMD model is coarsened by a mesh of triangular elements of size $20r_0$, which included one primary sampling atom in the center and secondary sampling atoms around the interpolating rep atoms (Figure 21 - right). The interactions are modeled with the Morse potential with parameters D_0 = 0.5093 eV, α = 1.4573 1/Å, and r_0 = 2.58 Å and mass is set to 26.9815 g/mole. The adaptivity criteria are set to -1.6 eV (91% of the initial energy value) potential energy threshold, 10 Å radius of influence, and 100 iterations of update frequency. Temperature of the system is kept constant at 1 K by the Berendsen thermostat. The models are cropped at their upper-left corners in order to trigger dislocation nucleation.

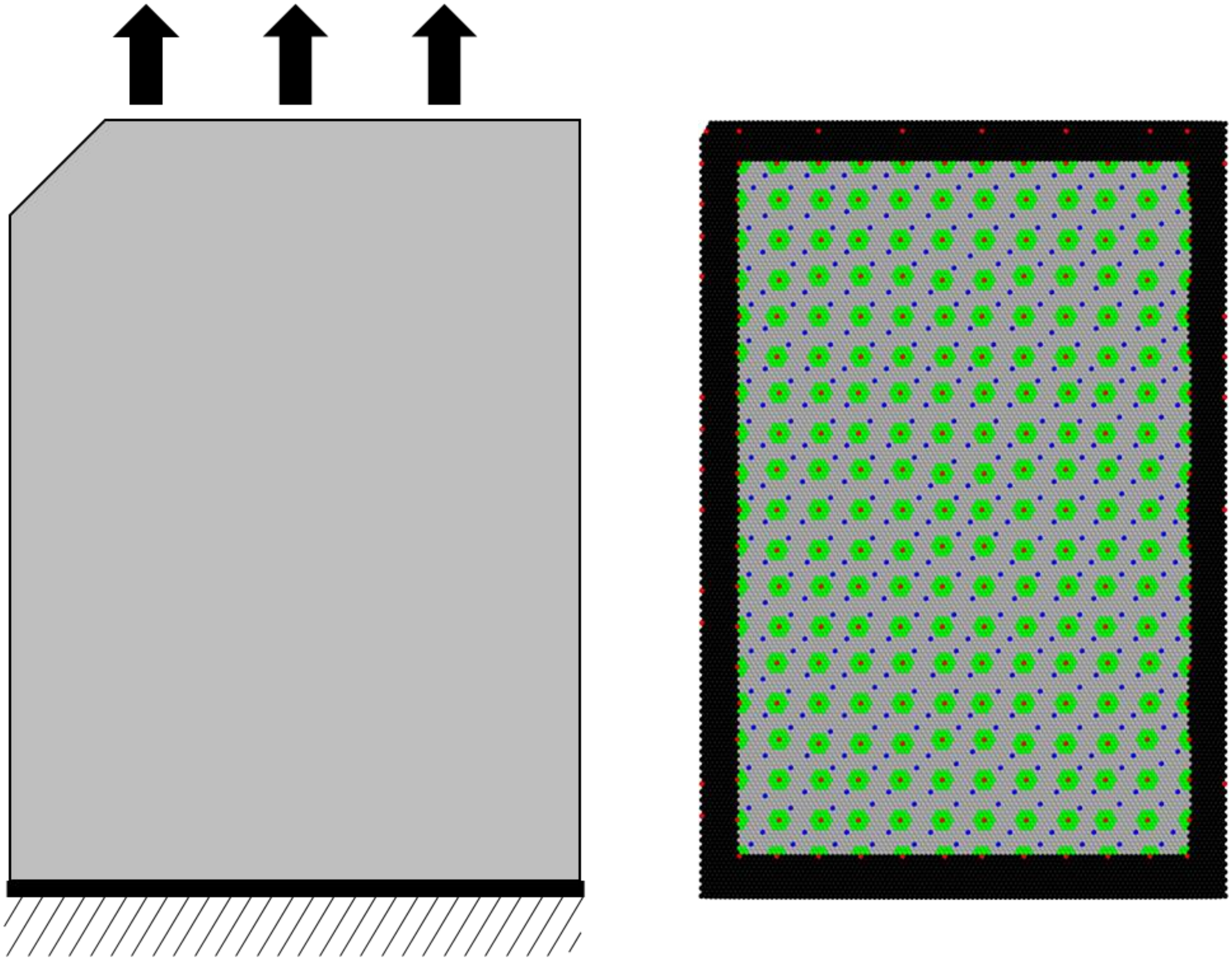

Figure 21. 2-D dislocation example setup (left): Bottom is fixed and top is pulled; and, MMD model atom types (right).

The simulations are run for 1.2 ns at a timestep of 0.5 fs. Two layers of atoms from top of the system are pulled apart at a constant speed corresponding to 10^{-8} s^{-1} strain rate. As the adaptive MMD method adds and removes rep atoms, the overall ratio of rep atoms fluctuated around 27±3%. Figure 22 shows trajectories of dislocation glide for full atomistics and MMD where only the atoms with higher energies are highlighted in the left and center panels. The snapshots are taken from different timesteps of full atomistic and MMD to emphasize the reaction of models to dislocation glide. The latency between the two models is apparent from the stress-strain curve as shows in Figure 23. The force in the figure is calculated by averaging the *y*-components of forces of the fixed atoms, while the strain is obtained from averaging *y*-components of the distance between fixed atoms. Until the yield point, where the dislocation nucleates, stress-strain curves match perfectly, which is attributed to the uniformity of the elastic deformation. Later on, two differences are observed: (i) nucleation of dislocation is predicted earlier by the MMD model and (ii) the burst in the stress-strain curve of the full atomistic model is recorded stronger. These differences are attributed to the constraining effects of the limited size of the atomistic region [15]. Nevertheless, the burst in the stress-strain curve can be further explained with the fact that, although the deformation mechanisms (i.e., perfect edge dislocation) of the two models are the same, energies of dislocations are different. The energy of a dislocation is sum of its nearby core energy and its far field elastic energy where the former is a small fraction of the latter [87]. The elastic energy decays only by $1/r$ where r is the distance from the dislocation core. Therefore, it may be the case that the long range distribution of the elastic energy is not completely covered by the confined atomistic region of the MMD model around the dislocation core. In addition, the difference in the energy of dislocations can be inferred from the stress-strain curves since the stronger burst is related to a higher decrease in the strain energy of the system. In agreement, a larger number of atoms are highlighted in Figure 22 since more atoms with higher energies are involved in the dislocation of the full atomistic model. After annihilation of the dislocations, both the full atomistic and MMD models continue to harden in a similar fashion. Overall, the trajectories and stress-strain curves agree well and the adaptivity successfully tracks and captures the dislocation glide.

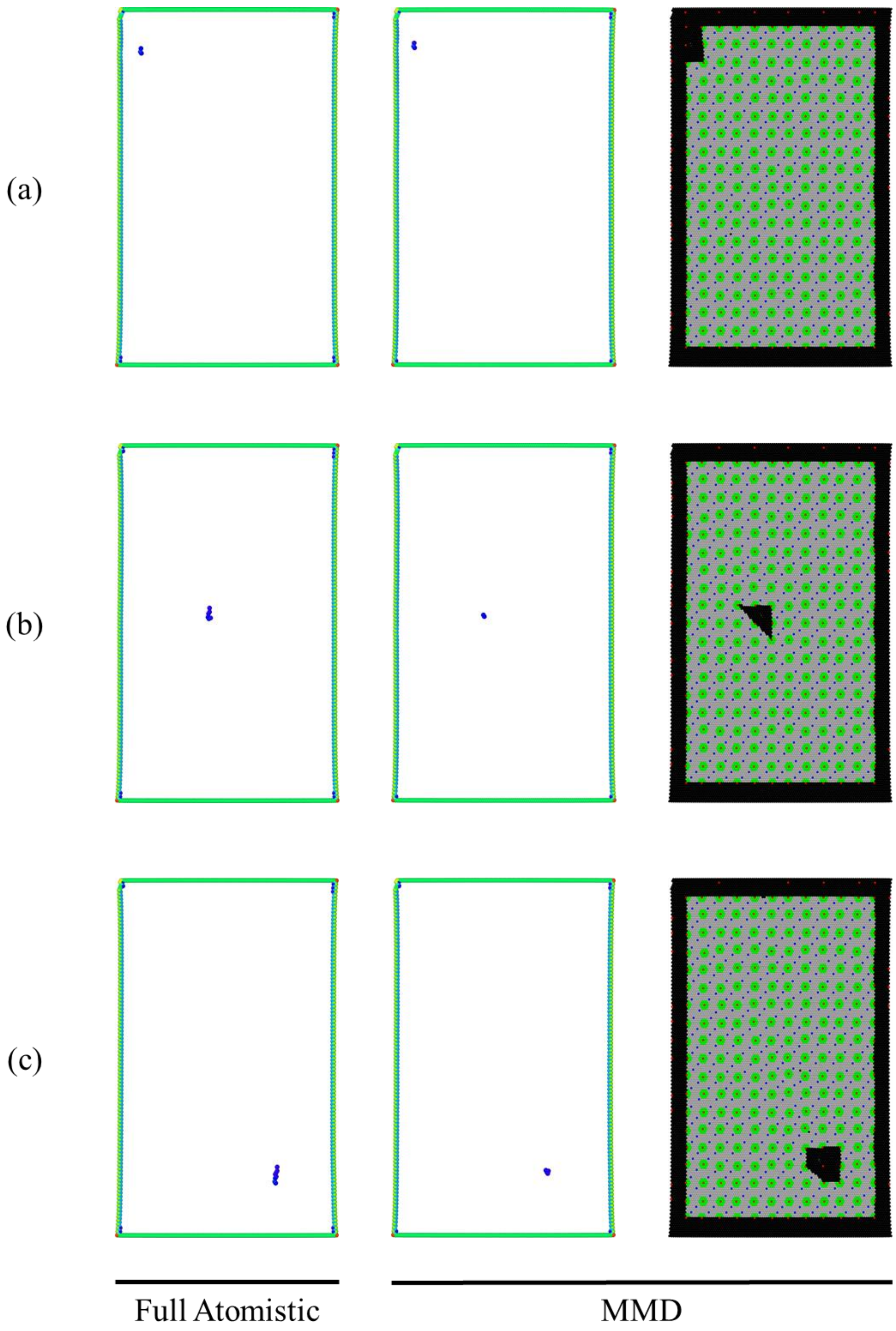

Figure 22. Comparison of trajectories of full atomistic and MMD model at the beginning (a), intermediate (b) and end (c) of the 2-D dislocation example.

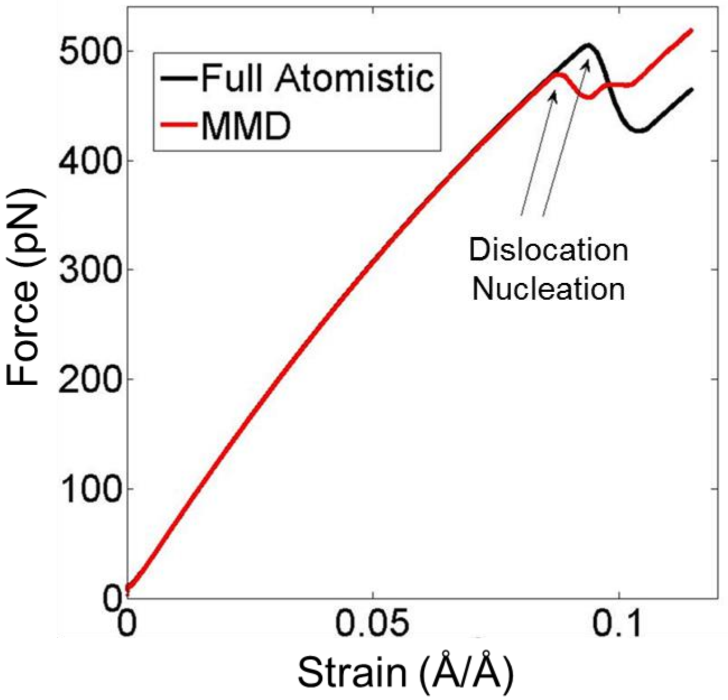

Figure 23. Comparison of force versus strain curves of full atomistic and MMD in the 2-D dislocation example.

3.3.3 3-D Nanoindentation

The third numerical example is a 3-D nanoindentation that is simulated to demonstrate the capability of the adaptivity to capture large amount of dislocations and stacking faults nested together. The model is consisted of a rectangular prism of 78,033 atoms in FCC configuration. The atoms are initially relaxed by static energy minimization. The models are fixed by two layers of atoms at bottom ends in all directions and they are fixed by two layers of atoms at the lateral faces in normal directions (Figure 24 - left). The MMD model is coarsened by tetrahedral elements of size $12r_0$, which consists of secondary sampling atoms (Figure 24 - right). The interactions are modeled with the Morse potential with parameters D_0 = 0.2703 eV, α = 1.1646

1/Å, and r_0 = 3.253 Å and mass is set to 26.9815 g/mole. The adaptivity criteria are set to -2.47 (97% of the initial energy value) and -2.50 eV (98% of the initial energy value) potential energy threshold, 2 Å radius of influence, and 100 iterations of update frequency. Temperature of the system is kept constant at 1 K by the Berendsen thermostat. The simulations are run for 300 ps at a timestep of 0.5 fs. An indenter is pushed towards the top surface with respect to Equation (34).

$$F(r) = -K(r-R)^3 \tag{34}$$

where K is 10 eV/Å^2, R is 80 Å, and r is the distance between the atom and the indenter. The indenter is pushed constantly at a speed of 50 Å/ns.

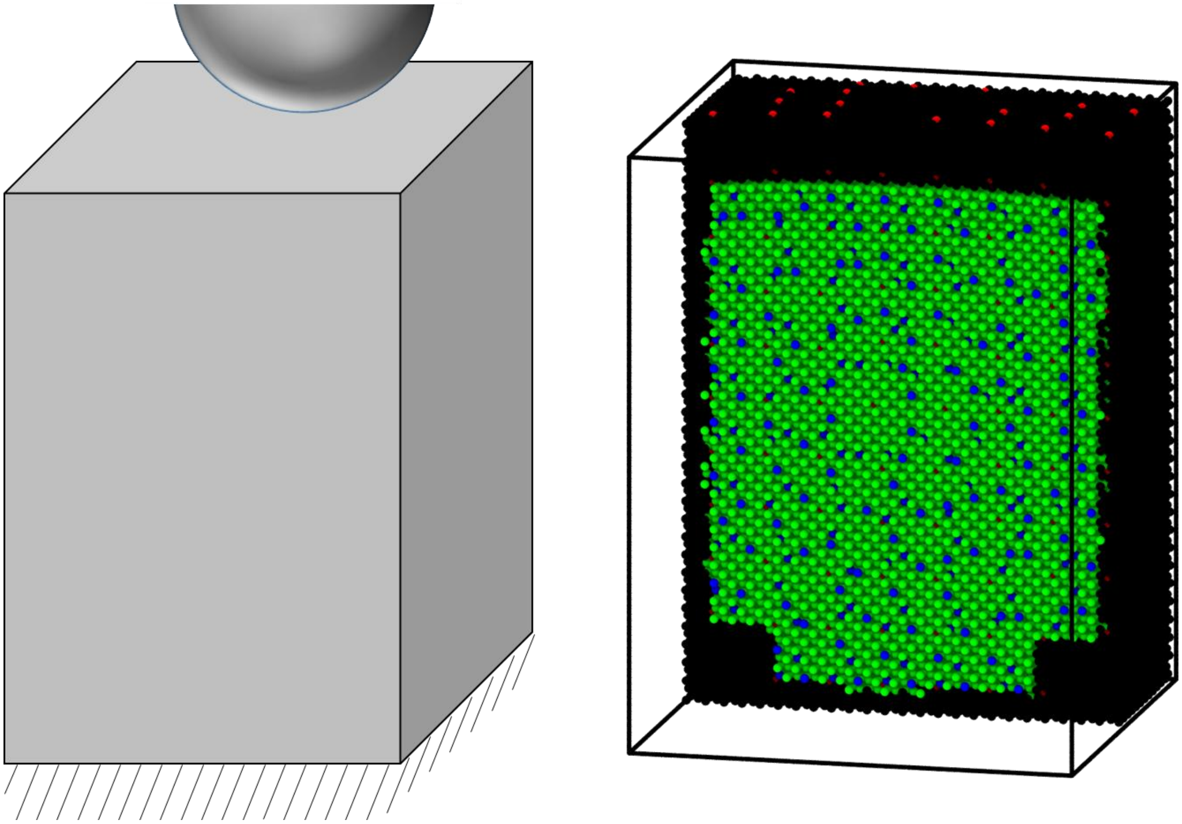

Figure 24. 3-D nanoindentation example setup (left): Bottom and lateral are fixed and top is indented; and, MMD model atom types (right).

As the adaptivity adds and removes rep atoms, the overall ratio of rep atoms reaches 54% and 75% for MMD models with criterion -2.47 eV and -2.59 eV, respectively. Figure 25 shows trajectories at the beginning of plastic deformation for full atomistic and MMD. The similarity of the deformation paths proves the success of MMD models. Figure 26 shows distorted atoms at a later stage of plastic deformation for full atomistic and MMD models. The atoms are colored with respect to deformation types utilizing centrosymmetry parameter [88]. Since it is difficult to distinguish the individual defects and decide if MMD models are successful, a statistical comparison is shown by means of distribution of the centrosymmetry parameter in Figure 27. The agreement between the statistical distributions suggests that the MMD model captures various types of defects well. In addition, the force-depth curves are shown in Figure 28. The forces are calculated from the total forces on the indenter and depth is calculated from the displacement of the indenter. The curve obtained from the MMD model with criterion -2.50 eV performs better and closely captures the burst of the curve.

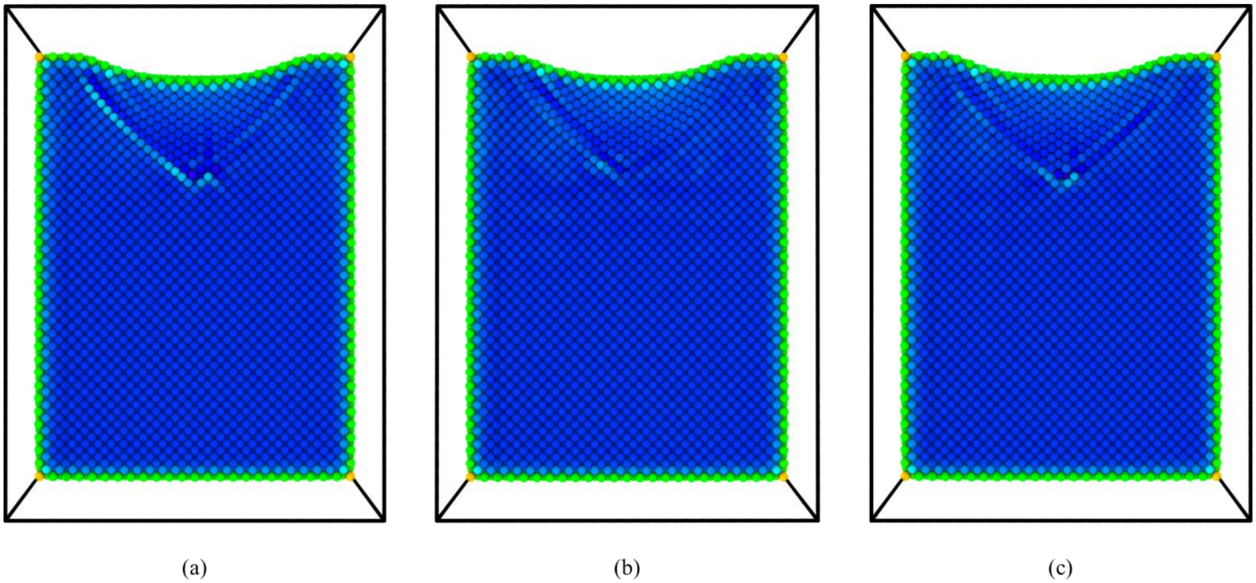

Figure 25. Comparison of configurations at the onset of plastic deformation in the 3-D nanoindentation example where atoms are colored with respect to their potential energies: Full atomistic (a), MMD with criterion -2.47 eV (b), and MMD with criterion -2.50 eV (c).

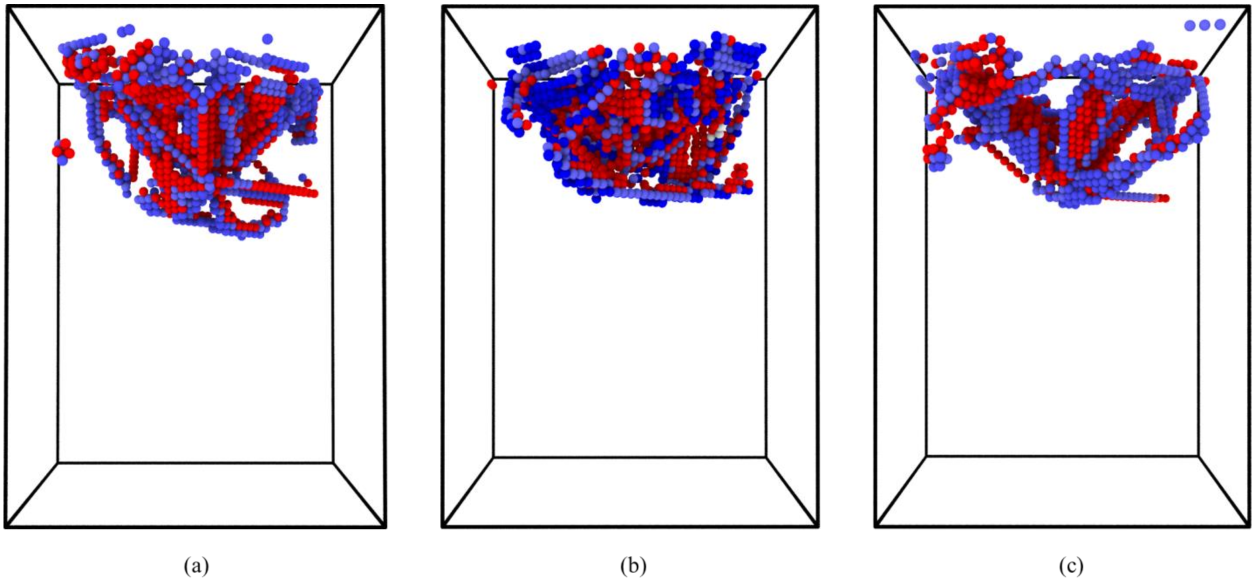

Figure 26. Comparison of configurations at a later stage of plastic deformation in the 3-D nanoindentation example where blue atoms denote dislocation and red atoms denote stacking faults: Full atomistic (a), MMD with criterion -2.47 eV (b), and MMD with criterion -2.50 eV (c).

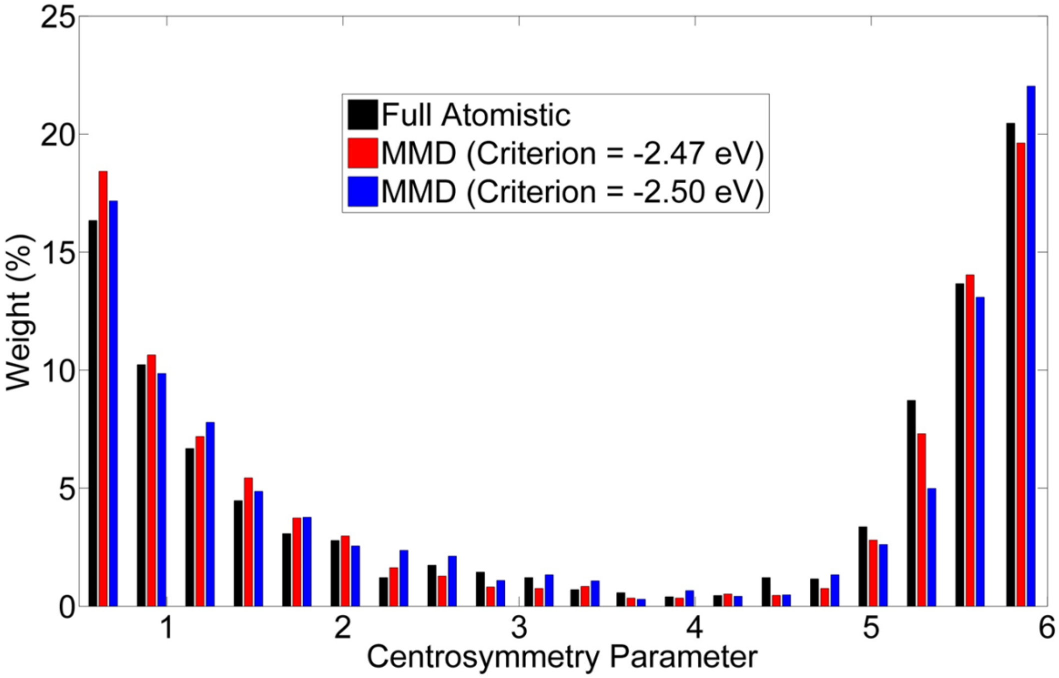

Figure 27. Comparison of centrosymmetry parameter histogram of full atomistic and MMD models with criterion -2.47 eV and -2.50 eV in the 3-D nanoindentation example.

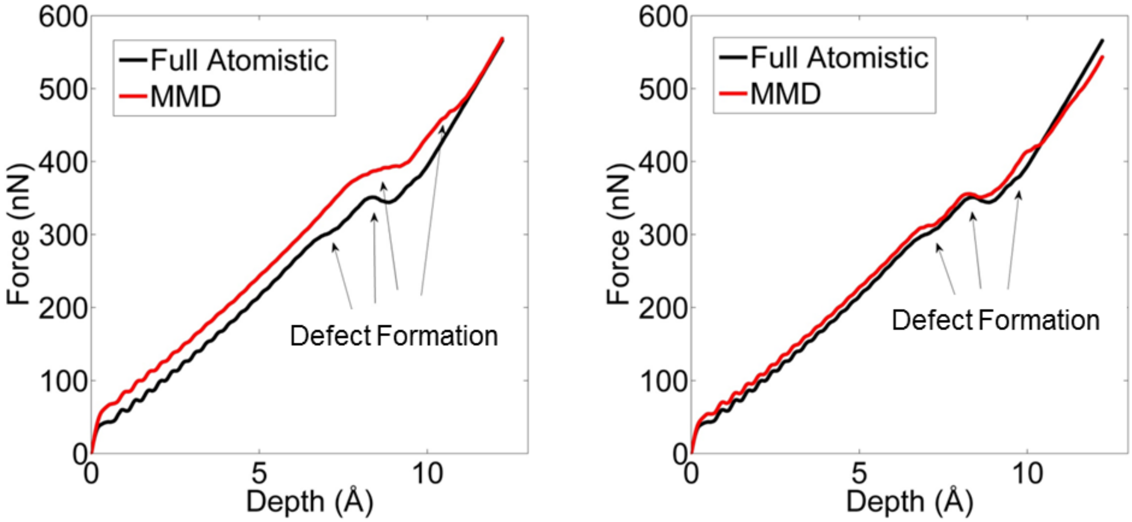

Figure 28. Comparison of force versus depth curves of full atomistic and MMD models with criterion -2.47 eV (left) and -2.50 eV (right) in the 3-D nanoindentation example.

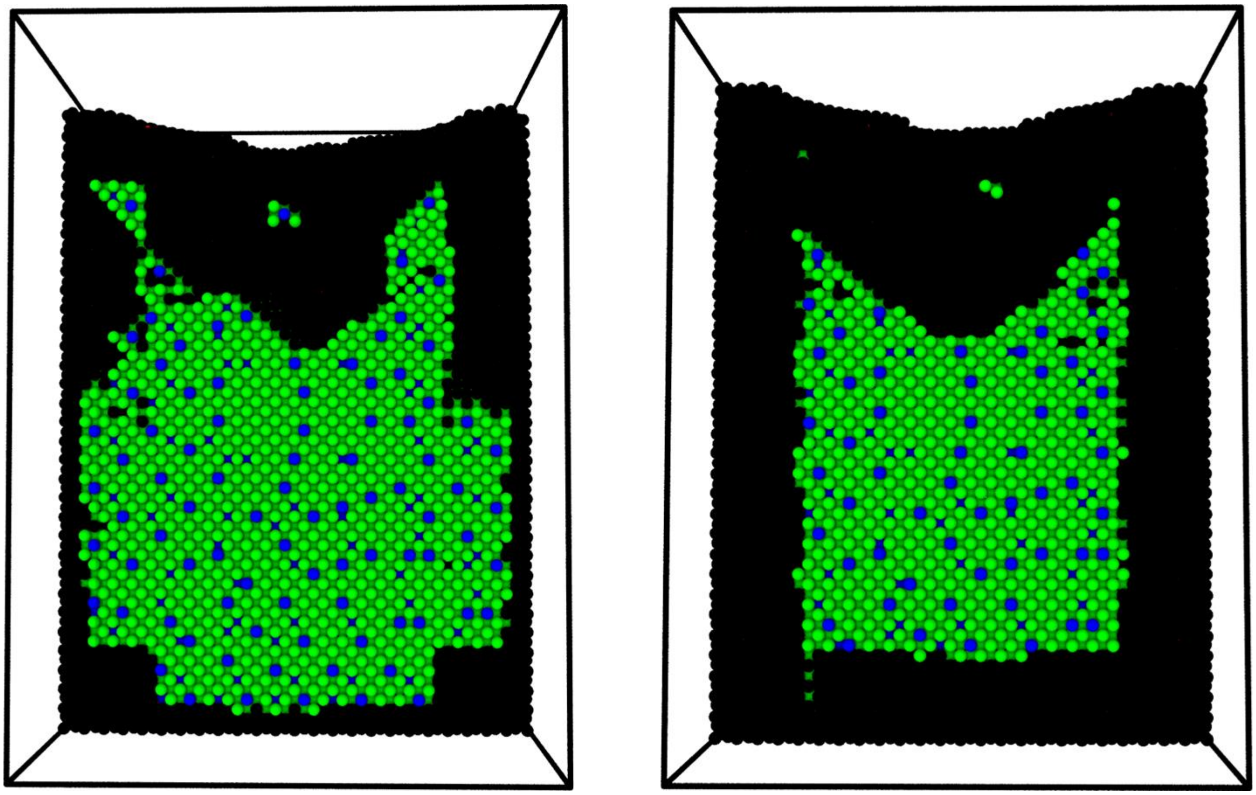

Figure 29. Comparison of atom types of MMD models with criterion -2.47 eV (left) and -2.50 eV (right) in the 3-D nanoindentation example.

3.3.4 Discussion on Adaptivity Criteria

The values of the adaptivity criteria are adjusted based on a trade-off between accuracy and efficiency. There are many settings in an MMM simulation that are subjected to this trade-off. For instance, the solutions by the MMD models with different adaptivity criteria came out different in the 3-D nanoindentation example (Figure 28). As expected, the MMD model with a more stringent energy criterion (Figure 28 - right) uses more rep atoms and the results are closer to the full atomistic solution. In contrast, the MMD model with a less stringent energy criterion

(Figure 28 - left) uses fewer rep atoms hence the efficiency is higher. Another instance is due to different MMM schemes. The scheme employed in the 3-D nanoindentation example is more expensive and more accurate as compared to the scheme employed in the 2-D dislocation example where the former scheme uses all secondary sampling atoms inside the elements. As a result, the results of the 3-D nanoindentation (especially with the more stringent energy criterion) agree better with full atomistic as can be seen from the force-depth curve (Figure 28) and stress-strain curve (Figure 23). Other settings subject to the trade-off between accuracy and efficiency include mesh size and influence of radius. All these trade-offs provide the means to adjust the adaptivity criteria to the preference of the user. As discussed at the beginning of this Chapter, these adjustments are ad-hoc in the sense that they are not based on rigorous theoretical estimations. Nevertheless, the author has experienced little difficulty in tuning the settings for the presented examples and the results agree well with the true solutions.

The physical meaning of the first adaptivity criterion (i.e., energy) is associated with the distortion of an atom from its rest position. At its rest position, an interior atom has the minimum potential energy it can have under the governing potential. At this state, the atom can be thought of lying at the bottom of the well of the curve of the potential energy function. In case of distortion, the potential energy of the atom increases and it elevates from the bottom of the well to a higher level regardless of the type of distortion. Since the goal of the adaptivity criterion is to detect distorted atoms, any atom with a relatively higher potential energy is a good indicator of distortion. There is no physical correspondence to the level of increase in the potential energy. However, there are physical correspondences to the level of increase in the centrosymmetry parameter. Particular ranges of the centrosymmetry parameters are known to be indicators of particular defects such as dislocation or stacking fault as shown in the 3-D nanoindentation example. In the current work, potential energy is utilized in the adaptivity scheme and centrosymmetry parameter in the post-processing tools. However, the MMM software has the groundwork for centrosymmetry parameter to be implemented as an adaptivity criterion in case the user prefers to distinguish between different types of defects during the simulation.

The results of the MMM simulations are qualitatively and quantitatively compared to the true solutions in the presented numerical examples of the current work. The success of the approximate solution of MMM in matching the true solutions is evaluated with respect to capturing certain material properties such as elastic modulus, yield stress/strain, natural frequency, and hardening behavior. However, definition of a criterion that measures the acceptance of the approximate solution is avoided in order to let the user decide about the level of accuracy. As discussed earlier, the MMM method features a trade-off between accuracy and efficiency by many settings. Thus, the user has several options to tune for the right balance between accuracy and efficiency with respect to what s/he demands from the problem. In one simulation, the user may want to quickly observe the general behavior of a deformation where s/he should adjust the settings for higher efficiency. In another simulation, the user may want to investigate the fine details of a deformation where s/he should adjust the settings for higher accuracy. In this context, the level of accuracy is highly relative and depends on the demands of the user and the subject problem. Consequently, the user has the power to adjust for the level of accuracy s/he demands with respect to the time and computational power s/he desires to spend on the problem.

4.0 IMPLEMENTATION

Molecular mechanics is addressed in its own way from a computational standpoint. Molecular mechanics prioritizes proper and efficient sampling of phase space rather than high accuracy, which can be well achieved by symplectic integrators [89]. Phase space can be explored stochastically, such as with conjugate gradient method [90], or deterministically, such as with MD [91], or with special relaxation algorithms [92]. Our focus is on the deterministic approach with MD, which solves Newton's equations of motion (2nd law) given by Equation (35).

$$m_i \frac{d\mathbf{v}_i}{dt} = \sum_j F_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_j \sum_k F_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \cdots$$
$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i \qquad (35)$$

Usually the terms beyond a few are excluded and their effects along with quantum effects are represented by the remaining terms [93]. The force term in Equation (35) is equal to the derivative of the potential energy with respect to the position of the atom. The energy of an MD system is given by Equation (36)

$$E = E_{bonded} + E_{electrostatic} + E_{van\ der\ Waalls} \qquad (36)$$

The last term in Equation (36) represents non-bonded interactions that are consisted of short and long range interactions [94] Usually, short range non-bonded interactions are taken into account and long range non-bonded interactions are either approximated or neglected due to their high cost. The strength of short range forces decay fast so that interactions beyond a certain distance (r_c) are truncated, or “cut-off”. Owing to this reduction, force calculations scale with $Nr_c^3\rho$ instead of N^2 where N is the number of atoms and ρ is the number of atoms per unit volume. In some cases, the effects of the cut-off region are compensated by some sort of correction.

The equation of motions are numerically integrated by the Velocity-Verlet scheme, which is symplectic, time-reversible, conserves linear and angular momentum and requires one force evaluation per timestep; and its error is proportional to Δt^2 [89]. Despite cut-off and efficient integration, MD is fine in length and time scales, and hence highly demanding [93]. The calculation of force takes about 80-95% (and even higher in the current work) of the simulation time and other operations takes about 5-20% [85]. Other operations include, for instance, building in neighbor lists, input/output, and integration, where the last operation takes about 2-3% of simulation time [93]. Memory is mostly consumed by neighbor lists [95]. Newton’s 3rd law may be exploited to halve the costs of calculation of interactions and memory consumption of neighbor lists. In parallel computing, this utilization is obviously useful in atom decomposition scheme. In a domain decomposition scheme, however, it is a matter of trade-off between costs of computation and communication due to the additional costs of the latter. The decision is usually reached with respect to communication cost of the subject algorithm. The advantages and disadvantages of this trade-off are detailed by Plimpton [93].

Searching for nearest neighbors in every iteration can be a bottleneck if it is not handled by an efficient algorithm. Plimpton introduces neighbor lists, link-cell, and a combination of the two [93]. Neighbor lists store the list of neighbors beyond the cut-off distance so that only a limited number of atoms (instead of all atoms) are looped to find the nearest neighbors in every iteration. Link-cell method utilizes binning algorithm that efficiently bins atoms into cubic cells and then searches for nearest neighbors in neighbor cells. There are 8 neighbor squares in 2-D and 26 neighbor cubes in 3-D. Plimpton efficiently combines the two methods by utilizing link-cells to build in the neighbor lists [93].

MD software is highly suitable for parallelization. Parallelization is mainly motivated by the fact that hardware technology has reached saturation in processor performance. In addition, the demanding nature of MD simulations strives for parallelization. However, the parallelization technology is considered to be immature in its current state [89] and requires efficient algorithms in order to show scalability [85]. Parallelization can be realized by data parallel (a.k.a. memory coupling) or message passing method (a.k.a. message coupling) [95]. Data parallel method instructs the compiler some arrays that work in parallel as in OpenMP and suitable to shared memory environments. Message passing method manages communication of messages explicitly as in MPI and suitable to distributed memory environments. Although message passing is more manual in terms of implementation, it can achieve higher performance by proper optimization, even for low number of processors [94]. Also, data parallel method is slower even if it is coupled with many different algorithms to solve conflicts due to race condition [91]. In some cases, data parallel and message passing methods are brought together such that the former is employed within computing nodes and the latter is employed among computing nodes.

There are three decomposition schemes in parallel computing MD simulations: atom, force, and domain decompositions. Atom decomposition (a.k.a. replicated data) assigns atoms to processors. This scheme has the advantage of easy implementation and geometric independency. This way, load balancing is trivial, e.g., can be realized by random permutation. Every processor has all the information thus memory consumption is not efficient and communication is all-to-all thus it may be inefficient. The scalability is weak due to high volume of communication for large systems but noted to be working well for tens of processors [85]. The communication scales by *O(N)*, therefore increasing the number of processors does not decrease the cost of communication [93]. Force decomposition assigns atoms to processors according to a permutation of the force matrix in order to cut the communication cost. As in the atom decomposition scheme, there is geometric independency enabling trivial load balancing. However, the communication cost scales better with $O(N/\sqrt{P})$, which can be decreased by increasing the number of processors. Domain decomposition (a.k.a. spatial decomposition) assigns each processor a subdomain of the simulation domain, hence different from the first two particle decomposition methods [95]. This scheme is difficult to implement (e.g., requires atom migration), it is geometrically dependent, and it suffers from load balancing for heterogeneous systems (e.g., multiscale models) and irregular shapes (there are helpful algorithms that organizes partitioning though [91]). In spite of its disadvantages, domain decomposition scheme is very efficient and communicates locally. Three schemes are compared in terms of their theoretical costs and simulation times in detail by Plimpton [93]. In another scheme, Shaw et al. developed the midpoint scheme for efficient decomposition of atom interactions in that particles interact at the box where their midpoint lies; where, the efficiency is achieved by a lower volume of communication compared to traditional spatial decomposition algorithms [94].

In parallel computing, proper load balancing is crucial in order to utilize processors efficiently. As noted earlier, load balancing is trivial with atom and force decomposition schemes such that atoms can be migrated at any time by assigning them to other processors. Load balancing is more difficult with domain decomposition schemes due to extra costs of migrating atoms. In this connection, heterogeneous and/or moving systems are challenging. Dynamic load balancing may be useful if its profits outweigh its overheads. Dynamic load balancing is realized in a measurement-based fashion in NAMD where the simulation is run for a short time for a few times and then load is balanced accordingly [85].

There are a few limited efforts to utilize High Performance Computing (HPC) in multiscale methods. Xiao et al. presented task and data decomposition in atomistic, continuum, and bridging domains of the BD method; however, their demonstration is limited to one-dimensional wave propagation [84]. Anciaux et al. presented parallel implementation of BD on a 2-D crack example, which showed load imbalance due to coupling overheads [96]. Fox et al. presented parallelization of their multiscale method using MPI and demonstrated up to 3.12 times speed up in simulation time [97]. In addition, there are initiatives to build computational libraries [98] and software infrastructure [99] for multiscale modeling and simulation.

In the following, MMM software, its test functions, and its efficient improvements are introduced, in order. Then, the performance of MMM software is compared with LAMMPS, and it is also evaluated as a function of rep atom ratio on a single processor. Finally, parallelization of MMM software is discussed and the overall efficiency of the software is evaluated by a 2-D nanoindentation numerical example.

4.1 MMM SOFTWARE

This section introduces specifications and features of the MMM software and can be regarded as a short manual. The details of implementation are further elaborated by comments in the source code. Considering the readability and modularity of the code, the user can easily understand, modify, extend, and optimize the software in part or entirely.

MMM is a homegrown code written in C++11 in an object-oriented fashion. The software is composed of a hierarchy of classes with the top level class, named *MMM* as listed in Table 5. The entire software is packed into top level *MMM* class; in this way, it can be conveniently coupled to other software.

Table 5. Hierarchy of classes in the MMM software.

Top Level	Middle Level	Visibility	Bottom Level
MMM	Adaptivity	(VV)	Mediator
	AddIn	(VV)	
	AtomGroup	(V)	
	ConjugateGradient	(VV)	
	Input	(VV)	
	Mesh	(V)	
	Model	(VV)	
	Neighbor	(V)	
	Output	(VV)	Matrix
	Parallel	(VV)	
	Potential	(VV)	
	Select	-	
	Temperature	(VV)	
	Time	(VV)	
	VelocityVerlet	(VV)	
BuildInitialConfiguration			
Test			
Utility			

MMM class has many subclasses, which are listed under middle level and bottom level tabs in Table 5. The subclasses tagged with *(VV)* have access to each other by means of holding a pointer to each other. This *N-to-N* relationship is easily built by inheriting these subclasses from the *Mediator* subclass, which establishes the entire underlying pointer structure. The subclasses tagged with *(V)* are designed to be independent and thus they do not belong to the previous group and do not have access to the *(VV)* subclasses. In contrast, *(VV)* subclasses have access to the *(V)* subclasses. Independency of a class or function indicates that the class is standalone and does not implicitly depend on any other part of the program. Subclasses without any tag are independent and they do not have access to other subclasses but *MMM* class owns a copy of them. In this regard, *BuildInitialConfiguration, Test*, and *Utility* are collections of independent functions. *Matrix* is a sublevel class that introduces 1-D and 2-D data structures designed for the convenience of the MMM software. *AddIn* is a subclass suggested for user-designed add-in functions, e.g., calculation and output of stress-strain values. Polymorphism is utilized in *Potential* class such that all potentials are derived from the *PairBase* base class. This way, a single class name (i.e., *pair_potential_*) masks the actual potential hence providing a convenient interface. The remaining classes are designed for the tasks according to their names. Detailed description of the various classes and functions can be found in their declarations and definitions.

The MMM software has many homegrown features regarding implementation and modeling. Exceptionally, the only external dependency is the Qhull meshing software [100]. The executable of the software is called by system commands to generate 2-/3-D Delaunay mesh. In a nutshell, some implementation features of the MMM software include:

- Parallelization with MPI (MPICH 3.1.3)

- Fast kd-tree nearest neighbor search (performs worse than the binning algorithm but the difference is negligibly small compared to the simulation times)
- Benchmarked against LAMMPS (15 May 2015) (details of the comparison with LAMMPS will be presented later)
- Tested against LAMMPS and against problems of the literature (details of the tests will be presented later)
- Profiled by Visual Studio (2013) and GNU gprof (2.21) for maximum efficiency
- Styled with Google C++ Style Guide for consistency and better readability: also, many design decisions are made according to the rationale provided in this document
- Portable: single core/multiple cores, Windows/Unix, Visual Studio/Makefile
- Common I/O format (e.g., LAMMPS trajectory output)
- User-friendly I/O formats: minimum number of abbreviations and minimal technical language
- Concrete implementations are interfaced in the header files with accessor/mutator functions for readability and modularity: in this way, an implementation can be easily modified on the background with minimal effect on the other parts of the code
- Source code is available online

Some modeling features of the software include:

- Both full atomistic and MMM on the same code: common MMM schemes are available
- 1/2/3 dimensions
- Solvers: conjugate gradient with line search for statics and Velocity-Verlet integrator for dynamics
- Potentials: spring, LJ, Morse

- Thermostats: velocity rescaling, Berendsen, Langevin
- Displacement and force boundary conditions with set/add options
- Various atom group selection schemes (surface, radial, grid, block, ID, file, and more)
- Qhull (2012.1) meshing (2-/3-D Delaunay)
- Linked to PETSc (3.4.3) and SLEPc (3.4.3) for parallel matrix operations and eigen-analysis (not included in the distributed version)
- Various pre-/post-processing tools (manual meshing, plotting stress-strain, etc.)
- Add-in that allows users to integrate their own code (this is different than scripting since a new add-in requires recompilation of the source code)

The simulation is performed by feeding the input file into the *MMM* class in the main function. In this way, a simulation can be easily setup from its input file and several setups of simulations can be readily saved. The format of the input file should be understood from command functions in *mmm.cc* source file. Commands are executed in the order they are listed in the input file. A sample input file is shown in Figure 30.

```
# simulation perf 130 r240
dimension 2
initial_configuration 26.981538 file m62.txt
neighbor 11.5 auto
potential morse 10.7844 0.49140659 1.457364 2.58
mesh file m62_mesh_spacing_120.txt
model mmm no_ssmp
select 1 surface x+ 4.0
select 2 surface x- 3.2
select 3 unite 1 2
select 4 surface y- 4.0
load displacement 3 set 0 NULL NULL
load displacement 4 set 0 0 NULL
output 1000 100
temperature berendsen 1.0 1.0 0.05
# indent2d inputs: x y del_y K R
add_in indent2d 1340.607325 2118.96755 0.0001 10.0 40.0
adaptivity 0 true true 1000 energy -1.67 -1.47 240
#run velocity_verlet 0.05 1000
run velocity_verlet 0.05 350000
```

Figure 30. A sample MMM software input file.

The commands starting with a # character are treated as comments and ignored. Since the commands are executed in the order they are read, it makes sense to start from more fundamental components of a simulation and build up the complexity. The order of input commands does not matter except the ones depending on each other. The dependency of commands should be obvious. For instance, the sample input file starts with the *dimension* command and then reads the initial configuration from a file. Building in neighbor lists should be after the generation of initial configuration. Likewise, the *mesh* command is put before the *model* command since building the model needs a mesh. The *run* command comes after all settings are input and iterates the system with respect to the input settings. A set of sample input files can be found under the *test/input* directory.

Each simulation outputs a log file and at the beginning of it the input file is pasted for the record. The log file provides detailed information after each command is executed. During the *run*, it provides information (e.g., potential energy) periodically in an input frequency. At the end of the *run*, many useful information (e.g., simulation time) are provided as well. Again, a set of sample output files can be found under the *test/true* and *test/true4* directories.

The MMM software consists of approximately 10,000 lines of code in 41 source (*.cc*) and header (*.h*) files under the *source* directory. Other directories are *input* to read input files, *output* to write output files, *object* to keep object files, *test* to keep test input file and true solutions, *visual_studio* to keep files related to Visual Studio, and *document* to keep *copyright*, *manual*, *readme* and *todo* files. In addition, there are many pre-/post-processing tools (mostly written in MATLAB) that amount to 5,000 lines of code. It is important to note that the source code is efficiently written in a concise and clear fashion. Although the software is developed by Visual Studio 2013 IDE in Windows, a makefile is used to build the software in UNIX environment. All simulations presented in this work are particularly compiled with GNU GCC 4.9.1 C++ compiler wrapped by MPICH 3.1.3 and run on a Dell cluster having 64x2.26GHz Intel processors with 24MB cache size and 189GB memory size on an openSUSE 11.4 operating system.

4.2 TESTING

The MMM software is accompanied with a set of test functions to verify the product. By this way, the users and developers can modify the code and then quickly verify that the software is functioning properly and no mistake has been done. There are 16 test simulations (grouped into 8 test cases) that check almost all parts of the code including 1/2/3 dimensions, static and dynamic solvers, MMM schemes, input/output, meshing, potentials, selection schemes, thermostats, boundary conditions, adaptivity, conservation of linear momentum, parallelization, and others. The tests are run for both single and four processors. The test simulations are verified by the output of selected variables that includes IDs, types, positions, energies, and forces of atoms at the beginning and end of the simulation. Input files of test simulations can be found under *test/input* directory. These test simulations also serve as samples to the MMM software. True solutions to single processor and four processor simulations can be found under *test/true* and *test/true4*. The MMM software can be tested by inputting *test* keyword as the simulation name. Sample output of running tests is shown in Figure 31.

```
Running test 1
----------------------------------------
All output: PASS!
It took 0.04 seconds.

Running test 2
----------------------------------------
All output: PASS!
It took 1.09 seconds.

Running test 3
----------------------------------------
All output: PASS!
All output: PASS!
All output: PASS!
All output: PASS!
All output: PASS!
It took 0.72 seconds.

…

Running test 8
----------------------------------------
All output: PASS!
It took 0.03 seconds.

If the program reached to this point, it means all tests are PASSED!
All tests took 5.47 seconds.
```

Figure 31. Sample output of test of the MMM software.

4.3 EFFICIENCY IMPROVEMENTS

In the following, a number of improvements that increase the efficiency of the MMM software are introduced. These improvements include latent ghost atoms, dynamically balancing the load on processors, and other modifications.

4.3.1 Latent Ghost Atoms

As the theory of MMM dictates, non-interpolating atoms do not interact (apply force) with each other. Non-interpolating atoms interact only with other types of atoms if there are any in their range. As a result, non-interpolating atoms having all non-interpolating atom neighbors do not have any interactions, hence inactive. However, these inactive atoms are redundantly included in the computations: (i) they are visited in loops of energy and force calculation, (ii) their positions are interpolated, and (iii) their forces, which are actually equal to zero, are extrapolated. In order to eliminate these redundant computations, inactive atoms are converted to latent atoms, which means that they are excluded from the aforementioned computations (the word “sleep” is used instead of “latent” in the MMM software). Figure 32 shows a 2-D square copper system consisted of 1,904 atoms in hexagonal configuration where interactions are truncated beyond the second shell of neighbors. The atoms that are not latent are neighbors of interpolating rep atoms and primary sampling atoms. The system consists of 1,671 latent atoms, which corresponds to 88% of all atoms.

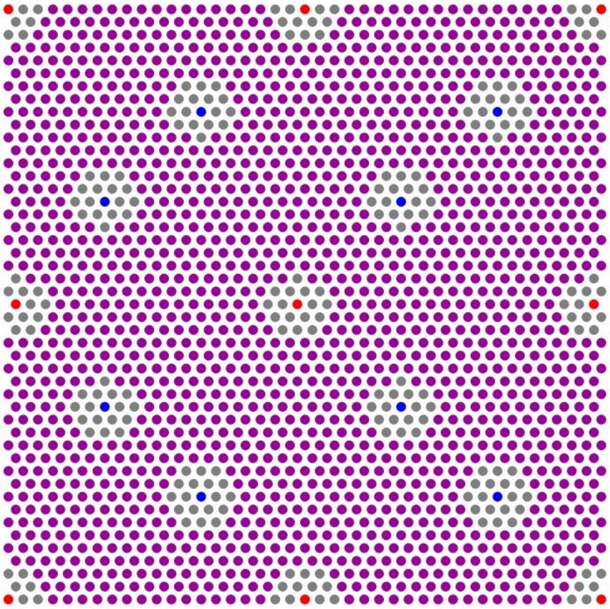

Figure 32. 2-D square system with latent atoms (that correspond to 88% of total atoms) shown in purple color.

The atoms are converted to latent atoms whenever they are not needed. The atoms are needed when it is time to output to file, adaptivity is performed, or neighbor lists are updated. When latent atoms are needed, they are activated, which means that their positions are updated by the displacement of interpolating rep atoms since the time they were converted to latent. Since only one set of positions of rep atoms are saved for the sake of efficient memory usage, all latent atoms are activated if any latent atom is activated and then atoms that are supposed to continue their latency are converted back to latent atoms.

4.3.2 Dynamic Load Balancing

MMM models are highly heterogeneous since the types of atoms are non-uniformly distributed in space. The cost of computation associated with each type of atom is difficult to anticipate since it also depends on the types of neighbor atoms. In addition, adaptive MMM models may change the types of many atoms suddenly, hence elevating the heterogeneity to an even more unpredictable level. These facts require careful partitioning of atoms onto the processors. As will be presented later, the MMM software employs atom decomposition scheme, which allows easy partitioning of atoms onto the processors without a need for atom migration. That is, an atom can be assigned to any processor at any time without any other additional effort. A simple scheme partitions atoms onto the processors by random permutation. However, this scheme may suffer from unbalanced loading of atom types. For instance, one processor may have rep atoms in majority and another processor may have ghost atoms in majority. It is obvious that the cost of computations associated with rep atoms is higher than that of ghost atoms. In this case, the first processor is overloaded. A second scheme partitions atoms onto the processors by interlacing atoms of the same type. That is, interpolating rep atoms are partitioned onto the processors one by one leaving every processor with almost equal number of interpolating rep atoms, and then non-interpolating rep atoms are partitioned onto the processors one by one, and so on. In fact, this scheme is much more efficient than random permutation. However, it still suffers from unbalanced loading for at least three reasons: (i) a surface atom has fewer number of neighbors compared to an interior atom, (ii) utilizing Newton's 3rd law to compute half of the interactions may reduce the load of atoms at very different rates, and (iii) processors perform differently due to hardware issues, for instance, a cluster may be made up of old and new processors [101].

The MMM software implements measurement-based dynamic load balancing, which is an efficient way to partition atoms onto the processors. In particular, the time spent in parallel regions by each processor is recorded over a period of time. At the moment of dynamic load balancing, processors that recorded parallel times lower or higher than the average are identified.

Then, a certain amount of atoms are collected from processors with higher parallel times. This amount is determined in proportion to the deviation of the processor's parallel time from the average. Collected atoms are then partitioned onto the processors with lower parallel times, again in proportion to the deviation of the processor's parallel time from the average. The frequency of dynamic load balancing is set to 500 iterations since a lower value is found to be too sensitive. The significance of dynamic load balancing is more pronounced with higher number of processors. In order to demonstrate the effect of dynamic load balancing, a 2-D rectangular copper system consisted of 101,537 atoms in hexagonal configuration is simulated. The material is modeled with Morse potential where interactions are truncated beyond the fourth shell of neighbors. The simulations are run on 32 processors with and without dynamic load balancing. Relative atom numbers (Equation (37)) and relative parallel times (Equation (38)) are compared.

$$(Relative\ Atom\ Number)_P = \frac{Number\ of\ atoms\ that\ belong\ to\ P}{Total\ number\ of\ atoms} x100 \tag{37}$$

$$(Relative\ Parallel\ Time)_P = \frac{Parallel\ time\ spent\ by\ P}{Total\ simulation\ time} x100 \tag{38}$$

where *P* is the processor ID.

Results of relative atom number and parallel time distributions are shown in Figure 33.

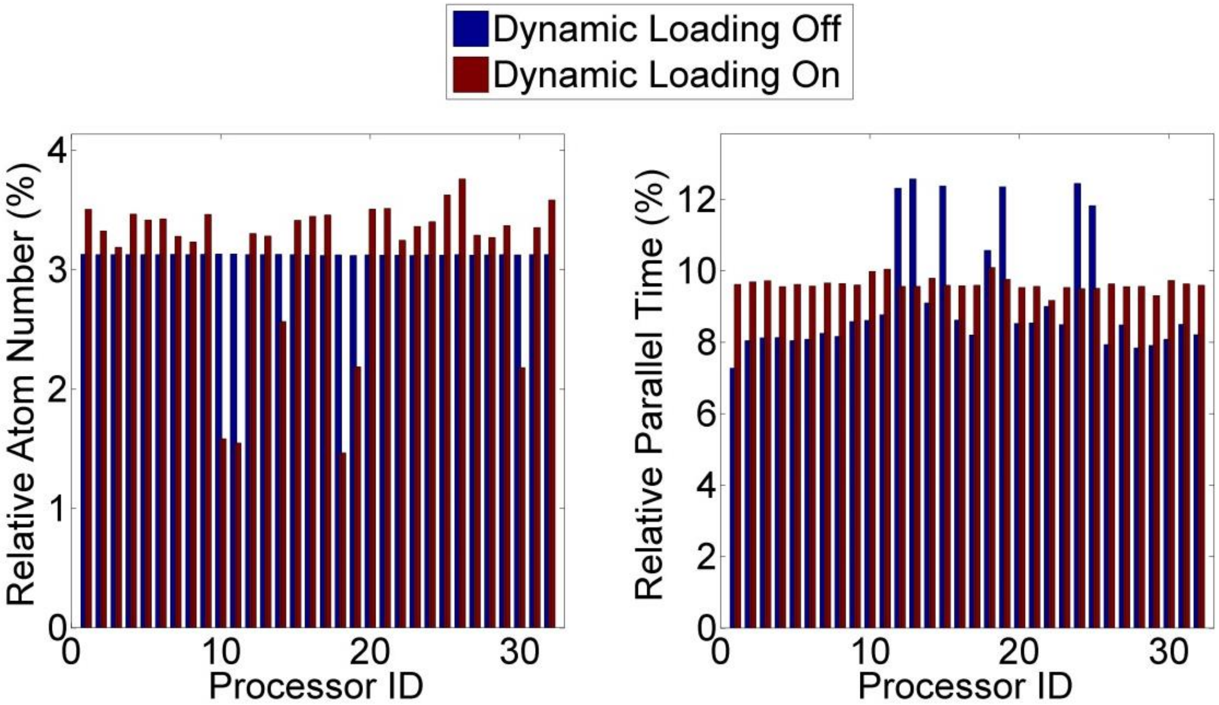

Figure 33. Relative atom number (left) and relative parallel time (right) vs. processor ID with and without dynamic loading.

Figure 33 shows that when the dynamic loading is turned off, atoms are partitioned with respect to the scheme that interlaces atom types. As a result, atoms are partitioned evenly resulting with an uneven distribution of parallel times. Since processors wait for each other in every iteration to exchange information, the iteration time is equal to the maximum parallel time of that iteration. On the contrary, when dynamic loading is turned on, atoms are partitioned unevenly with respect to their parallel times. As a result, parallel times are evenly distributed and the iteration time is lower than that of dynamic loading turned off by 9% (corresponds to 3% in simulation time). This reduction would be much more pronounced in case of an adaptive simulation where types of many atoms change suddenly, hence the workload of a processor. Some additional statistical results are listed in Table 6.

Table 6. Relative standard deviation of relative atom numbers and relative parallel times when dynamic loading is turned on and off.

Dynamic Loading	Relative Standard Deviation of Relative Atom Number	Relative Standard Deviation of Relative Parallel Time
Off	0.1%	18.1%
On	20.0%	1.9%

Table 6 shows that relative standard deviation of relative parallel time significantly decreases when atoms are efficiently partitioned by dynamic loading, which is evident from the increase in relative standard deviation of relative atom number.

4.3.3 Other Improvements

Four other improvements are implemented in the MMM software in order to increase efficiency. The first one is lazy potential energy computation, which is motivated by the fact that the potential energies of atoms are not required to iterate the trajectory. Therefore, energies of atoms are computed only when they are required, e.g., output to file. This reduction increased the efficiency by about 10%. The second improvement is building in interaction lists. Interaction list of an atom consists of a list of neighbor atoms that is reduced by utilizing Newton's 3rd law and atom types. Utilizing Newton's 3rd law, the force between two atoms is calculated once and added to the resultant force of both atoms. In this way, the interactions are cut in half. In the other reduction, the interactions between non-interpolating atoms are excluded from the interaction list. Since interaction lists are the most frequently visited sections of the software, these reductions contributed significantly to the efficiency. Specific to 3-D models, the third improvement requires search of the mesh element where each atom lies in – an atom does not necessarily lie in a mesh element throughout a simulation. Given the nodal coordinates of a tetrahedron and coordinates of an atom, checking if the atom lies inside the tetrahedron consist of computing determinants of five *4x4* matrices, which is computationally expensive. This operation is repeated for *M* by *N* times where *M* is the number of elements and *N* is the number of atoms. Due to high computational cost in question, this section is parallelized. This parallelization is exceptional in the context of the MMM software since the mainly parallelized section is energy/force computation, as will be presented later. The last improvement is periodically writing user-specific outputs to file. In the simulations where stress-strain values per iteration are requested, these values are expected to be written to an additional output file in every iteration. However, access to a file has a fixed cost which may introduce a bottleneck if performed every iteration. Therefore, values of a number of iterations are saved to a buffer and written to file periodically. This way, the cost of writing user-specific inputs to file is significantly reduced.

4.4 COMPARISON WITH LAMMPS

In addition to MMM models, the MMM software is also capable of performing full atomistic MD simulations. This feature provides the basis of benchmarking the software with other similar software. Performance of the software is thus compared with full atomistic models to the well-known Large-scale Atomistic Molecular Massively Parallel Simulator (LAMMPS) software [93]. LAMMPS software is very efficient and stands to be one of the best state-of-the-art MD programs. The comparison is carried out with a 3-D cubic copper system consisted of 9,842 atoms in FCC configuration. The material is modeled with Morse potential where interactions are truncated beyond the fourth shell of neighbors. The simulations are run for 1,000 iterations by LAMMPS and MMM. Measurements are taken as the average of many trials and loop times (the time spent on iterating the system) are compared. As a result, MMM loop time is recorded as 164 seconds while LAMMPS loop time is recorded as 157 seconds. That is, the MMM software is only 4% slower than LAMMPS. This slowness is attributed to: (i) the MMM software writes additional information (ID, type, positions, energies, and forces) to an output file that may account for up to 1% and (ii) the MMM software is a research code, which means that it stands to be improved [26]. In other words, research code implements novel approaches, which require software to be maintainable and extensible [85].

4.5 EFFICIENCY OF MMM ON A SINGLE PROCESSOR

The primary motivation of developing MMM is to run simulations at a cost lower than that of full atomistic. In order to show the capability of MMM in this respect, the efficiency of MMM models are compared to a full atomistic model. This comparison can be extended to full atomistic software in general since full atomistic performance run of MMM is shown to be as good as LAMMPS in the previous section. In the comparison, a 3-D cubic copper system consisted of 102,690 atoms in FCC configuration is simulated. The material is modeled with

Morse potential where interactions are truncated beyond the fourth shell of neighbors. The simulations are run for 10,000 iterations and simulation time is considered. The rep atom ratios of MMM models are varied from 0.0078% to 100% (full atomistic) and their simulation times are recorded. The speed-up of MMM models are calculated according to Equation (39).

$$Speed-up=\frac{Simulation\ time\ of\ full\ atomistic\ model}{Simulation\ time\ of\ MMM\ model} \tag{39}$$

In addition, the theoretical speed-up is investigated. The performance of a full atomistic simulation scales with $O(NN_b)$ where N is the number of atoms and N_b is the number of atoms in a ball defined by the cut-off distance [43]. Likewise, the performance of an MMM simulation scales with $O(N_rN_b)$ where N_r is the number of rep atoms. Dropping the identical terms, the MMM speed-up is inversely proportional to the reduction in the rep atom ratio as in Equation (40).

$$Theoretical\ speed-up=\frac{100}{Rep\ atom\ ratio} \tag{40}$$

The numerator of Equation (40) should actually be equal to 1, but it is set to 100 since the rep atom ratio is expressed in percentage. Speed-up of MMM models with respect to varying rep atom ratios along with theoretical speed-ups are shown in Figure 34.

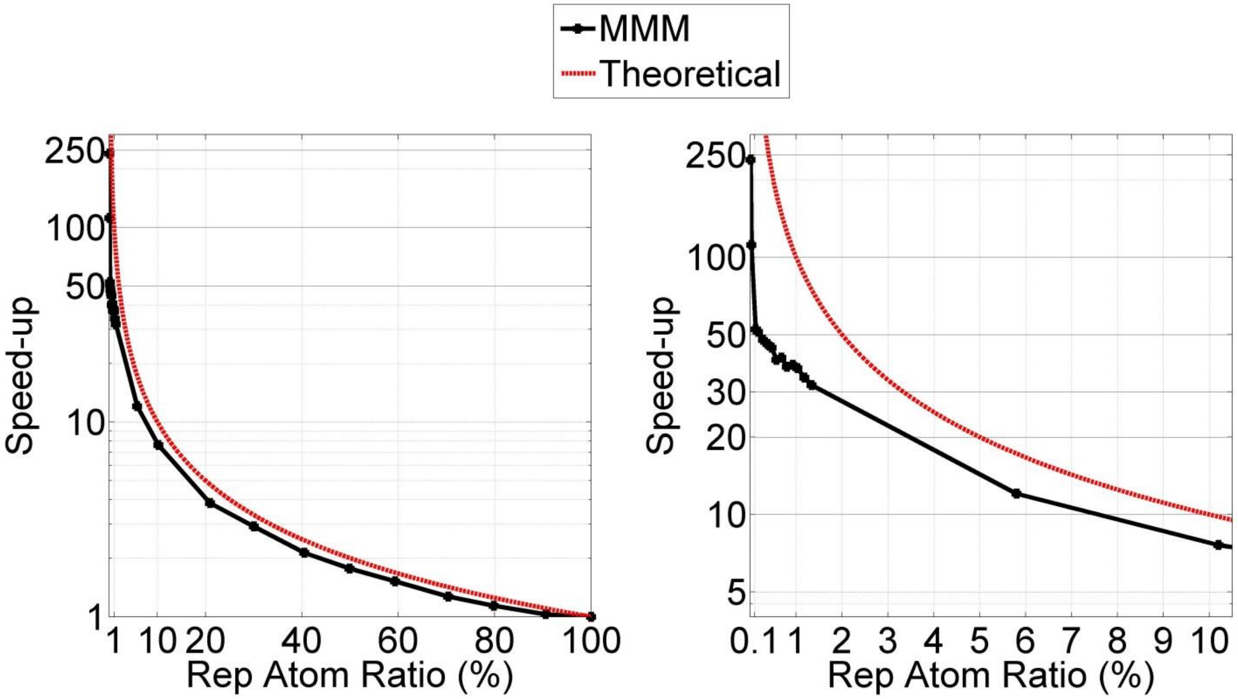

Figure 34. Speed-up of MMM models with respect to varying rep atom ratio (left) and its close up (right) on a semi-log scale.

Figure 34 shows that the actual speed-up closely follows the theoretical speed-up in general; however, a discrepancy is observed for low rep atom ratio. This discrepancy is expected since there is a fixed cost in every simulation and also MMM introduces some additional costs to the simulation. These fixed and additional costs become more apparent when the overall simulation cost is very low as in the case of low rep atom ratio. Nevertheless, it can be concluded that the reduction in the simulation time can be well estimated from the reduction in the rep atom ratio.

Figure 34 shows that speed-up is below 2 times for rep atom ratio higher than 50%; and, from 50% to 10% rep atom ratio, speed-up increases to 8 times. Then, speed-up increases exponentially for rep atom ratio decreasing from 10%. The maximum speed-up (239.2 times) is reached at 0.0078% rep atom ratio, which corresponds to only 8 rep atoms out of 102,690 atoms. These rep atoms are the interpolating rep atoms at the corners of the cube, and it stands to be the minimum number of rep atoms possible for this cube. Therefore, the speed-up reaches to its higher limit at this point. Potentially, this limit may be pushed further by increasing the mesh size. In general, however, when the rep atom ratio is assumed to vary between 5 to 30% on average, the speed-up corresponds to about 3 to 12 times. This assumption of rep atom ratio is reasonable since a lower value indicates use of redundant atoms and a higher value may be too costly, which would defeat the purpose of multiscale modeling.

4.6 PARALLELIZATION

The MMM software is parallelized using the Message Passing Interface (MPI), using MPICH 3.1.3. Two sections of the code are parallelized: (i) energy/force computation and, as noted earlier, (ii) searching for tetrahedral elements where the atoms lie in. The rest of the code is sequentially executed by each processor. Energy/force computation takes about 80-95% (even higher with expensive potentials and/or larger cut-off distances) of the simulation time. Because of the partitioning of atoms onto the processors, each processor is only responsible for computing energies and forces of a group of atoms in every iteration. Following the parallel energy/force computation, energy/force vectors are communicated among processors using the *MPI_Allreduce* command. As a matter of trade-off between storage and communication, Shaw et al. pack/unpack data [94] and Buchholz et al. keep data in MPI-compatible structure [91]. In MMM, data (i.e., energy/force) is stored in 2-D noncontiguous vectors (using matrix format of MMM). Before and after a communication, this 2-D data is packed to and unpacked from 1-D contiguous data. A design choice is made in this way because storing 1-D contiguous data all the time introduced a higher cost due to index operations such as accessing *y*-position of 773th atom by *3x773+1*.

The MMM software employs atom decomposition scheme. As noted earlier, this scheme requires all-to-all communication of energies and forces. For instance, considering 1,000,000 atoms in a system, the force data is about 23 MB. Transfer of such a size from *P* processors to *P* processors is too expensive. However, the MMM framework eliminates the majority of atoms from the system, which leaves only rep atoms that actually matters for the simulation. Thus, it is sufficient to only communicate data of rep atoms, which is much smaller than full atomistics. As a result, the elimination of redundant atoms is not only useful for reducing energy/force computations, but it is also useful for reducing communication costs among processors. It should be noted that a disadvantage of synchronous communication is that every processor has to wait for all other processors to finish the iteration before the simulation can proceed. This synchronization may add an extra few percent to the simulation time. In order to observe communication and wait times separately, two blocking barriers (*MPI_Barrier*) are located before and after the communication sections. By this means, the time each processor spends in the first barrier is recorded as the wait time and the time each processor spends between the two barriers is recorded as the communication time. Communication and wait times are reported at the end of a simulation in its log file.

There is a competing effect between the two following phenomena: cost reductions of MMM and parallelization. In order to demonstrate this competing effect, a 3-D cubic copper system consisted of 102,690 atoms in FCC configuration is simulated. The material is modeled with Morse potential where interactions are truncated beyond the fourth shell of neighbors. A full atomistic model is run on a single processor and an MMM model of 1.3% rep atom ratio is run on 1, 2, and 4 processors. The results are compared in terms of parallel efficiency, which is given by Equation (41) as

$$E_P = \frac{T_s}{PT_P} \tag{41}$$

where E_p is the parallel efficiency for P processors, T_s is the serial time, and T_P is the parallel time for P processors. The results are listed in Table 7.

Table 7. Parallel performance of the MMM software by comparing full atomistic model to MMM model on 1, 2, and 4 processors.

	Full Atomistic	**MMM 1P**	**MMM 2P**	**MMM 4P**
Simulation Time (min)	394.7	12.4	7.8	6.5
Parallel Efficiency of Simulation (%)	-	-	79.5%	47.7%
Parallel Time (min) – Ratio to Simulation Time	391.6 – 99.2%	9.1 – 73.4%	4.6 – 58.2%	2.6 – 39.2%
Parallel Efficiency of Parallel Portion (%)	-	-	98.9%	87.5%
Serial Time (min) – Ratio to Simulation Time	3.1 – 0.8%	3.3 – 26.6%	3.2 – 41.8%	3.9 – 60.8%

As Table 7 shows, MMM run on a single processor reduces the simulation time of the example about 43.0 times. More importantly, this reduction is mostly effective on the parallel portion of the code as in the case of a parallel simulation. Therefore, both cost reductions of MMM and parallelization are effective on the parallel portion of the code, hence their effects are competing. With ideal scaling behavior, parallelization would require 43 processors to achieve the same amount of reduction. However, MMM software achieved this reduction with only one single processor. As a result, the conclusion is that the MMM software needs fewer processors.

From full atomistic to MMM on single processor, the time that the parallel portion of the code spends is reduced from 99.2% to 73.4% percent. However, parallel efficiency of a code strongly depends on the parallel portion of the code. This phenomenon is described by Amdahl’s law as given by Equation (42).

$$S_P = \frac{1}{(1-B)+\frac{B}{P}} \tag{42}$$

where S_P is the speed-up for P processors and B is the parallel portion of the code. For instance, if the parallel portion of a code is 90% (B = 0.9) and P is assumed to be infinity, maximum speed-up of the code is only 10 times. In this regard, Amdahl’s law proves that speed-up of a code is very sensitive to the parallel portion of the code.

In order to further evaluate the scaling of the MMM software, a 3-D cubic copper system consisted of 164,255 atoms in FCC configuration is simulated. The material is modeled with Morse potential where interactions are truncated beyond the fourth shell of neighbors. An MMM model of 29.7% rep atom ratio is run by the MMM software on 1, 2, 4, 8, and 16 processors. The results are shown in Figure 35.

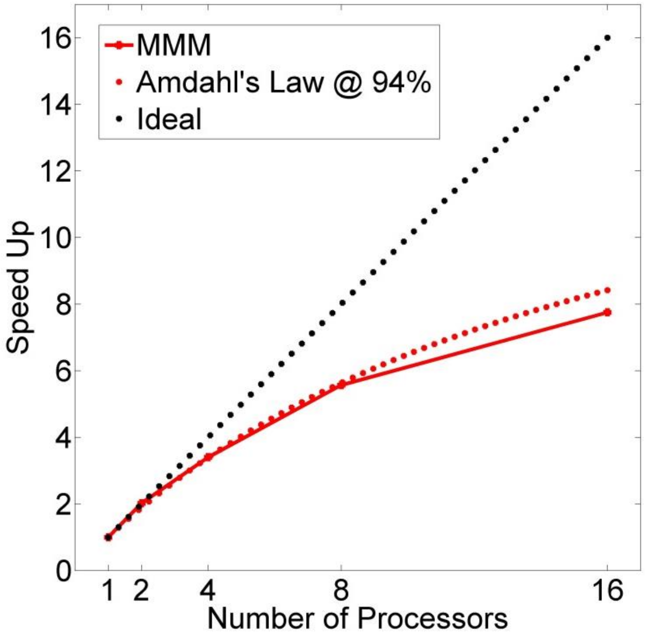

Figure 35. Scaling of MMM and estimation by Amdahl's law. The black dotted line represents the ideal scaling.

Figure 35 also shows the estimation by Amdahl's law. The parallel portion of the MMM code for this model is taken from the MMM run on a single processor where the parallel time is recorded as 94% of the simulation time. As expected, this number well approximates the parallel portion of the MMM code for this model and, when input to Amdahl's law, closely captures the MMM speed-up. The scaling of MMM is better for 1, 2, and 8 processors compared to 16 processors. The decline in the efficiency is due to the fact that the communication time is starting to take over the computation time for more than 16 processors. In a full atomistic simulation on 16 processors, each processor has about 10,000 atoms, thus the decline in efficiency could be expected for a bigger number of processors. However, an MMM simulation is different than a full atomistic simulation since the actual number of atoms that are included in the computations is the number of rep atoms rather than the total number of atoms. In MMM run on 16 processors, there are about 50,000 rep atoms and each processor has about 3,000 rep atoms per processor, which explains the decline in efficiency. Consequently, the scaling of MMM software is governed by the actual number of atoms included in the computations, hence the estimations should be better based on number of rep atoms instead of the total number of atoms.

4.7 NUMERICAL EXAMPLE: 2-D NANOINDENTATION

The MMM method is tested for a 2-D nanoindentation numerical example to demonstrate its efficiency. The copper model is consisted of a rectangular prism of 1,009,441 atoms in hexagonal configuration and has width of 262.1 nm and height of 211.9 nm. The material is modeled with Morse potential where interactions are truncated beyond the fourth shell of neighbors. The atoms are initially relaxed by both static energy minimization and dynamics. The MMM model is coarsened by 712 elements of size 12 nm, which corresponds to about 1,400 atoms per element. The lateral edges of the system are fixed in horizontal dimensions and bottom edge is fixed in all dimensions. Temperature of the system is kept constant at 1 K by the Berendsen thermostat. The simulations are run for 175 ps at a timestep of 0.5 fs. An indenter is pushed towards the top surface with respect to Equation (34) where K is 10 eV/Å^2$_,$ and R is 40 Å. The indenter is pushed constantly at a speed of 200 Å/ns.

The adaptivity scheme employed in this example is slightly different than the one presented in the Adaptivity Chapter. In the current scheme, two potential energy thresholds are input: lower and higher. Thus, an atom is activated for refinement if its energy fits into the input interval. This way, surface atoms can be excluded from activation as is the case in this example. In addition, the proximity is now utilized to search for nearby elements (rather than nearby atoms). That is, elements with centroids within the influence radius of an excited atom are activated for refinement. Last, a new adaptivity criterion is introduced to enforce activation of elements that are neighbors to the active elements. Two elements are defined as neighbors if they share a node. These changes contributed to the efficiency of the adaptivity scheme that is critical to simulate large systems. In this respect, potential energy thresholds are input as -1.67 (94% of the initial energy value) and -1.47 eV (82% of the initial energy value) and influence radius is input as 240 Å. The adaptivity scheme is called in every 1,000 iterations. Results are compared to LAMMPS in terms of accuracy and efficiency. Force-depth curves are shown in Figure 36, snapshots from simulations are shown in Figure 37 and Figure 38, and rep atom ratio and iteration time evolutions are shown in Figure 39.

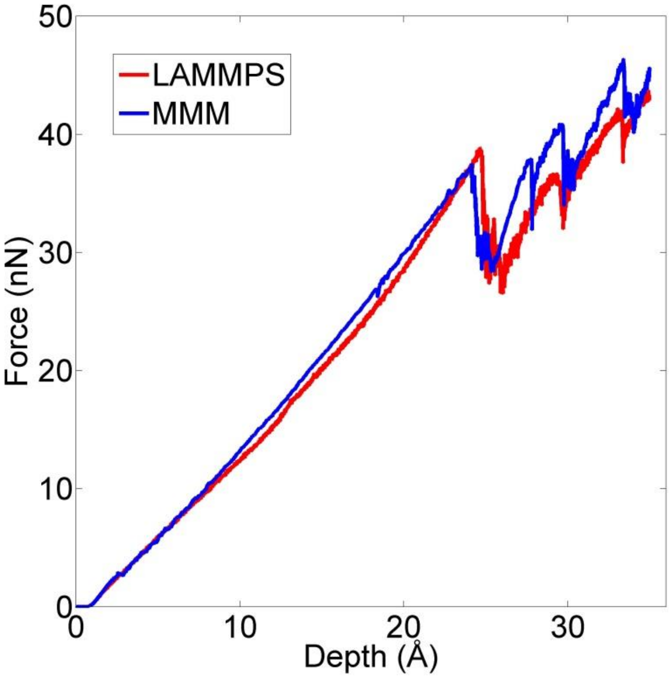

Figure 36 Comparison of force-depth curves of LAMMPS and MMM simulations of 2-D nanoindentation example.

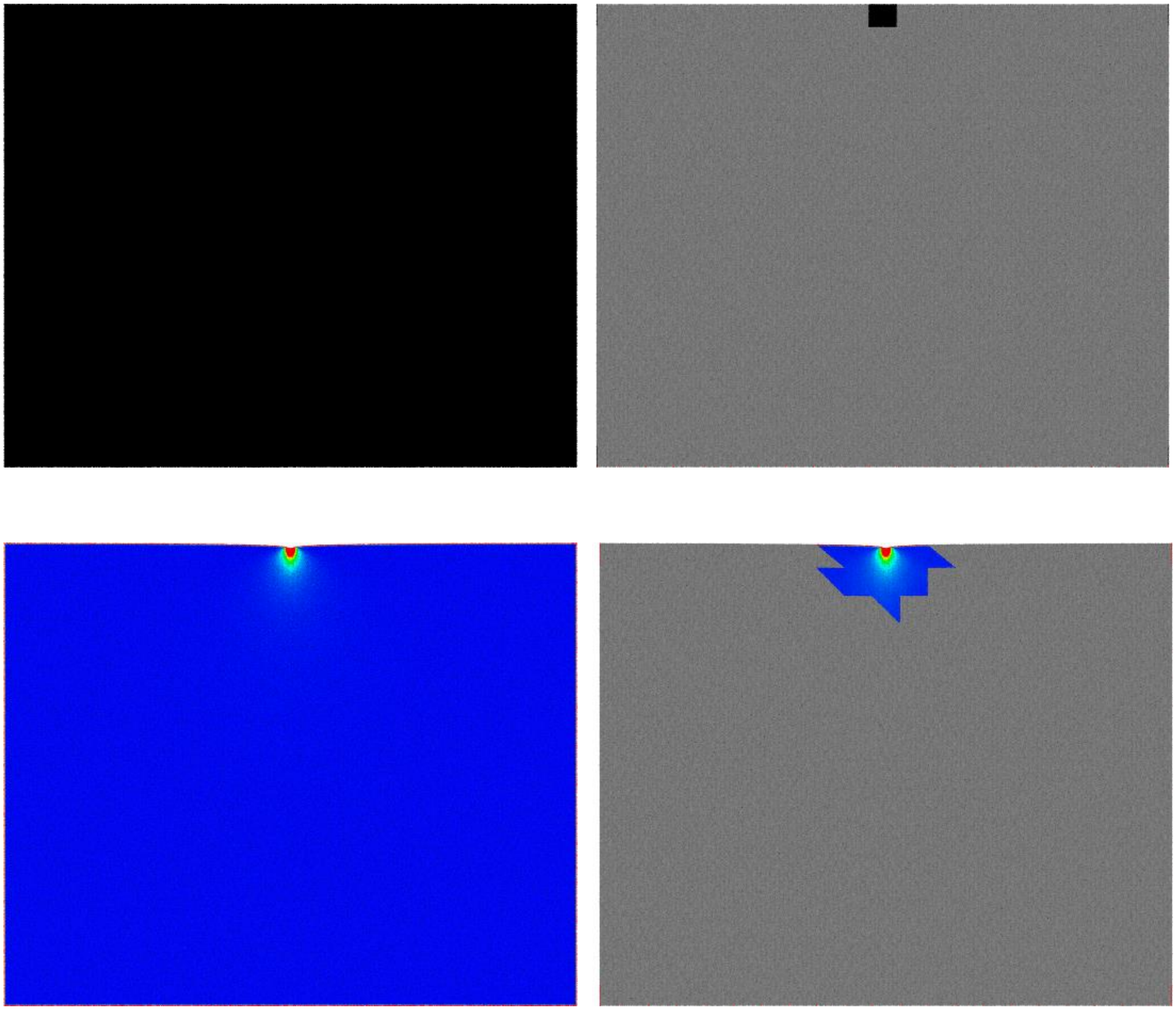

Figure 37. Snapshots of LAMMPS (left) and MMM (right) simulations of 2-D nanoindentation example. The top panel shows the beginning of simulations where LAMMPS is full of non-interpolating rep atoms (black) and MMM has only a small region of non-interpolating rep atoms with the rest is dominated by non-sampling atoms (gray). The bottom panel shows the onset of dislocation nucleation and color coded with respect to potential energy. In the MMM figure, only fine regions are color coded with respect to potential energy and the rest is color coded with respect to atom type.

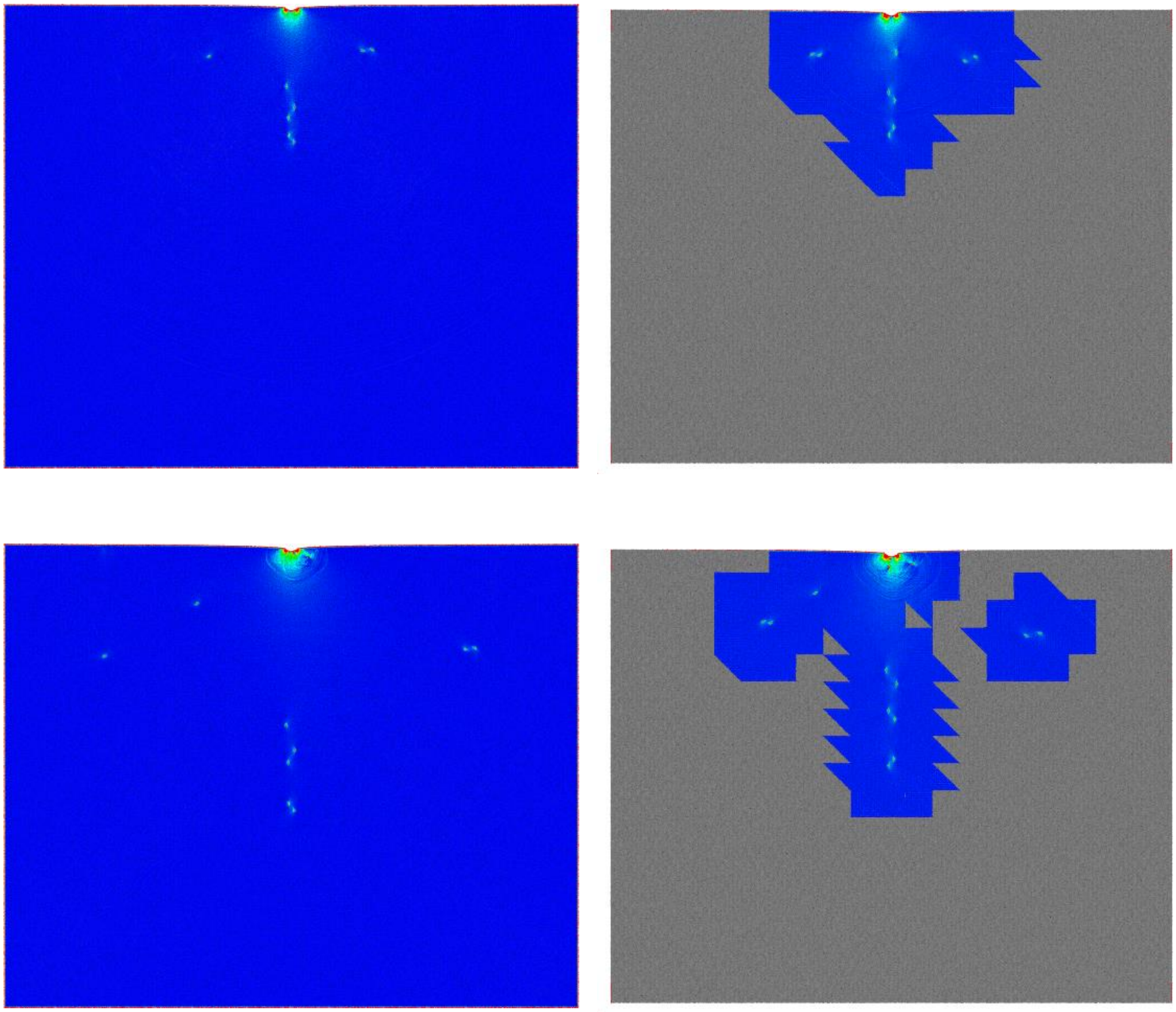

Figure 38. Snapshots of LAMMPS and MMM simulations of 2-D nanoindentation example as the dislocations propagate (continued from previous figure).

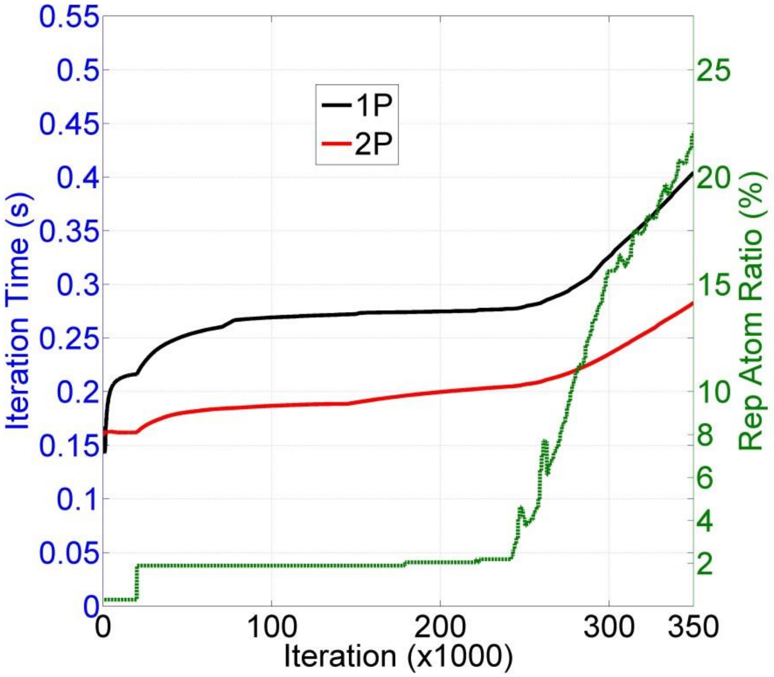

Figure 39. Iteration time vs. iteration with 1 and 2 processors; and, rep ratio vs. iteration of 2-D nanoindentation example.

Figure 36 shows that force-depth curve of MMM agrees well with that of LAMMPS. MMM closely captures the elastic modulus, yield stress, yield strain, and hardening along with three bursts in the stress-strain curve. Figure 37 and Figure 38 show that the dislocation distributions of MMM match that obtained from MD simulation using LAMMPS. In MMM, this match is achieved by only refining the localized regions of the dislocations and efficiently coarsening the rest. This high level of accuracy is expected due to earlier results presented in Dynamics and Adaptivity Chapters.

More importantly, this Chapter focuses on the efficiency results shown in Figure 39. The rep atom ratio is 0.3% in the beginning. As the indenter penetrates into the material, the material starts deforming. As a result, the rep atom ratio increases to 1.9% at iteration number 20,000. In correspondence, iteration times of both single and two processors starts increasing from iteration number 20,000 onward. The iteration time at an iteration number is calculated as an average from the beginning of the simulation until that iteration. Therefore, the reactions of iteration times to increases in rep atom ratio are not instant but rather cumulative. The rep atom ratio sharply increases from iteration number 242,000 onward. This time, reactions of iteration times are much smoother due to the long history of simulation. The increase in the rep atom ratio leads to substantial increase in the number of rep atoms. For instance, 10% rep atom ratio corresponds to approximately 100,000 rep atoms. In turn, the cost of simulation substantially increases too. This additional cost is undertaken by a single or two processors. As evident from the figure, the increase in iteration time is sharper in the former whereas the increase in iteration time is smoother in the latter as a result of relative costs per processor. Final iteration times are recorded at the final iteration and listed in Table 8.

Table 8. Iteration times of LAMMPS and MMM on various number of processors.

Software	Number of Processors	Iteration Time (s)	Efficiency
LAMMPS	4	0.8611	~8.5 times
LAMMPS	8	0.4304	
MMM	1	0.4043	
LAMMPS	6	0.5819	~6.3 times
LAMMPS	12	0.3040	
MMM	2	0.2828	

Efficiency of MMM here is defined by Equation (43) as

$$Efficiency\ of\ MMM = \frac{t_{iter,FA}}{t_{iter,MMM}} x \frac{P_{iter,FA}}{P_{iter,MMM}} \tag{43}$$

where $t_{iter,FA}$ is the iteration time of full atomistic simulation, $t_{iter,MMM}$ is the iteration time of MMM simulation, $P_{iter,FA}$ is the number of processors used in full atomistic simulation, and $t_{iter,FA}$ is the number of processors used in MMM simulation. According Table 8, when we take the gray rows as a case, the MMM iteration time on a single processor is close to LAMMPS iteration time on 8 processors. Alternatively, iteration time of MMM on a single processor is slightly lower than half of LAMMPS iteration time on 4 processors. Either option amounts to ~8.5 times in efficiency. When we take the white rows as a case and compare results in the same way, either option amounts to ~6.3 times in efficiency.

The reduction in number of processors is included in the definition of efficiency of MMM since MMM shows a better performance with low number of processors due to its scaling behavior as discussed earlier. As a matter of fact, the reduction in number of processors is considered to be as important as reduction in iteration time because it enables simulation of large systems on desktop computers instead of supercomputers. In the current example, for instance, a system of about one million atoms can be efficiently simulated on a desktop computer on a single processor or two processors. Taking into account that current generation of desktop computers has more than four processors on average, running a simulation of one million atoms on a few processors is very feasible. In addition, the fact that maintaining a supercomputing facility over 3 year period equals the hardware costs [102], eliminating the need for large number of processors is even more important.

An important point regarding the numerical example presented is that nanoindentation stands to be a tough case since rep atom ratio considerably increases over time. The effect of increase in the rep atom ratio on iteration times is evident in Figure 39. However, nanoindentation example is chosen on purpose to show that MMM is efficient in the worst case. Numerical examples with localized defects have nearly constant rep atom ratio, e. g., 1-D wave propagation or 2-D dislocation in Adaptivity Chapter; thus, they are expected to show better efficiency.

In conclusion, the efficiency of MMM is discussed by a 2-D nanoindentation numerical example. In that, the accuracy is matched very well. An efficiency of 6.3 – 8.5 times is achieved by means of combined reductions in iteration time and processor number where the latter is more pronounced. The reduction in processor number is believed to be very promising to enable simulations of large system on regular desktop computers, which in turn eliminates the need of using expensive supercomputers.

5.0 CONCLUSION

Dynamics, adaptivity analysis, and implementation of the concurrent atomistic/continuum coupling method MMM are presented. Governing equations of the dynamics are derived from a Hamiltonian that systematically approximates the energy of the original system. Further, the refinement and coarsening procedures of the adaptivity scheme along with adaptivity criteria are outlined in detail. Last, the MMM software is presented. The structure of the software and many implementation aspects are described. The efficiency and scalability of the software are discussed as well. The dynamics, adaptivity, and efficiency of the method are demonstrated with many numerical examples including wave and crack propagations, dislocation glide, nanoindentation, and modal analysis in 1/2/3 dimensions. All results are compared to true full atomistic solutions. Good agreement is observed in all numerical examples. Also, the MMM method is shown to have 6.3 – 8.5 times efficiency by means of a combined reduction in simulation time and number of processors. In conclusion, the method is proved to be accurate, efficient, flexible, and consistent.

5.1 MAIN CONTRIBUTIONS

The main contributions of this dissertation include the following:

- This research has shown that dynamics concurrent atomistic/continuum coupling is feasible under the MMM framework. The energy-based MMD method is distinguished by initializing from a full atomistic model and systematically eliminating the degrees of freedom of the system. This way, the MMD method does not differentiate between the atomistic and continuum descriptions. Employing atomistic description everywhere in the domain, the MMD method eliminates the requirement for a specialized continuum model. Comparison of MMD solutions to true full atomistic solutions demonstrates that the MMD method produces high accuracy.
- The MMD method does not need to implement a cumbersome handshake region at the atomistic/continuum interface. A thermostat, also utilized to regulate the temperature of the system, mitigates the adverse effects of the high frequency wave reflections in the atomistic region. The high accuracy of the MMD method further supports the adopted approach.
- The MMD method is capable of capturing the fine details of wave fronts in wave problems. This is accomplished by adaptively tracking the wave front with atomistic description during the course of the simulation.
- Modal analysis shows that the approximation of full atomistic solution extends to calculation of natural frequencies. The monotonic convergence of the MMD solutions with respect to decreasing mesh size indicates the consistency of the method.

- The target efficiency of the MMM method is retained by an adaptive scheme. The scheme is able to accurately switch the subregions of the domain between the atomistic/continuum descriptions. As the refinement captures the progress of the material defects, coarsening retains the target efficiency. Restricting it to a confined region, overuse of expensive atomistic description is avoided.
- The efficient, readable, user-friendly, and portable MMM software is presented to let the users to run their own multiscale simulations. A short manual and testing are further presented and the source code is publicly shared to let the developers to extend and modify the MMM software. The efficiency of the software is validated by presenting the efficiency improvements and demonstrating that the method is as fast as one of the state-of-the-art MD codes (i.e., LAMMPS).
- The MMM method reaches the originally intended purpose: efficient simulations with high accuracy. The speed-up of the MMM method is demonstrated to be inversely proportional to the rep atom ratio. Even though the edge cases are presented, realistic expectations of the speed-up are discussed. Agreement of the actual speed-up with the theoretical speed-up provides a tool to closely estimate the cost of the simulation.
- The MMM software requires much fewer processors than a standard MD code. Reducing the cost of the parallel portion of the code, the MMM software lowers the required number of processors. Consequently, problems with up to millions of atoms can be efficiently simulated on desktop computers with only a few processors. In turn, the need for high-maintenance and expensive supercomputers is eliminated.

5.2 FUTURE WORK

The MMM method is introduced for statics [46] and dynamics [77] followed by convergence and error structure analysis [78] and introduction of a unified and consistent framework for general FE shape functions [79]. Also, adaptivity analysis and implementation are presented in the current work. Still, the MMM method needs to be further developed in many directions:

- The MMM method features pair potentials including spring, Lennard-Jones (LJ), and Morse. However, it is indispensable to extend the method to many-body potentials. For instance, the EAM potential is different than the pair potentials since it loops the neighbors twice in every iteration: to account for the electron charge density in the first iteration and to perform the actual computations in the second iteration. That's why; the EAM potential must be treated differently than the pair potentials.
- The dynamics of the MMM method is limited to zero temperature. This incapability is limiting in the sense that the heat problems and problems where the heat may play an important role cannot be simulated. Therefore, the MMM method should be extended to finite temperature dynamics. It has been introduced to some multiscale methods as cited in the introduction. The extension to finite temperature substantially affects the dynamics of the system thus requires some sophisticated approach.

5.2.1 Discussion on Scaling Performance of the MMM Software

The scaling behavior of the MMM software is previously presented in the Implementation Chapter. Although the MMM software needs fewer processors and also shows good scaling performance, there is still more room for improvement of the scaling performance of the software. In the following, a few potential improvements are going to be discussed. First, the serial portion of the code can be parallelized as well since it is growing in proportion as the time spent in the parallel portion of the code is reduced. However, in an atom decomposition scheme, the parallelization of the serial portion of the code would require more communication, cost of which may outweigh the reduction in cost of the parallel portion. A better solution is to implement domain decomposition since it naturally parallelizes more parts of the code and requires less communication. However, domain decomposition may not be suitable for MMM models for a few reasons. First, MMM models are highly heterogeneous and vibrant that may often require migration of atoms. However, migration of atoms is difficult to implement and adds an extra cost [103]. The difficulties regarding parallel implementation of adaptive systems is also evident from the literature on the subject in FE [104-106]. Second, ghost atoms have long-range effects that might extend beyond the neighbor boxes of a domain decomposition scheme [103]. For instance, position of a ghost atom has to be interpolated by the interpolating rep atoms of the element it belongs to. These interpolating rep atoms may be in a faraway box for a coarse mesh size. It is noteworthy that domain decomposition schemes owe their efficiency to minimal volume of communication among thin surface layers of neighbor boxes. That is why; long-range effects of ghost atoms may disrupt the efficient communication of domain decomposition schemes. Third, the relationship between ghost atoms and interpolating rep atoms are beyond the interpolation of positions of ghost atoms. For instance, forces of a ghost atom have to be extrapolated to the interpolating rep atom of the element it belongs to. For another instance, mass and velocity of a ghost atom is represented by the interpolating rep atoms. In case of an adaptive refinement, mass and velocity of a ghost atom are derived from the interpolating rep atoms of the element it belongs to. As in the second reason, these long-range relationships may disrupt the efficient communication of domain decomposition schemes. Despite all these factors, domain decomposition scheme may be inevitable in order to attain much larger length and time scales due to the fact that it is more memory efficient. The domain decomposition scheme disintegrates the data structures onto the processors, which is a labor-intensive task since its implementation would substantially change the software.

Another improvement to parallelize the serial portion of the MMM software is to employ OPENMP instead of MPI. OPENMP provides easy parallelization of the loops in a code without an effort to build communication schemes. In order to compare OPENMP to MPI, a 3-D cubic copper system consisted of 9,842 atoms in FCC configuration with 7.9% rep atom ratio is simulated. The material is modeled with Morse potential where interactions are truncated beyond the fourth shell of neighbors. The simulation is run with MPI on 1, 2, 4, 8, and 16 processors and with OPENMP on 1, 2, 4, 8, and 16 threads. With OPENMP, more sections of the codes including integration, thermostat, and boundary conditions are parallelized. Parallel efficiencies of MMM with MPI, MMM with OPENMP, and LAMMPS are compared in Figure 40.

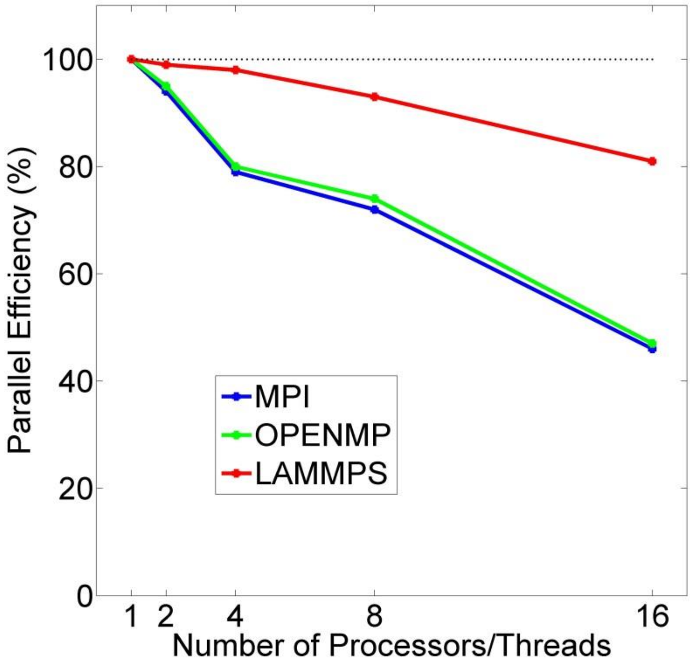

Figure 40. Comparison of parallel efficiencies of MMM with MPI, MMM with OPENMP, and LAMMPS.

Figure 40 shows that performances of MPI and OPENMP are similar although the parallel portion of the latter is bigger. The close track of MPI with less the parallel portion is attributed to the fact that MPI is known to be more efficient than OPENMP. It is noteworthy that although the respective performances of MPI and OPENMP are similar, OPENMP is much more efficient in memory usage since MPI duplicates data for each processor in an atom decomposition scheme. As a result, when compared to MPI, memory usage of OPENMP is as few as the number of processors. A disadvantage of OPENMP is that it is limited to shared memory environments while MPI is compatible with both shared and distributed memory environments.

BIBLIOGRAPHY

[1] P. Aubertin, J. Réthoré, R. De Borst, Energy conservation of atomistic/continuum coupling, International journal for numerical methods in engineering, 78 (2009) 1365-1386.

[2] M. Jebahi, F. Dau, J.-L. Charles, I. Iordanoff, Multiscale modeling of complex dynamic problems: an overview and recent developments, Archives of Computational Methods in Engineering, (2014) 1-38.

[3] F.F. Abraham, J.Q. Broughton, N. Bernstein, E. Kaxiras, Spanning the length scales in dynamic simulation, Computers in Physics, 12 (1998) 538-546.

[4] D. Jang, X. Li, H. Gao, J.R. Greer, Deformation mechanisms in nanotwinned metal nanopillars, Nature nanotechnology, 7 (2012) 594-601.

[5] W.K. Liu, E. Karpov, S. Zhang, H. Park, An introduction to computational nanomechanics and materials, Computer Methods in Applied Mechanics and Engineering, 193 (2004) 1529-1578.

[6] Y. Mo, K.T. Turner, I. Szlufarska, Friction laws at the nanoscale, Nature, 457 (2009) 1116-1119.

[7] R.E. Miller, E.B. Tadmor, The quasicontinuum method: Overview, applications and current directions, Journal of Computer-Aided Materials Design, 9 (2002) 203-239.

[8] R. Gracie, T. Belytschko, An adaptive concurrent multiscale method for the dynamic simulation of dislocations, International Journal for Numerical Methods in Engineering, 86 (2011) 575-597.

[9] F.F. Abraham, R. Walkup, H. Gao, M. Duchaineau, T.D. De La Rubia, M. Seager, Simulating materials failure by using up to one billion atoms and the world's fastest

computer: Work-hardening, Proceedings of the National Academy of Sciences, 99 (2002) 5783-5787.

[10] J. Marian, J. Knap, G. Campbell, A quasicontinuum study of nanovoid collapse under uniaxial loading in Ta, Acta Materialia, 56 (2008) 2389-2399.

[11] D. Qian, G.J. Wagner, W.K. Liu, A multiscale projection method for the analysis of carbon nanotubes, Computer Methods in Applied Mechanics and Engineering, 193 (2004) 1603-1632.

[12] R.E. Miller, E.B. Tadmor, Hybrid continuum mechanics and atomistic methods for simulating materials deformation and failure, MRS bulletin, 32 (2007) 920-926.

[13] N.M. Ghoniem†, E.P. Busso, N. Kioussis, H. Huang, Multiscale modelling of nanomechanics and micromechanics: an overview, Philosophical magazine, 83 (2003) 3475-3528.

[14] W.A. Curtin, R.E. Miller, Atomistic/continuum coupling in computational materials science, Modelling and simulation in materials science and engineering, 11 (2003) R33.

[15] J. Fish, M.A. Nuggehally, M.S. Shephard, C.R. Picu, S. Badia, M.L. Parks, M. Gunzburger, Concurrent AtC coupling based on a blend of the continuum stress and the atomistic force, Computer Methods in Applied Mechanics and Engineering, 196 (2007) 4548-4560.

[16] R.E. Miller, E. Tadmor, A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods, Modelling and simulation in materials science and engineering, 17 (2009) 053001.

[17] B. Eidel, A. Stukowski, A variational formulation of the quasicontinuum method based on energy sampling in clusters, Journal of the Mechanics and Physics of Solids, 57 (2009) 87-108.

[18] M.L. Parks, P.B. Bochev, R.B. Lehoucq, Connecting atomistic-to-continuum coupling and domain decomposition, Multiscale Modeling & Simulation, 7 (2008) 362-380.

[19] Y. Zhang, M. Gunzburger, Quadrature-rule type approximations to the quasicontinuum method for long-range interatomic interactions, Computer Methods in Applied Mechanics and Engineering, 199 (2010) 648-659.

[20] L. Shilkrot, R. Miller, W. Curtin, Coupled atomistic and discrete dislocation plasticity, Physical Review Letters, 89 (2002) 025501.

[21] B. Luan, S. Hyun, J. Molinari, N. Bernstein, M.O. Robbins, Multiscale modeling of two-dimensional contacts, Physical Review E, 74 (2006) 046710.

[22] X. Nie, M.O. Robbins, S. Chen, Resolving singular forces in cavity flow: multiscale modeling from atomic to millimeter scales, Physical review letters, 96 (2006) 134501.

[23] T. Belytschko, S. Xiao, Coupling methods for continuum model with molecular model, International Journal for Multiscale Computational Engineering, 1 (2003).

[24] S. Badia, M. Parks, P. Bochev, M. Gunzburger, R. Lehoucq, On atomistic-to-continuum coupling by blending, Multiscale Modeling & Simulation, 7 (2008) 381-406.

[25] P. Moseley, J. Oswald, T. Belytschko, Adaptive atomistic-to-continuum modeling of propagating defects, International Journal for Numerical Methods in Engineering, 92 (2012) 835-856.

[26] P. Moseley, J. Oswald, T. Belytschko, ADAPTIVE ATOMISTIC-CONTINUUM MODELING OF DEFECT INTERACTION WITH THE DEBDM, International Journal for Multiscale Computational Engineering, 11 (2013).

[27] W. Shan, U. Nackenhorst, An adaptive FE–MD model coupling approach, Computational Mechanics, 46 (2010) 577-596.

[28] J. Tinsley Oden, S. Prudhomme, A. Romkes, P.T. Bauman, Multiscale modeling of physical phenomena: Adaptive control of models, SIAM Journal on Scientific Computing, 28 (2006) 2359-2389.

[29] V. Shenoy, R. Miller, E. Tadmor, D. Rodney, R. Phillips, M. Ortiz, An adaptive finite element approach to atomic-scale mechanics—the quasicontinuum method, Journal of the Mechanics and Physics of Solids, 47 (1999) 611-642.

[30] T. Shimokawa, J.J. Mortensen, J. Schiøtz, K.W. Jacobsen, Matching conditions in the quasicontinuum method: Removal of the error introduced at the interface between the coarse-grained and fully atomistic region, Physical Review B, 69 (2004) 214104.

[31] M. Xu, T. Belytschko, Conservation properties of the bridging domain method for coupled molecular/continuum dynamics, International Journal for Numerical Methods in Engineering, 76 (2008) 278-294.

[32] S. Xiao, T. Belytschko, A bridging domain method for coupling continua with molecular dynamics, Computer Methods in Applied Mechanics and Engineering, 193 (2004) 1645-1669.

[33] G.J. Wagner, W.K. Liu, Coupling of atomistic and continuum simulations using a bridging scale decomposition, Journal of Computational Physics, 190 (2003) 249-274.

[34] A. Sadeghirad, A. Tabarraei, A damping boundary condition for coupled atomistic–continuum simulations, Computational Mechanics, (2013) 1-17.

[35] A.C. To, S. Li, Perfectly matched multiscale simulations, Physical Review B, 72 (2005) 035414.

[36] S. Li, X. Liu, A. Agrawal, A.C. To, Perfectly matched multiscale simulations for discrete lattice systems: Extension to multiple dimensions, Physical Review B, 74 (2006) 045418.

[37] S. Qu, V. Shastry, W. Curtin, R. Miller, A finite-temperature dynamic coupled atomistic/discrete dislocation method, Modelling and simulation in materials science and engineering, 13 (2005) 1101.

[38] E.B. Tadmor, M. Ortiz, R. Phillips, Quasicontinuum analysis of defects in solids, Philosophical Magazine A, 73 (1996) 1529-1563.

[39] E. Weinan, J. Lu, J.Z. Yang, Uniform accuracy of the quasicontinuum method, Physical Review B, 74 (2006) 214115.

[40] J. Knap, M. Ortiz, An analysis of the quasicontinuum method, Journal of the Mechanics and Physics of Solids, 49 (2001) 1899-1923.

[41] M. Luskin, C. Ortner, An analysis of node-based cluster summation rules in the quasicontinuum method, SIAM Journal on Numerical Analysis, 47 (2009) 3070-3086.

[42] R. Miller, E. Tadmor, R. Phillips, M. Ortiz, Quasicontinuum simulation of fracture at the atomic scale, Modelling and simulation in materials science and engineering, 6 (1998) 607.

[43] M. Gunzburger, Y. Zhang, A quadrature-rule type approximation to the quasi-continuum method, Multiscale Modeling & Simulation, 8 (2010) 571-590.

[44] P. Lin, Convergence analysis of a quasi-continuum approximation for a two-dimensional material without defects, SIAM Journal on Numerical Analysis, 45 (2007) 313-332.

[45] L. Beex, R. Peerlings, M. Geers, Central summation in the quasicontinuum method, Journal of the Mechanics and Physics of Solids, 70 (2014) 242-261.

[46] Q. Yang, E. Biyikli, A.C. To, Multiresolution Molecular Mechanics: Statics, Computer Methods in Applied Mechanics and Engineering, (2013).

[47] V. Shenoy, V. Shenoy, R. Phillips, Finite temperature quasicontinuum methods, in: MRS Proceedings, Cambridge Univ Press, 1998, pp. 465.

[48] E. Tadmor, G. Smith, N. Bernstein, E. Kaxiras, Mixed finite element and atomistic formulation for complex crystals, Physical Review B, 59 (1999) 235.

[49] J.Y. Park, S. Im, Adaptive nonlocal quasicontinuum for deformations of curved crystalline structures, Physical Review B, 77 (2008) 184109.

[50] D.M. Kochmann, G.N. Venturini, A meshless quasicontinuum method based on local maximum-entropy interpolation, Modelling and simulation in materials science and engineering, 22 (2014) 034007.

[51] S. Zhang, R. Khare, Q. Lu, T. Belytschko, A bridging domain and strain computation method for coupled atomistic–continuum modelling of solids, International Journal for Numerical Methods in Engineering, 70 (2007) 913-933.

[52] E. Marenic, J. Soric, A. Ibrahimbegovic, Adaptive modelling in atomistic-to-continuum multiscale methods, Journal of the Serbian Society for Computational Mechanics/Vol, 6 (2012) 169-198.

[53] H.S. Park, E.G. Karpov, W.K. Liu†, P.A. Klein, The bridging scale for two-dimensional atomistic/continuum coupling, Philosophical Magazine, 85 (2005) 79-113.

[54] W.K. Liu, H.S. Park, D. Qian, E.G. Karpov, H. Kadowaki, G.J. Wagner, Bridging scale methods for nanomechanics and materials, Computer Methods in Applied Mechanics and Engineering, 195 (2006) 1407-1421.

[55] E. Karpov, H. Park, W.K. Liu, A phonon heat bath approach for the atomistic and multiscale simulation of solids, International Journal for Numerical Methods in Engineering, 70 (2007) 351-378.

[56] D. Qian, W.K. Liu, Q. Zheng, Concurrent quantum/continuum coupling analysis of nanostructures, Computer Methods in Applied Mechanics and Engineering, 197 (2008) 3291-3323.

[57] H. Kadowaki, W.K. Liu, Bridging multi-scale method for localization problems, Computer Methods in Applied Mechanics and Engineering, 193 (2004) 3267-3302.

[58] H.S. Park, E.G. Karpov, P.A. Klein, W.K. Liu, Three-dimensional bridging scale analysis of dynamic fracture, Journal of Computational Physics, 207 (2005) 588-609.

[59] D.E. Farrell, H.S. Park, W.K. Liu, Implementation aspects of the bridging scale method and application to intersonic crack propagation, International Journal for Numerical Methods in Engineering, 71 (2007) 583-605.

[60] D. Farrell, E. Karpov, W. Liu, Algorithms for bridging scale method parameters, Computational Mechanics, 40 (2007) 965-978.

[61] L. Xiong, Q. Deng, G. Tucker, D.L. McDowell, Y. Chen, A concurrent scheme for passing dislocations from atomistic to continuum domains, Acta Materialia, 60 (2012) 899-913.

[62] L. Xiong, Q. Deng, G.J. Tucker, D.L. McDowell, Y. Chen, Coarse-grained atomistic simulations of dislocations in Al, Ni and Cu crystals, International Journal of Plasticity, 38 (2012) 86-101.

[63] P.R. Budarapu, R. Gracie, S.P. Bordas, T. Rabczuk, An adaptive multiscale method for quasi-static crack growth, Computational Mechanics, 53 (2014) 1129-1148.

[64] S.-W. Yang, P.R. Budarapu, D.R. Mahapatra, S.P. Bordas, G. Zi, T. Rabczuk, A meshless adaptive multiscale method for fracture, Computational Materials Science, 96 (2015) 382-395.

[65] Y. Liu, X. Zhang, K. Sze, M. Wang, Smoothed molecular dynamics for large step time integration, Cmes-Computer Modeling in Engineering and Sciences, 20 (2007) 176-192.

[66] W. Hankui, Z. Xiong, Q. Xinming, Adaptive smoothed molecular dynamics for multiscale modeling, Computational Materials Science, 46 (2009) 713-715.

[67] M. Praprotnik, L. Delle Site, K. Kremer, Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly, The Journal of chemical physics, 123 (2005) 224106.

[68] A. Heyden, D.G. Truhlar, Conservative algorithm for an adaptive change of resolution in mixed atomistic/coarse-grained multiscale simulations, Journal of Chemical Theory and Computation, 4 (2008) 217-221.

[69] Q. Yang, E. Biyikli, P. Zhang, R. Tian, A.C. To, Atom collocation method, Computer Methods in Applied Mechanics and Engineering, 237 (2012) 67-77.

[70] H.S. Park, P.A. Klein, G.J. Wagner, A surface Cauchy–Born model for nanoscale materials, International Journal for Numerical Methods in Engineering, 68 (2006) 1072-1095.

[71] J.Q. Broughton, F.F. Abraham, N. Bernstein, E. Kaxiras, Concurrent coupling of length scales: methodology and application, Physical review B, 60 (1999) 2391.

[72] J. Fish, Bridging the scales in nano engineering and science, Journal of Nanoparticle Research, 8 (2006) 577-594.

[73] G. Lu, E. Kaxiras, An overview of multiscale simulations of materials, arXiv preprint cond-mat/0401073, (2004).

[74] H.S. Park, W.K. Liu, An introduction and tutorial on multiple-scale analysis in solids, Computer Methods in Applied Mechanics and Engineering, 193 (2004) 1733-1772.

[75] J. Wernik, S.A. Meguid, Coupling atomistics and continuum in solids: status, prospects, and challenges, International Journal of Mechanics and Materials in Design, 5 (2009) 79-110.

[76] A. Stukowski, Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool, Modelling and Simulation in Materials Science and Engineering, 18 (2010) 015012.

[77] E. Biyikli, Q. Yang, A.C. To, Multiresolution Molecular Mechanics: Dynamics, Computer Methods in Applied Mechanics and Engineering, 274 (2014) 42-55.

[78] Q. Yang, E. Biyikli, A.C. To, Multiresolution molecular mechanics: Convergence and error structure analysis, Computer Methods in Applied Mechanics and Engineering, 269 (2014) 20-45.

[79] Q. Yang, A.C. To, Multiresolution molecular mechanics: A unified and consistent framework for general finite element shape functions, Computer Methods in Applied Mechanics and Engineering, (2014).

[80] H.J. Berendsen, J.P.M. Postma, W.F. van Gunsteren, A. DiNola, J. Haak, Molecular dynamics with coupling to an external bath, The Journal of chemical physics, 81 (1984) 3684-3690.

[81] S. Kwon, Y. Lee, J.Y. Park, D. Sohn, J.H. Lim, S. Im, An efficient three-dimensional adaptive quasicontinuum method using variable-node elements, Journal of Computational Physics, 228 (2009) 4789-4810.

[82] T. Shimokawa, T. Kinari, S. Shintaku, Adaptive Mesh Refinement with Elastic Stiffness Coefficients in the Quasicontinuum Model, Journal of Computational Science and Technology, 3 (2009) 408-416.

[83] R. Tong, G. Liu, L. Liu, L. Wu, Adaptive multiscale method for two-dimensional nanoscale adhesive contacts, Chinese Journal of Mechanical Engineering, 26 (2013) 606-612.

[84] S. Xiao, J. Ni, S. Wang, The bridging domain multiscale method and its high performance computing implementation, Journal of Computational and Theoretical Nanoscience, 5 (2008) 1220-1229.

[85] L. Kalé, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, K. Schulten, NAMD2: greater scalability for parallel molecular dynamics, Journal of Computational Physics, 151 (1999) 283-312.

[86] D.A. Pearlman, D.A. Case, J.W. Caldwell, W.S. Ross, T.E. Cheatham, S. DeBolt, D. Ferguson, G. Seibel, P. Kollman, AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules, Computer Physics Communications, 91 (1995) 1-41.

[87] D. Hull, D.J. Bacon, Introduction to dislocations, Pergamon Press Oxford, 1984.

[88] C.L. Kelchner, S. Plimpton, J. Hamilton, Dislocation nucleation and defect structure during surface indentation, Physical Review B, 58 (1998) 11085.

[89] J.C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R.D. Skeel, L. Kale, K. Schulten, Scalable molecular dynamics with NAMD, Journal of computational chemistry, 26 (2005) 1781-1802.

[90] J.R. Shewchuk, An introduction to the conjugate gradient method without the agonizing pain, in, Carnegie-Mellon University. Department of Computer Science, 1994.

[91] M. Buchholz, H.-J. Bungartz, J. Vrabec, Software design for a highly parallel molecular dynamics simulation framework in chemical engineering, Journal of Computational Science, 2 (2011) 124-129.

[92] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, P. Gumbsch, Structural relaxation made simple, Physical review letters, 97 (2006) 170201.

[93] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, Journal of Computational Physics, 117 (1995) 1-19.

[94] K.J. Bowers, E. Chow, H. Xu, R.O. Dror, M.P. Eastwood, B. Gregersen, J.L. Klepeis, I. Kolossvary, M. Moraes, F.D. Sacerdoti, Scalable algorithms for molecular dynamics simulations on commodity clusters, in: SC 2006 Conference, Proceedings of the ACM/IEEE, IEEE, 2006, pp. 43-43.

[95] H.J. Berendsen, D. van der Spoel, R. van Drunen, GROMACS: A message-passing parallel molecular dynamics implementation, Computer Physics Communications, 91 (1995) 43-56.

[96] G. Anciaux, O. Coulaud, J. Roman, High performance multiscale simulation or crack propagation, in: Parallel Processing Workshops, 2006. ICPP 2006 Workshops. 2006 International Conference on, IEEE, 2006, pp. 8 pp.-480.

[97] B. Fox, P. Liu, C. Lu, H.P. Lee, Parallel multi-scale computation using the message passing interface, in: Parallel Processing Workshops, 2003. Proceedings. 2003 International Conference on, IEEE, 2003, pp. 199-204.

[98] H. Talebi, M. Silani, S.P. Bordas, P. Kerfriden, T. Rabczuk, A computational library for multiscale modeling of material failure, Computational Mechanics, 53 (2014) 1047-1071.

[99] F. Delalondre, C. Smith, M.S. Shephard, Collaborative software infrastructure for adaptive multiple model simulation, Computer Methods in Applied Mechanics and Engineering, 199 (2010) 1352-1370.

[100] C.B. Barber, D.P. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Transactions on Mathematical Software (TOMS), 22 (1996) 469-483.

[101] U. Rencuzogullari, S. Dwardadas, Dynamic adaptation to available resources for parallel computing in an autonomous network of workstations, ACM SIGPLAN Notices, 36 (2001) 72-81.

[102] O. Sigmund, On the usefulness of non-gradient approaches in topology optimization, Structural and Multidisciplinary Optimization, 43 (2011) 589-596.

[103] D. Krause, K. Fackeldey, R. Krause, A Parallel Multiscale Simulation Toolbox for Coupling Molecular Dynamics and Finite Elements, in: M. Griebel (Ed.) Singular Phenomena and Scaling in Mathematical Models, Springer International Publishing, 2014, pp. 327-346.

[104] W. Bangerth, C. Burstedde, T. Heister, M. Kronbichler, Algorithms and data structures for massively parallel generic adaptive finite element codes, ACM Transactions on Mathematical Software (TOMS), 38 (2011) 14.

[105] C. Burstedde, M. Burtscher, O. Ghattas, G. Stadler, T. Tu, L.C. Wilcox, ALPS: A framework for parallel adaptive PDE solution, in: Journal of Physics: Conference Series, IOP Publishing, 2009, pp. 012009.

[106] T. Heister, M. Kronbichler, W. Bangerth, Massively parallel finite element programming, in: Recent Advances in the Message Passing Interface, Springer, 2010, pp. 122-131.