

COARSE-GRAINING ATOMISTIC DYNAMICS OF FRACTURE BY FINITE ELEMENT
METHOD: FORMULATION, PARALLELIZATION AND APPLICATIONS

By

QIAN DENG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2011

©2011 Qian Deng

To my wife and mother

ACKNOWLEDGMENTS

Though only my name appears on the cover of this dissertation, a great many people have contributed to its production. I owe my gratitude to all those people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

My deepest gratitude is to my advisor, Dr. Youping Chen. In the past four years, she has always been there to support and give guidance to me. She taught me how to do research and showed me the frontier of my research field. Most importantly, she taught me the way to explore the unknown by myself. I have learned a great deal from her and I will never forget the valuable lessons she taught me. I hope that one day I would become a scholar as good as Dr. Chen. I would also like to thank Dr. James Lee for his excellent guidance, caring and patience when I was in the George Washington University.

I would like to thank Dr. Loc Vu-Quoc for his valuable discussions with me that helped me understand my research area better. I would like to thank Dr. Bhavani V. Sankar for giving me kind help and advice during the past years. I would also like to thank Dr. Beverly A. Sanders for helping me to develop my background in parallel computing. Special thanks go to Dr. Peter G. Ifju, who was willing to participate in my final defense committee in the last semester.

I would also like to thank my wife and my parents for their encouragement and unwavering support throughout my life. Their love is always providing me inspiration and is my driving force. I could not have come this far without their assistance and I want to express my deepest appreciation to them.

I would also like to thank Dr. Liming Xiong, my lab mate, for his discussions and

cooperation in the development of the computational codes in this work. I would also like to thank all of the other lab mates, Ms. Ning Zhang, Mr. Shi Li, Mr. Shengfeng Yang, Mr. Yongtao Li and Mr. Yu Hong, for providing such a thought-provoking and cooperative atmosphere in our lab. My special thanks also go to Mr. Eddie McCumiskey, my lab mate, for his kind help and valuable suggestions in the past few years.

Finally, the generous financial support from NSF (National Science Foundation), DARPA (Defense Advanced Research Projects Agency) and DOE (Department of Energy) on this project is highly appreciated.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS.....	4
LIST OF TABLES	8
LIST OF FIGURES.....	9
ABSTRACT	11
CHAPTER	
1 MOTIVATIONS.....	13
2 BACKGROUND	18
State of the art of numerical methods for fracture	18
Continuum Methods	18
Atomistic Simulation Methods	22
Multiscale Simulations Methods	24
Sequential multiscale simulations:	24
Concurrent multiscale simulations:	25
Summary.....	27
Introduction to parallel computing	28
What is Parallel Computing?	28
Why Do We Use Parallel Computing Here?	28
High Performance Computing Resources	29
Message Passing Interface (MPI)	30
Running a Parallel Code on High Performance Computers	33
Summary.....	33
3 A COARSE GRAINED METHODOLOGY BASED ON THE ATOMISTIC FIELD THEORY (AFT).....	35
Introduction	35
A brief review of coarse-grained (CG) models	35
An introduction to AFT	38
Decomposition of the Atomic Motion	38
The Local Density and Its Time Derivative	39
Balance Equations	41
Difference between Particles Dynamics and Continuous Field Theories.....	44
The finite element implementation of AFT.....	45
Weak Form of the Governing Equation	45
Numerical Evaluation of Each Term in Weak Form Equation	47
Mesh Strategies.....	51
2D triangular element.....	52
3D hybrid element.....	53

Summary	54
4 PARALLEL IMPLEMENTATION OF CG-AFT METHOD	59
Introduction	59
Four design spaces for the parallel code	59
The structure of the parallel code	62
Domain Decomposition, GETMAP	63
Construct Neighborlist, ELIST	63
Communication Between Processors, EXCHANGE_U	64
Calculation of Nodal Forces, CFORCE	65
Time Integrations, CENTRAL	65
Performance of the parallel code	65
Stability	65
Scalability	66
Summary	69
5 SIMULATIONS OF 2D DYNAMIC BRITTLE FRACTURE.....	76
Introduction	76
The computational model.....	76
Simulation details and results	78
Comparison of CG And MD Simulations.....	78
Large-Scale FE Simulations.....	81
Summary	83
6 SIMULATIONS OF 3D DYNAMIC BRITTLE FRACTURE.....	95
Introduction	95
Comparison of CG and MD simulations.....	96
Application of the CG method in 3D brittle fracture.....	98
3D Crack Branching Under Mixed-Mode Loading	98
Impact Fracture of a Plate	99
3D Crack Surface Evolution	101
Summary	102
7 SUMMARY AND FUTURE WORK.....	113
Summary and conclusion.....	113
Future work.....	115
LIST OF REFERENCES	117
BIOGRAPHICAL SKETCH	123

LIST OF TABLES

<u>Table</u>		<u>page</u>
4-1	Number of elements and processors used in ST and FT.....	70
4-2	FT results (unit: second)	70
4-3	ST results (unit: second).....	70
5-1	Geometric parameters and loading rates for the six models.....	85
5-2	K_{IC} for models A~E	85
5-3	Time interval during which crack branching occurs and time of arrival of reflected stress waves.....	85

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1-1 Crack oscillation and branching under mode I loading	16
1-2 Helical crack-front under mixed-mode (I+III) loading	16
1-3 Fracture at different length scales	16
1-4 Fracture of some single crystals	17
2-1 Cohesive element and bulk element.....	34
3-1 Relationship between AFT and continuum mechanics	56
3-2 Decomposition of atomic positions	56
3-3 Schematic scratch for finite elements in the FE implementation of AFT	56
3-4 Nodal integration and Gauss integration	57
3-5 Two options of the mesh strategy	57
3-6 The triangular element which shows the weakest planes	57
3-7 The primitive unit cell and slip planes of FCC crystal	58
3-8 Sketch of one hybrid element	58
4-1 Structure of the parallel code	71
4-2 Procedures for domain decomposition	71
4-3 Interfaces between domains.....	71
4-4 Communication between neighboring domains	72
4-5 Central difference method	72
4-6 Setup of the model for the force perturbation test.....	73
4-7 Vibration of a FCC bar predicted by the two different methods	73
4-8 Vibration of the FCC bar predicted with varies number of processors	74
4-9 Speedup curves for FT and ST.....	74
4-10 Efficiency curves for FT and ST.....	75

5-1	Simulation Model and Boundary Conditions	86
5-2	The finite element mesh of the computer model.....	86
5-3	Comparison between FE and MD simulations results	87
5-4	Average stress vs. strain curves	88
5-5	Comparison of FE and MD simulations results	88
5-6	Stress distribution in front of crack tip when crack start to propagate	90
5-7	Stress distribution in front of crack tip when crack start to propagate	90
5-8	FE results of dynamic crack propagation.....	91
5-9	Crack speed measured for model E during the period of crack initiation and first branching.....	93
5-10	Stress waves at the time of crack branching in models A-E.	93
6-1	Notched film for the first test case	104
6-2	Comparison of CG and MD results for the first case.....	104
6-3	Comparison of the average stress - time curves	105
6-4	Computation model for 3D crack branching.....	105
6-5	MD simulation of crack branching under mixed-mode loading	106
6-6	CG simulation of crack branching under mixed-mode loading.	106
6-7	Computation model for 3D impact test.....	107
6-8	Fracture of the plate under the impact of a rigid ball.....	108
6-9	Impact test for a smaller plate by MD simulation	108
6-10	Computation model for the bulk material under Mode I loading.....	109
6-11	Crack surface evolution in bulk material	109

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

COARSE-GRAINING ATOMISTIC DYNAMICS OF FRACTURE BY FINITE ELEMENT
METHOD: FORMULATION, PARALLELIZATION AND APPLICATIONS

By

Qian Deng

December 2011

Chair: Youping Chen
Major: Mechanical Engineering

The accurate prediction of material behaviors is one of the most important fields in both science and engineering communities. Understanding the mechanisms of material behaviors sometimes needs us to study the material over a wide span of length scales. In this work we present a new methodology which is able to coarse-grain the atomistic dynamics of fracture by finite element method.

First, based on the Atomistic Field Theory (AFT) (Chen and Lee 2005, Chen et al. 2006, Chen 2009), a finite elements method with built in atomistic information is presented. Then a high efficiency parallel code for large scale computation is described. This code was written in FORTRAN language and uses the standard parallel programming environment message passing interface (MPI). The performance of the parallel code was tested on the supercomputer Trestle of SDSC (San Diego Supercomputer Center). Through the comparison of the coarse-grained (CG) simulation results with the molecular dynamics (MD) simulation results, it is found that the new CG method is able to predict the crack tip stress, dynamic crack propagation and even crack branching, with results similar to that of the atomic-level molecular dynamics

simulations. Finally, both 2D (2 dimensional) and 3D (3 dimensional) dynamic fracture problems were computed through the CG method. In 2D dynamic fracture simulations, the relationship between stress waves and crack propagations was studied. It is found that the stress waves reflected back from the boundary can trigger the dynamic crack branching. In 3D simulations, the dynamic fractures under different loading were simulated. The largest 3D model is composed of over 0.1 million elements which are equivalent to over 0.1 billion atoms. To show the performance of the parallel code in dealing with large number of processors, the crack surface evolution in this model was simulated using 512 processors. All of simulations conducted in this study show the robustness of the parallel code in dealing with dynamic fracture problems.

CHAPTER 1 MOTIVATIONS

Fracture is the (local) separation of an object or material into two or more pieces when sufficient work is applied on the atomic level to break the bonds that hold atoms together (Anderson, 1991). The bond strength is supplied by the interactions between the electrons of atoms. Dynamic fracture problems are those where the inertia forces, rate-dependent material behavior, and reflected stress waves can significantly affect the fracture behavior. Since bond rupture at the atomic level is a dynamic process, all fracture is dynamic at the atomic scale. The dynamic fracture problem is the most fundamental in the science of fracture (Cox et al. 2005).

The instability of crack propagation is one of the most important phenomena in dynamic fracture. Experiments have demonstrated unstable crack propagation (crack branching) in metallic glass. It is found that the dynamic cracks will oscillate and branch when they become unstable (Flores and Dauskardt 1999). Although the theoretical researches have shown that the crack propagation under Mode I loading could become unstable when its speed reaches a critical value (Yoffe 1951, Freund 1974, 1998, Gao 1994), the mechanisms for dynamic fracture are still far from well understood. The origin of unstable crack propagation and how a crack will behave when it becomes unstable are still open topics currently.

Under complex boundary conditions, analytical prediction of the dynamic crack propagation is too difficult to achieve. Although the crack surface under mixed-mode loading has been the focus of the community of dynamic fracture for a long time, because of the complexity of the problem, few successful works in this field have been reported. Figure 1-1 shows the unstable crack propagation, or crack branching, under

the Mode I loading (Flores and Dauskardt 1999). Recent research (Figure 1-2) shows that the crack propagation will become universally unstable under a mixed-mode (I+III) loading (Pons and Kamar 2010). The mechanical origin of this helical crack-front instability is still poorly understood.

As shown in Figure 1-3, dynamic fracture is a multiscale phenomenon which concurrently involves the breakdown of material on different length scales. To understand the origins and mechanisms of dynamic fracture, it is necessary to investigate the failure of materials on different length scales. At the nanoscale or even mesoscale, the atomic structure of material plays an important role in its fracture behavior.

In single crystals, some planes are naturally weaker than others. Experiments show that the fracture in single crystals always happens on these weaker planes. As shown in Figure 1-4, the fracture surfaces of four different single crystals are not smooth when observed at the micro scale. The staircase fracture surfaces show the easiest fracture planes in the material. In Figure 1-4 (a) (Pashley 1960), the fracture surfaces along $\{1\ 1\ 1\}$ planes are caused by the dislocations in a gold thin film, a face center cubic (FCC) single crystal. In Figure 1-4 (b) ~ (d) (Schultz et al. 1994), the fracture surfaces are the brittle cleavage planes of three different brittle single crystals, NaCl, CaCO₃, and CaF₂.

With the development of large scale computing techniques, numerical methods have become some of the most important and efficient ways to study the mechanisms for fracture. During the past few decades, a large number of numerical methods have been proposed to simulate fracture. The continuum mechanics based methods, such as:

extended finite element method (XFEM), cohesive zone model (CZM), and peridynamics, have been widely used to simulate the crack propagation problems without taking into account the atomistic information of the material. As one of the most popular approaches for atomistic simulation, molecular dynamics (MD) simulation is useful in simulating dynamic crack propagation. There are also many multiscale models which aim to simulate systems with larger sizes relative to MD simulations. However, all the above methods have their limitations. Currently, there are no successful numerical methods for dynamic fracture. In this work, a more prospective method is presented and used for the study of dynamic fracture problems.

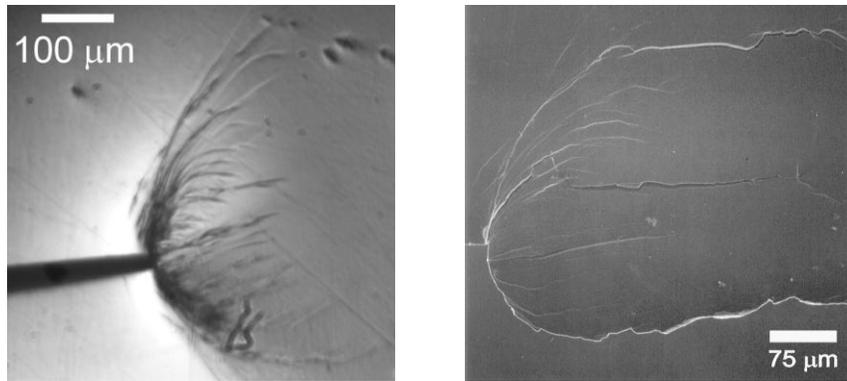


Figure 1-1. Crack oscillation and branching under mode I loading (Flores and Dauskardt 1999)

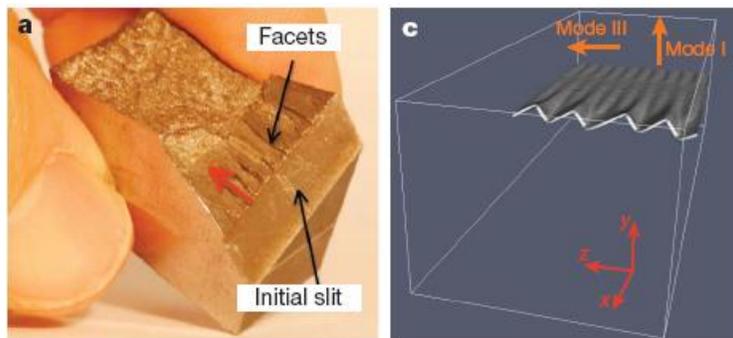


Figure 1-2. Helical crack-front under mixed-mode (I+III) loading (Pons and Kamar 2010)

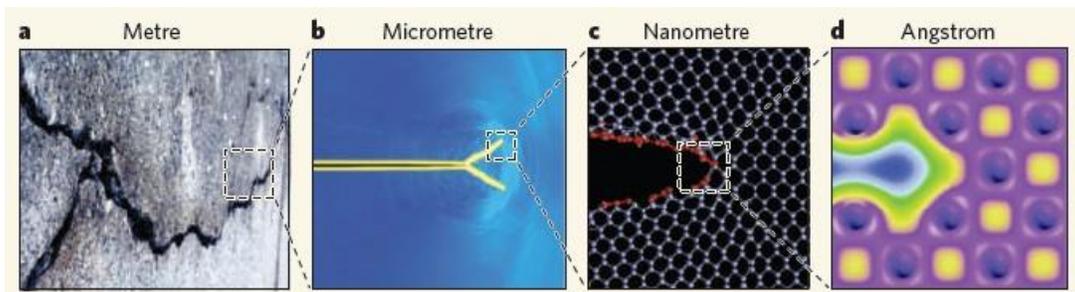
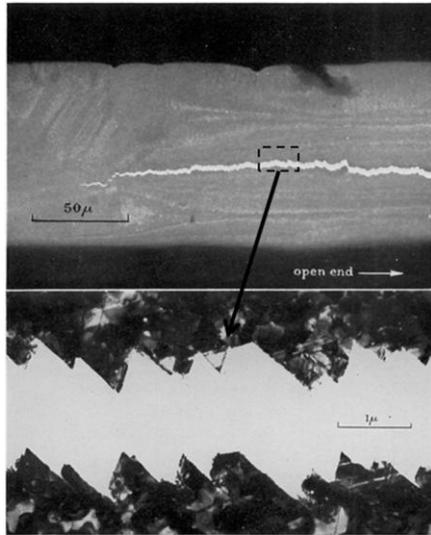
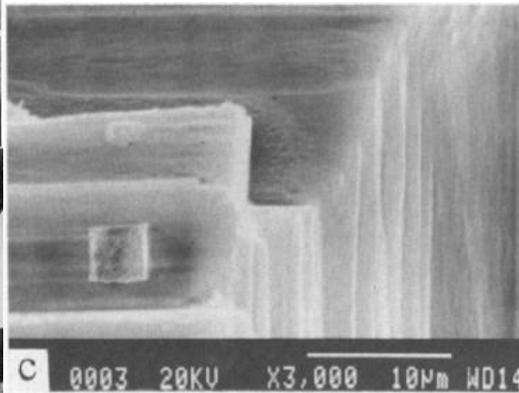


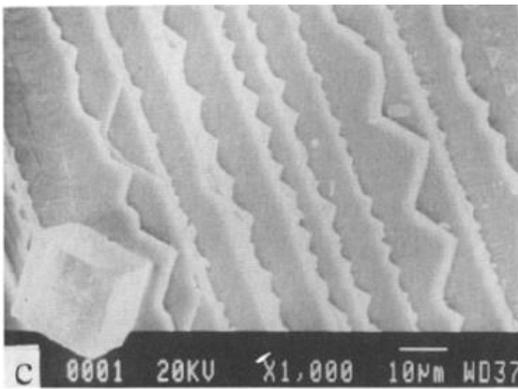
Figure 1-3. Fracture at different length scales (Markus 2010)



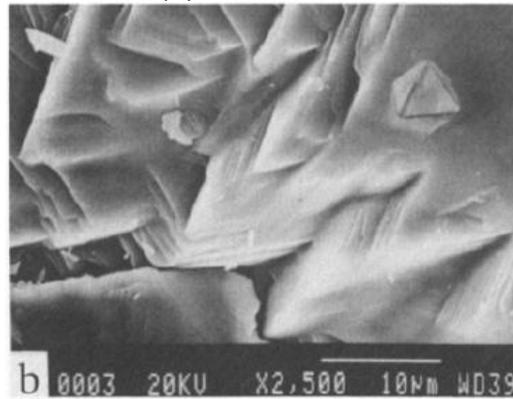
(a) gold



(b) NaCl



(c) CaCO₃



(d) CaF₂

Figure 1-4. Fracture of some single crystals (Schultz et al. 1994)

CHAPTER 2 BACKGROUND

State of the art of numerical methods for fracture

In this work, a new method for coarse graining (simulating a system using fewer degrees of freedom than they actually have) the atomistic dynamics of fracture is presented. Since there have already been a lot of numerical models for modeling fracture, why do we need another one? To answer this question, in this section, a brief review on the state of the art and the challenges of current numerical methods for dynamic fracture is made.

Continuum Methods

Historically, many continuum methods based on continuum mechanics have been successfully applied to study fracture at macroscopic scale. Continuum mechanics is a branch of physical sciences concerned with the deformation and the motion of continuous material media under the influence of external effects. External effects that influence bodies appear in the form of forces, displacements and velocities that arise from contact with other bodies, gravitational forces, thermal changes, chemical interactions, electromagnetic effects and other environmental changes. The theory of continuous media is built upon four basic laws: (1) the conservation of mass, (2) the balance of linear momentum, (3) the balance of angular momentum, and (4) the conservation of energy. They are universal and are results from our experiences with the physical world. The constitutive relations are constructed to take the nature of different materials into consideration. These relations should not violate the second law of thermodynamics, which is the famous Clausius-Duhem inequality in continuum mechanics.

In the past few decades, classical continuum mechanics has provided most of the theoretical and computational tool for understanding fracture. However, the mathematical framework that has been developed for continuum mechanics is not well suited to simulate such problems as crack propagation, since the balance equations of continuum mechanics are partial differential equations, and the presence of the spatial derivatives require continuity of the material. Therefore, cracks within a material have to be treated as surface boundaries of the material.

Many numerical methods have thus been proposed to overcome this barrier in the simulation of fracture. Successful techniques include the extended finite element method (XFEM) (Fries and Belytschko 2006, Huynh and Belytschko, 2009), in which a standard displacement-based approximation is enriched by incorporating discontinuous fields. The most significant character of XFEM is that discontinuity can exist within elements. The governing equation for XFEM is just the weak form of the traditional balance equations. To allow the representation of crack discontinuities and voids independent of the mesh, the traditional shape function is modified as follows:

$$\mathbf{u}^h(\mathbf{x}) = \sum_i \mathbf{u}_i \phi_i(\mathbf{x}) + \sum_j \mathbf{b}_j \phi_j(\mathbf{x}) H(f(\mathbf{x})) + \sum_k \phi_k(\mathbf{x}) \left[\sum_l \mathbf{c}_k^l F_l(\mathbf{x}) \right] \quad (2-1)$$

where $\phi_i(\mathbf{x})$ are the traditional shape functions, \mathbf{u}_i is the displacement at node i , $H(\mathbf{x})$ is step function which introduces the discontinuity, $F_l(\mathbf{x})$ are the functions which introduce singularity, \mathbf{b}_j and \mathbf{c}_k^l are variables caused by the introduction of discontinuity and singularity. This method is static and cannot model dynamic fracture. The crack propagation and crack path depend on the empirical criteria which should be given before the computation.

Another well-recognized method in fracture modeling and simulation is the

cohesive zone model (CZM) (Needleman 1987, Camacho and Ortiz 1996, Remmers et al 2003, Park et al. 2009). The advantage of CZM is its ability in modeling arbitrary dynamic crack propagation. By inserting interface elements between continuum finite elements along the potential crack path, the cohesive crack propagations can be modeled. The governing equation for this method is as following:

$$\int_V \rho \ddot{\mathbf{u}} \cdot \delta \mathbf{u} dV + \int_V \mathbf{P} : \delta \mathbf{F} dV = \int_{S_{ext}} \bar{\mathbf{T}} \cdot \delta \mathbf{u} dS + \int_{S_{int}} \mathbf{T} \cdot \delta \Delta dS \quad (2-2)$$

where \mathbf{u} is the displacement, \mathbf{P} is the first Piola-Kirchhoff stress, which is an energy conjugate of deformation gradient \mathbf{F} , $\bar{\mathbf{T}}$ is the surface traction on external boundary S_{ext} , \mathbf{T} is the internal traction on internal boundary S_{int} , and Δ is the separation of the internal boundary.

As shown in Figure 2-1, two kinds of element, a bulk element and an interfacial element (cohesive element), are used this model. Discontinuities are introduced by the failure of the cohesive element. It is worthwhile to mention that the cohesive element is infinitely thin in its undeformed stage. So it is very difficult to define deformation (or strain) for this element. Actually, the constitutive relation for the cohesive element is described as a traction-separation relation. The constitutive relations for the interfacial element and the bulk element can be expressed in the following forms:

$$\begin{aligned} \text{bulk element:} \quad \mathbf{P}(\mathbf{F}) &= \frac{\partial W(\mathbf{F})}{\partial \mathbf{F}} \\ \text{interfacial element:} \quad \mathbf{T}(\Delta) &= \frac{\partial \phi(\Delta)}{\partial \Delta} \end{aligned} \quad (2-3)$$

where $W(\mathbf{F})$ and $\phi(\Delta)$ are potential energy density stored in bulk element and interfacial element, respectively. It is obvious that the constitutive descriptions of these two elements are different. This inconsistency causes the difference of the mechanical behaviors of these two kinds of element.

Some laterdeveloped continuum theories have been trying to extend continuum mechanics to more fields of application, such as microscale. Micromorphic theory (Eringen and Suhubi 1964) is a continuum field theory which considers a material body as a continuous collection of deformable particles. It is an attempt to associate the microstructure of a material to its macroscale behaviors under the framework of continuum mechanics. Another interesting attempt, nonlocal continuum theory (Eringen and Edelen 1972, and Eringen 1972a,b), is concerned with the physics of material bodies whose behavior at a point in a body material is influenced by the state of other points in the body.

It is worthwhile to mention another attempt, peridynamics theory, to improve the situation through reformulating elasticity theory. The peridynamics theory, proposed and developed by Dr. Silling (Silling, 2000), is a nonlocal continuum theory. Compared with traditional nonlocal continuum mechanics, the primary advantage of the Peridynamics approach for fracture modeling is that it does not require any supplemental relation to dictate crack initiation, propagation velocity and direction, crack branching, and crack arrest. All of these phenomena emerge as the consequences of the equations of motion and the constitutive model. Like MD simulations, it does not need any special numerical treatment for modeling of arbitrary crack growth. The advantage of peridynamics in simulating discontinuities has been well demonstrated through simulations of the fragmentation process of brittle materials (Silling and Askari 2005). However, peridynamics simulations are always limited by the accurate determination of material constants. The physics of these parameters and how to measure them in experiments are still challenges currently.

Despite their tremendous success, continuum methods have their limitations as well. One main limitation is the complete neglect of atomistic mechanisms which are sometimes of interest in fracture problems. Another limitation is that the constitutive models are all empirically derived from macroscale experiments. For these reasons, continuum mechanics is only suitable for macro- and micro-scales.

Atomistic Simulation Methods

There is little doubt that most of the low-energy physics, chemistry, materials science, and biology can be explained by the quantum mechanics (QM) of electrons and ions and that, in many cases, the properties and the behaviors of materials derived from the quantum mechanical description of events at the atomic scale (Ghoniem et al. 2003). However, solving the Schrödinger Equations for a large system is very computationally expensive. To determine the properties of an ensemble of atoms larger than that which can be handled by computational QM, molecular dynamics (MD) simulation is one of the most popular ways. With the advent of high-performance computers, atomic-level molecular dynamics simulation has become a powerful tool to study fracture at the atomic scale (Holian and Ravelo 1996, Zhou et al. 1996, Abraham et al. 1994, Abraham 1996, Abraham and Gao, 2000, Buehler et al. 2003, and Buehler and Gao 2006).

The basic idea of MD is using empirical, effective interatomic potential energy functions to eliminate all electronic degrees of freedom. These potentials are represented by functions that depend on the atomic configuration (i.e. relative displacement) and the local environment (i.e. electrons). In general, the total effective potential of the whole system is

$$\begin{aligned}
E &= \phi_2 + \phi_3 + \dots + \phi_N \\
&= \sum_{i,j} V_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k} V_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \sum_{i,j,k,l} V_4(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \mathbf{r}_l) + \dots + \sum_{i,j,k,\dots} V_N(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \dots)
\end{aligned} \tag{2-4}$$

where $V_2(r_i, r_j)$, $V_3(r_i, r_j, r_k)$, $V_4(r_i, r_j, r_k, r_l)$ and $V_N(r_i, r_j, r_k, \dots)$ correspond to two-body interactions, three-body interactions, four-body interactions, and N-body interactions, respectively. This expression converges relatively quickly. In practice, only keeping the first 3 terms is sufficient for the desired accuracy. The formulations for the effective potential serve as the constitutive equation in molecular dynamics simulation.

The dynamic evolution of the system is governed by classical Newtonian mechanics. For each atom i the equation of motion is given by:

$$m_i \frac{d^2 r_i}{dt^2} = F_i = -\nabla_{r_i} E \tag{2-5}$$

which is derived from the classical Hamiltonian of the whole system:

$$H = \sum_{i=1,N} \frac{1}{2} m_i v_i^2 + E \tag{2-6}$$

In addition to the unique advantage that MD simulation can provide with regard to the atomic scale mechanism of fracture, it possesses no barrier in modeling and simulation of arbitrary crack growth, branching, deflection, and arrest. Different from continuum mechanics, the governing equation for MD simulation treats discrete atoms in the ensemble, so the variables do not need to be continuous in space (they are continuous in time). In this sense, MD simulation is very suitable for simulating the problems with space discontinuities, such as crack propagation.

The main limitation of current MD simulation is the spacial and temporal size achievable. Current state-of-the-art supercomputer MD simulations can handle several billions of atoms, amounting to a volume of less than a cubic micron. There is still a gap between experiments and MD simulations.

Multiscale Simulations Methods

Motivated by the limitation of MD simulations, tremendous efforts have been made in developing some methods that are both efficient and accurate. These approaches are called multiscale methods. Usually, multiscale methods are classified into sequential methods and concurrent methods. Sequential methods (also called hierarchical methods) attempt to piece together a hierarchy of computational approaches for different length scales. In this methodology, the fine-scale model (atomistic model) and coarse-scale (coarse grained model) model are weakly coupled and run sequentially. The parameters (or information) obtained from the fine-scale are passed to as an input of the coarse-scale. Concurrent methods are those in which the fine-scale model is embedded and strongly coupled with the coarse-scale model; the fine-scale model and coarse-scale model are always run concurrently. In concurrent simulations, the system is often partitioned into domains characterized by different scales.

Sequential multiscale simulations:

For the applications in different fields, a large number of sequential multiscale models have been developed. Here we only concentrate on those approaches for fracture problems. One of the most significant approaches of sequential multiscale methods aims to simulate the fracture dynamics, which couple the atomic scale and macroscale, is proposed by Yamakov and colleagues (Yamakov et al. 2006). In their work, crack propagation at the grain boundary of two aluminum crystals was simulated by a cohesive zone model and a traction-separation law is extracted from MD simulation results. It is worthwhile to note that this methodology is more suitable for determining the elastic properties of materials. However, since the descriptions of fracture at the atomic scale and macroscale are quite different, the transferability of the atomistically

fitted decohesion law is very poor. Recent research shows that cohesive laws of interfaces are not sufficient for modeling fracture problems using the cohesive zone model. (Coffman et al. 2008)

Concurrent multiscale simulations:

Fracture dynamics is one of the most challenging problems in material science and solid mechanics because fracture phenomena are governed by processes occurring over a wide range of length scales. In particular, the physics on different length scales interacts dynamically. Therefore, the concurrent multiscale methods with strongly coupled fine and coarse scales are adequate for the study of fracture dynamics. This category of methodologies in which certain key regions are treated with fine-scale models while most of the domain is treated with a coarse-scale model is called partition-domain methods.

In the past decades, a number of concurrent multiscale methods have been proposed. The Macroscopic Atomistic Ab Initio Dynamics (MADD) method proposed by Abraham, Broughton, and their colleagues (Abraham et al. 1998, Broughton et al. 1999) aims at linking the length scales ranging from the atomic scale through the microscale. In their approach, the crack tip region is treated with a quantum-mechanical tight-binding approximation method; then molecular dynamics simulation is used for the region around the quantum region; the region far away from the crack tip is treated via the finite element method in the context of continuum elasticity. Handshaking regions, the regions between the atomic and the continuum regions, are used to link each two neighboring regions. In handshaking regions, the governing equations in both of the two regions contribute to the displacement field. As shown in the governing equation for this approach

$$\begin{aligned}
H_{Tot} = & H_{FE}(\{\mathbf{u}, \dot{\mathbf{u}}\} \in FE) + H_{FE/MD}(\{\mathbf{u}, \dot{\mathbf{u}}, \mathbf{r}, \dot{\mathbf{r}}\} \in FE / MD) \\
& + H_{MD}(\{\mathbf{r}, \dot{\mathbf{r}}\} \in MD) + H_{MD/TB}(\{\mathbf{r}, \dot{\mathbf{r}}\} \in MD / TB) \\
& + H_{TB}(\{\mathbf{r}, \dot{\mathbf{r}}\} \in MD / TB)
\end{aligned} \tag{2-7}$$

the region FE/MD is the handshaking region for FE and MD, the region MD/TB is the handshaking region for MD and TB. Actually, MD and TB regions are fully overlapped which means that the TB region is a subset of the MD region.

A primary concern in this method is the seamless coupling of force and displacement at the interface between the two regions. The inherent mismatch of the description between different length scales (such as atomistic and continuum scales) can lead to various numerical and theoretical difficulties. For example, the description of continuum regions and atomic regions are different; this difference prevents the propagation of high frequency waves from atomic to continuum regions. Instead, high frequency waves are reflected and can melt the atomic region.

Another kind of concurrent model is based on the Cauchy-Born rule, or approximation, which states that in a crystalline solid subjected to a small strain, the positions of the atoms within the crystal lattice follow the overall strain of the medium. Quasicontinuum (QC) is a famous example in this kind of model. QC was firstly proposed by Tadmor, Ortiz and Phillips (Tadmor, Ortiz and Phillips 1996) and has been used to model the static fracture of nickel (Miller et al. 1998). In their QC models, an atomic region (fine region) is concurrently linked to a coarse region. Energy minimization is used for the whole specimen to determine the displacements of the atoms in the fine region as well as the element nodes in the coarse region. In the coarse region, each element node is treated as a 'repatom', which represents a group of atoms. When the element size is reduced to an atomic size, there is no difference between

nodes and atoms. Motivated by this approach, Xiao and Belytsko (Xiao and Belytsko 2004) studied the static fracture of a graphene sheet through a proposed bridging domain method (BDM) which links the molecular mechanics region and the QC region by using Lagrangian multipliers. Because they lack the balance equations for energy, the aforementioned approaches, as a series of energy minimization methods, can only simulate static, zero-temperature problems.

There are two common pitfalls in all kinds of partition-domain methods. (1) Only static and zero temperature problems can be investigated. (2) The critical phenomenon should be confined in the fine-scale region and other parts of the specimen only serve as boundary conditions. So these methods are not suitable for the investigation of dynamic fracture.

Summary

In this section, the state-of-the-art of numerical methods for fracture is reviewed. From the review, it is seen that the modeling of dynamic fracture is still a challenging topic. A new model which can take into account not only the large-scale fracture dynamics but also the atomistic mechanism is required. Most of current models only meet one of the two requirements. The goal of multiscale methods is to overcome this difficulty. However, both sequential and concurrent multiscale methods still have their own difficulties currently. One of the most significant barriers to achieving this goal is the inconsistency of the descriptions between atomic and continuous regions.

The recently developed atomistic field theory (AFT) (Chen and Lee 2005, Chen 2006, 2009) provides a way to achieve this goal. In this work, a new coarse-grained model based on AFT is introduced and used to deal with the dynamic fracture problem. Some details of this method will be presented later in Chapter 3. We will also show that

the atomistic dynamics of fracture can be treated accurately with a coarse-grained model.

Introduction to parallel computing

What is Parallel Computing?

Parallel computing (from Wikipedia) is a 'form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently' ("in parallel"). If a computational problem has exploitable concurrency, then this problem can be decomposed into several parallel sub-problems. These sub-problems can be executed separately. By using multiple processors to compute these sub-problems concurrently, we can obtain the solution faster than only using one processor for the computation of the whole problem. Also, if each processor has its own memory, partitioning the data between the processors may allow larger problems to be handled faster than could be handled on a single processor.

Why Do We Use Parallel Computing Here?

Scientific computations are costly in two domains: the computational time and the memory. These barriers can be overcome by using parallel computing. With multiple processors working together, we can always compute larger systems in shorter time. Two of the successful examples are the applications of parallel computing in molecular dynamics and finite element method. Most of the popular MD computation software, such as AMBER, LAMMPS, DL_POLY, etc., have been well parallelized. For those well-known finite element method packages, such as ANSYS, ABAQUS, MSC/NASTRAN, etc., parallel performance is also an important standard for the evaluation of these products. The better the parallel performance is, the larger the system that can be

computed with a fixed amount of processors.

In the current work, a field theory is implemented by finite element method. To simulate large systems, a parallelization for the finite element method is conducted. Then the parallel code is used to investigate dynamic crack propagation in crystals. Since dynamic crack propagation is a multiscale phenomenon which involves local material behavior, the mesh for the computed model should be fine enough to capture the local information. Since a large amount of memory and computational time is required for the computation of the model, parallel implementation is necessary for the current work.

Another reason for using parallel computing here is the high concurrency of the problem. For the current method, over 95% of the computations time is taken by the calculation of the internal forces in the finite elements. Using the explicit time integrate strategy, the calculation of the internal force on each element node can be solved independently. This means that the step for internal force computation can be parallelized with high efficiency. So the current method is very suitable for parallel computing.

High Performance Computing Resources

Currently, there are thousands of high performance computational resources all over the world. Recently, the world's top performance supercomputer changes about every six months. In June of 2010, the number one supercomputer was JAGUAR, located at Oak Ridge National Laboratory in U.S. Five months later, TIANHE-A1 in China overtook it. By June of 2011, the top ranking supercomputer was changed to K supercomputer in Japan. The fast development of supercomputers indicates that high performance computing is being more and more important in research and industry.

For our current research, the available high performance computing resources include the high performance computer (HPC) center in University of Florida (UF) and TeraGrid. The HPC of UF runs several clusters which is open to the faculties and students of UF. This Center runs several clusters with over 3,000 cores in dual-core, quad-core, and hex-core servers. Most of the servers are part of one of two distinct InfiniBand fabrics. The clusters share over 120 terabytes of distributed storage via the Lustre parallel file system. The performance of the HPC in UF is about 15 teraflops. One teraflop means the ability to do 1×10^{12} floating-point operations per second. Each dual-core, quad-core, or hex-core server has its own memory from 4GB to 64GB.

TeraGrid is an open scientific discovery infrastructure combining leadership class resources at eleven partner sites to create an integrated, persistent computational resource. Using high-performance network connections, TeraGrid integrates high-performance computers, data resources and tools, and high-end experimental facilities around the country. Currently, TeraGrid resources include more than 2 petaflops of computing capability and more than 50 petabytes of online and archival data storage, with rapid access and retrieval over high-performance networks. One petaflops means the ability to do 1×10^{15} floating-point operations per second. Researchers can also access more than 100 discipline-specific databases. With this combination of resources, TeraGrid is the world's largest, most comprehensive distributed cyber infrastructure for open scientific research.

Message Passing Interface (MPI)

During parallel computation, it is always necessary for processors to communicate with each other for sharing their computation results dynamically. MPI, a parallel

programming environment, provides a convenient way to achieve this goal. MPI, created in the early 1990s, is the standard programming environment for distributed-memory parallel computers. The standard defines the syntax and semantics of a core of library routines used for writing portable message-passing codes in Fortran or C programming languages.

Currently, there are several implementations of MPI, such as OpenMPI, MPICH, HPMPPI, IntelMPI, and Microsoft MPI. Among these implementations, OpenMPI and MPICH are the most popular. Each implementation includes some compilers, which are used to compile the source code in different languages. For example, in OpenMPI, mpicc is the compiler for the code written with C; mpif90 is the compiler for the code written in Fortran 90; mpif90c++ is the compiler for the code written in C++. An implementation is a package installed in a system and a compiler is used for compiling a source code written by the user. The executable generated by the compiler can be run by using the command “mpirun” or “mpiexec”.

There are some important concepts introduced in the first version of MPI, such as environment management, point-to-point communication, collective communications and communicator. For the second version, MPI-2, some extensions, including parallel I/O, one side operation, and dynamic process management, are provided. These concepts are closely related to the application of the MPI standard.

A portable code written with MPI must be compatible for different systems. To achieve this portability, a subroutine called MPI_INIT(), which is used for the setup of the parallel environment, needs to be called before any other MPI routines. At the end of the parallel code, another subroutine, MPI_FINALIZE(), is used to clean up all MPI

states.

Point-to-point communication, the transmittal of data from one process to another, is the basic communication mechanism of MPI. In this type of communication, a SEND operation is always used combined with a RECV operation. There are two kinds of point-to-point communication modes: blocking and unblocking. For blocking mode, the SEND and RECV operations are always finished at the same time. For unblocking mode, one of these two operations could be finished first. Then a subroutine, MPI_WAIT(), is used before the next communication to avoid conflict in data transmittal.

Collective communications provide a convenient way to transmit data among a group of processes. The functions of collective communications include: synchronization, broadcasting, gathering, scattering, and reduction. From our own experiences, it is found that the collective communications always take a long time when the number of processes is large.

A communicator is an object that specifies a communication domain for process groups. It includes the management of groups, the communications within or between groups. At program startup, the runtime system creates a common communicator called MPI_COMM_WORLD.

MPI-2 provides subroutines for transferring data to or from files on an external storage in a high performance parallel way. The files can be randomly or sequentially accessed using the parallel I/O. For the sequential mode, the file pointer for read and write should be shared among all the processes in the communicator's group.

Regular send/receive communication requires matching operations by both the sender and the receiver. Remote Memory Access (RMA), an extension in MPI-2, allows

one process to specify all communication parameters, both for the sending side and for the receiving side. This communication mechanism provides more flexibility for the parallel coding.

The MPI-2 process model allows for the creation and cooperative termination of processes after an MPI application has started. A process can start new processes and establish communication for them by calling the subroutines `MPI_COMM Spawn()` and `MPI_COMM Spawn_Multiple()`. The new processes will be cooperatively terminated if the subroutine `MPI_Init()` is called.

Running a Parallel Code on High Performance Computers

Running a parallel code may be different depending on the systems. However, all the systems have a similar structure, which is composed of submit nodes, computing nodes, and visualization nodes. Submit nodes are for submitting jobs, compiling codes, and file transferring. Computing nodes, being the largest part of the system, are used for running jobs. Visualization nodes are often used to visualize the results, which could contain an extremely large amount of data.

After an executable is generated, a script file is used for submitting the computation job. In general, the script file should specify the characters of the job, such as: the name and the path of the executable, the path for the I/O, the number of cores to be used, and the memory required. Then the job is submitted using the command “`submit scrip-file-name`”.

Summary

With the fast development of high performance computing technology, parallel computing is being widely applied in the field of scientific computation. This technology allows us to investigate the real mechanisms more accurately and reasonably with the

aid of computer. MPI is a parallel programming environment that provides a convenient way for the implementation of parallel code. Our current high performance computing resources include the HPC in UF and the TeraGrid. In this work, a high performance parallel code is developed using the programming language FORTRON in the MPI environment.

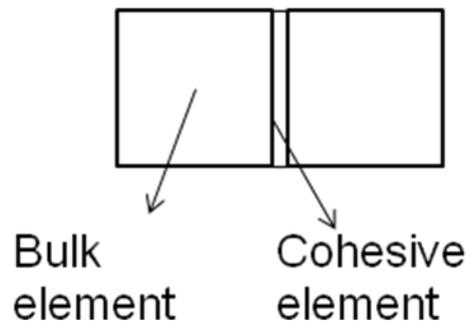


Figure 2-1. Cohesive element and bulk element

CHAPTER 3 A COARSE GRAINED METHODOLOGY BASED ON THE ATOMISTIC FIELD THEORY (AFT)

Introduction

In this chapter, a newly developed predictive coarse grained method based on the Atomistic Field Theory (AFT) (Chen and Lee 2005; Chen 2009) is introduced. AFT is a continuous field theory derived solely from atomistic model. This theory provides a new framework for coarse-grained and multiscale simulations. Within this framework, both fine- and coarse- regions are treated by the same set of governing equations which will be introduced in some details later. Because of the consistency of the description, the transition from fine- to coarse- regions becomes natural. The finite element implementation of AFT leads to a new CG method. The key feature that makes this method different from the conventional finite element method is the construction of the finite element. For the new CG method based on AFT, the finite element mesh design is closely related to the lattice structure of the material. Two different finite elements corresponding to 2D triangular lattice and 3D face center cubic (FCC) crystal are introduced. The internal force field is directly derived from the interatomic potential. Both node integration and Gauss integration strategies for the calculation of nodal force are presented in this chapter.

A brief review of coarse-grained (CG) models

In general, CG models indicate those models with fewer degrees of freedom than they actually have (Español 2004). It is well accepted that MD simulations can predict the properties and behaviors of various systems at atomic level from first principals. However, the limitation of MD is also obvious. It is limited by the spatial and temporal size of the simulation system. So CG models are necessary for the systems larger than

that can be handled by MD simulation.

One of the most intuitive ways to construct a CG model is to treat a group of atoms as a rigid particle. Then an effective force field for the CG particles is used instead of the interatomic potentials. By grouping atoms together, the total number of degrees of freedom can be significantly reduced. There are two key concerns that affect the accuracy of a CG model: (1) the selection of the center of CG particle, and (2) the construction of the effective force field. There are many ways, or criteria, proposed to deal with these two concerns. For example, minimizing the error of distributions and thermodynamic properties (Soper 1996, 1997, Lyubartsev and Laaksonen 1995, 1999); or using the force matching (FM) approaches (Izvekov et al. 2004, Izvekov and Voth 2005, Hone et al. 2005). These models have been widely used in the field of macro molecular soft materials where some of the interatomic bonds are much stronger than others. However, since the deformation of each CG particle is constrained, this class of models is not suitable for crystalline material where all the bonds are at the same level.

Based on the assumption of Cauchy-Born rule, there is another class of CG methods which is suitable for crystalline material. In these models, only a small portion of atoms in the system are selected to represent the whole system. The selected atoms are called repatoms (representative atoms). These repatoms determine the displacement field in the system. The motions of other atoms are approximated by some collective motions. As is mentioned in Chapter 2, quasicontinuum (QC) method falls in the scope of this category.

All of the CG models reviewed above are particle based models whose theoretical framework is discrete particle dynamics. There is another class of CG models based on

field representations. Classical continuum mechanics, as one of the earliest coarse-grained field theories, view a material as a homogeneous and continuous medium. The basic or the smallest structural unit is idealized as a point mass without structure. All the quantities in the system can be described by field functions. These density functions are related to each other by governing equations. The expressions for these functions can be found by solving the governing equations analytically or numerically. In continuum mechanics, the atomistic information, such as the atomic structure and the internal deformations, of materials are ignored. So the description of materials is based on the macroscopic, homogeneous motion of the material particles. For this reason, the material properties involved in continuum mechanics cannot be linked to the motion of atoms at nanoscale. The most common way to obtain these material properties is from macroscopic experiments. In other words, this model is not predictive.

The fundamental goal of statistical mechanics is to link the microscopic many-body dynamics to the phenomenological averaged continuum description. Equilibrium classical statistic mechanics provides a successful link between the atomic motion and the static, equilibrium macroscopic variables (Gibbs 1902). The nonequilibrium statistical mechanics extends this link to dynamic situations. The most significant works in the field of nonequilibrium statistical mechanics are a series of articles of Kirkwood and his cooperators (Kirkwood 1946, 1947, Kirkwood et al. 1949, Irving and Kirkwood 1950, Irving and Zwanzig 1951, Zwanzig et al. 1953, Zwanzig et al. 1954, Ross and Kirkwood 1954). In their works, the equation of continuity, the equation of motion, and the equation of energy transport are expressed in terms of molecular variables. However, classical statistical mechanics has only worked for macroscopic properties

from the simplest lattice systems. Hardy's work in 1982 (Hardy 1982) is an extension of Kirkwood and his cooperators' theory. With the new formulas, some difficulties arising in the old theory can be avoided. They make the old theory easier to use in the practice.

An introduction to AFT

Decomposition of the Atomic Motion

Atomistic Field Theory is a generalized continuum theory. Different from traditional continuum mechanics, in AFT the atomic information is associated with each material point. Figure 3-1 shows the relationship between traditional continuum mechanics, micromorphic theory and AFT. In classical continuum mechanics, each material point is treated as an infinitesimal point, the degree of freedom for this point is 3. In the later developed micromorphic theory, the material point is not longer an infinitesimal rigid point but a deformable particle. This particle has nine independent degrees of freedom describing both stretches and rotations, in addition to the three classical translation degrees of freedom. The recent developed AFT views a crystalline material as a continuous collection of lattice points, while embedded within each point is a group of discrete atoms. Thus the degree of freedom for each lattice point depends on the number of atoms associated to this point.

It is worthwhile to mention that the governing equations for AFT could be based on time-averaged quantities or instantaneous quantities. The governing equations based on time-averaged quantities are derived by Chen and her cooperators in 2005 (Chen and Lee 2005) and the governing equations based on instantaneous quantities are derived by Chen in 2009 (Chen 2009). It is also interesting to find that both the time-averaged quantities and the instantaneous quantities are following the same set of governing equations. In this dissertation, the instantaneous version is chosen for the

introduction of AFT.

One of the key issues of AFT is the decomposition strategy which allows the concurrent treatment of the crystalline material at different length scales. As shown in Figure 3-2, the atomic position is decomposed into lattice position and the relative internal position:

$$\mathbf{R}^{k\alpha} = \mathbf{R}^k + \Delta\mathbf{r}^{k\alpha} \quad (3-1)$$

where $\mathbf{R}^{k\alpha}$ is the position of the α -th atom in the k -th unit cell, \mathbf{R}^k is the position of the k -th unit cell, and $\Delta\mathbf{r}^{k\alpha}$ is the relative position between the α -th atom and the centre of the unit cell. The correspondence of this decomposition in the physical space is:

$$\mathbf{z} = \mathbf{x} + \mathbf{y} \quad (3-2)$$

where \mathbf{z} , \mathbf{x} , \mathbf{y} are the counterparts of $\mathbf{R}^{k\alpha}$, \mathbf{R}^k and $\Delta\mathbf{r}^{k\alpha}$, respectively.

The Local Density and Its Time Derivative

The link between the local density function in physical space and any measurable dynamics function $A(\mathbf{r}, \mathbf{p})$ in phase space is established as

$$\begin{aligned} \mathbf{a}(\mathbf{z}, t) &= \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^{k\xi} - \mathbf{z}) \\ &= \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k + \Delta\mathbf{r}^{k\xi} - \mathbf{x} - \mathbf{y}) \\ &= \mathbf{a}(\mathbf{x}, \mathbf{y}, t) \end{aligned} \quad (3-3)$$

where N is the number of lattice cells and n is the number of atoms within a lattice cell;

$\mathbf{a}(\mathbf{x}, \mathbf{y}, t)$, or $\mathbf{a}(\mathbf{z}, t)$, is the local density function corresponds to $A(\mathbf{r}, \mathbf{p})$; $\delta(\cdot)$ is the localization function which can be a Dirac δ -function or a distribution of weighting function and has a unit of inverse volume.

Note that the function $\mathbf{a}(\mathbf{x}, \mathbf{y}, t)$ is continuous for both \mathbf{x} and \mathbf{y} . The cell-averaged continuously distributed local density function is defined as:

$$\begin{aligned}
\bar{a}(\mathbf{x}, t) &= \int_{V(\mathbf{x})} \sum_{k=1}^N \sum_{\xi=1}^n a(\mathbf{x}, \mathbf{y}, t) d\mathbf{y} / V(\mathbf{x}) \\
&= \int_{V(\mathbf{x})} \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k + \Delta\mathbf{r}^{k\xi} - \mathbf{x} - \mathbf{y}) d\mathbf{y} / V(\mathbf{x}) \\
&= \int_{V(\mathbf{x})} \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k + \Delta\mathbf{r}^{k\xi} - \mathbf{x} - \mathbf{y}) d\mathbf{y} \delta(\mathbf{R}^k - \mathbf{x}) \\
&= \int_{V(\mathbf{x})} d\mathbf{y} \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \delta(\Delta\mathbf{r}^{k\xi} - \mathbf{y})
\end{aligned} \tag{3-4}$$

In the above equation, $V(\mathbf{x})$ is the volume of the lattice cell that is located at spatial place \mathbf{x} . The condition $1/V(\mathbf{x}) = \sum_{k=1}^N \delta(\mathbf{R}^k - \mathbf{x})$ is derived from the normalization conditions:

$$\int_V \delta(\mathbf{R}^k - \mathbf{x}) d\mathbf{x} = 1 \quad (k = 1, 2, 3, \dots, N) \tag{3-5}$$

Let \mathbf{y}^α represent the internal relative position of α -th atom embedded within the lattice cell. It is obvious that the volume of a unit cell is the sum of the volume of each atom within the unit cell. So Equation (3-4) can be further written into the following form

$$\begin{aligned}
\bar{a}(\mathbf{x}, t) &= \int_{V(\mathbf{x})} d\mathbf{y} \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \delta(\Delta\mathbf{r}^{k\xi} - \mathbf{y}) \\
&= \sum_{\alpha=1}^n \int_{V(\mathbf{y}^\alpha)} d\mathbf{y} \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \delta(\Delta\mathbf{r}^{k\xi} - \mathbf{y}) \\
&= \sum_{\alpha=1}^n \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \int_{V(\mathbf{y}^\alpha)} d\mathbf{y} \delta(\Delta\mathbf{r}^{k\xi} - \mathbf{y}) \\
&= \sum_{\alpha=1}^n \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta\mathbf{r}^{k\xi} - \mathbf{y}^\alpha)
\end{aligned} \tag{3-6}$$

From the above equation, a new atomic-scale local density is defined as

$$\begin{aligned}
a_1(\mathbf{x}, \mathbf{y}^\alpha, t) &= \sum_{k=1}^N \sum_{\xi=1}^n A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta\mathbf{r}^{k\xi} - \mathbf{y}^\alpha) \\
\tilde{\delta}(\Delta\mathbf{r}^{k\xi} - \mathbf{y}^\alpha) &= \int_{V(\mathbf{y}^\alpha)} d\mathbf{y} \delta(\Delta\mathbf{r}^{k\xi} - \mathbf{y}) = \begin{cases} 1 & \text{if } \xi = \alpha \text{ and } \Delta\mathbf{r}^{k\alpha} = \mathbf{y}^\alpha \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{3-7}$$

This new local density function is continuous in \mathbf{x} , while discontinuous in α . The time evolution of this function can be obtained in the spattial description directly from its

definition as

$$\begin{aligned}
\left. \frac{\partial a_1(\mathbf{x}, \mathbf{y}^\alpha, t)}{\partial t} \right|_{\mathbf{x}, \mathbf{y}^\alpha} &= \int_{V(\mathbf{y}^\alpha)} d\mathbf{y} \sum_{k=1}^N \sum_{\xi=1}^n \frac{\partial}{\partial t} \left[A(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \delta(\Delta \mathbf{r}^{k\xi} - \mathbf{y}) \right] \\
&= \int_{V(\mathbf{y}^\alpha)} d\mathbf{y} \sum_{k=1}^N \sum_{\xi=1}^n \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \delta(\Delta \mathbf{r}^{k\xi} - \mathbf{y}) \\
&\quad + \int_{V(\mathbf{y}^\alpha)} d\mathbf{y} \sum_{k=1}^N \sum_{\xi=1}^n \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \delta(\Delta \mathbf{r}^{k\xi} - \mathbf{y}) \mathbf{V}^k \cdot \nabla_{\mathbf{R}^k} \delta(\mathbf{R}^k - \mathbf{x}) \\
&\quad + \int_{V(\mathbf{y}^\alpha)} d\mathbf{y} \sum_{k=1}^N \sum_{\xi=1}^n \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \Delta \mathbf{v}^{k\xi} \cdot \nabla_{\Delta \mathbf{r}^{k\xi}} \delta(\Delta \mathbf{r}^{k\xi} - \mathbf{y}) \quad (3-8) \\
&= \sum_{k=1}^N \sum_{\xi=1}^n \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y}) \\
&\quad + \sum_{k=1}^N \sum_{\xi=1}^n \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y}) \mathbf{V}^k \cdot \nabla_{\mathbf{R}^k} \delta(\mathbf{R}^k - \mathbf{x}) \\
&\quad + \sum_{k=1}^N \sum_{\xi=1}^n \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \Delta \mathbf{v}^{k\xi} \cdot \nabla_{\Delta \mathbf{r}^{k\xi}} \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y})
\end{aligned}$$

where $\mathbf{V}^k = \dot{\mathbf{R}}^k$ and $\Delta \mathbf{v}^{k\xi} = \Delta \dot{\mathbf{r}}^{k\xi}$. From the definition of localization function, one can

readily prove that

$$\nabla_{\mathbf{R}^k} \delta(\mathbf{R}^k - \mathbf{x}) = -\nabla_{\mathbf{x}} \delta(\mathbf{R}^k - \mathbf{x}) \quad \text{and} \quad \nabla_{\Delta \mathbf{r}^{k\xi}} \delta(\Delta \mathbf{r}^{k\xi} - \mathbf{y}^\alpha) = -\nabla_{\mathbf{y}^\alpha} \delta(\Delta \mathbf{r}^{k\xi} - \mathbf{y}^\alpha) \quad (3-9)$$

Then it follows that

$$\begin{aligned}
\left. \frac{\partial a_1(\mathbf{x}, \mathbf{y}^\alpha, t)}{\partial t} \right|_{\mathbf{x}, \mathbf{y}^\alpha} &= \sum_{k=1}^N \sum_{\xi=1}^n \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y}) \\
&\quad - \nabla_{\mathbf{x}} \sum_{k=1}^N \sum_{\xi=1}^n \mathbf{V}^k \otimes \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y}) \quad (3-10) \\
&\quad - \nabla_{\Delta \mathbf{r}^{k\xi}} \sum_{k=1}^N \sum_{\xi=1}^n \Delta \mathbf{v}^{k\xi} \otimes \dot{A}(\mathbf{r}(t), \mathbf{p}(t)) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y})
\end{aligned}$$

Balance Equations

Following from Equation (3-7), the continuous local mass density ρ^α , the linear momentum density $\rho^\alpha (\mathbf{v} + \Delta \mathbf{v}^\alpha)$, and the total energy density $\rho^\alpha e^\alpha$ in a multiatom system can be defined as

$$\rho^\alpha = \sum_{k=1}^N \sum_{\xi=1}^n m^\xi \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y}^\alpha) \quad (3-11)$$

$$\rho^\alpha (\mathbf{v} + \Delta \mathbf{v}^\alpha) = \sum_{k=1}^N \sum_{\xi=1}^n m^\xi \mathbf{V}^{k\xi} \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y}^\alpha) \quad (3-12)$$

$$\rho^\alpha e^\alpha = \sum_{k=1}^N \sum_{\xi=1}^n \left[\frac{1}{2} m^\xi (\mathbf{V}^{k\xi})^2 + U^{k\xi} \right] \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} - \mathbf{y}^\alpha) \quad (3-13)$$

where $U^{k\xi}$ is the site potential energy and $\mathbf{v} + \Delta \mathbf{v}^\alpha$ is the atomic-level local velocity field.

With the above three expressions, the conservation of mass, the balance of linear momentum and the conservation of energy equations can be derived following the time evolution laws given in Equation (3-10). It is also proved that the balance of moment of momentum equation is automatically valid in AFT (Chen and Lee 2005).

Conservation of mass:

$$\frac{\partial \rho^\alpha}{\partial t} = -\nabla_{\mathbf{x}} \cdot (\rho^\alpha \mathbf{v}) - \nabla_{\mathbf{y}^\alpha} \cdot (\rho^\alpha \Delta \mathbf{v}^\alpha) \quad (3-14)$$

Balance of linear momentum:

$$\begin{aligned} \frac{\partial \rho^\alpha (\mathbf{v} + \Delta \mathbf{v}^\alpha)}{\partial t} &= \nabla_{\mathbf{x}} \cdot (\mathbf{t}^\alpha - \rho^\alpha \mathbf{v} \otimes (\mathbf{v} + \Delta \mathbf{v}^\alpha)) \\ &\quad + \nabla_{\mathbf{y}^\alpha} \cdot (\boldsymbol{\tau}^\alpha - \rho^\alpha \Delta \mathbf{v}^\alpha \otimes (\mathbf{v} + \Delta \mathbf{v}^\alpha)) + \mathbf{f}^\alpha \end{aligned} \quad (3-15)$$

Conservation of energy:

$$\begin{aligned} \frac{\partial (\rho^\alpha e^\alpha)}{\partial t} &= \nabla_{\mathbf{x}} \cdot (\mathbf{q}^\alpha + \mathbf{t}^\alpha \cdot (\mathbf{v} + \Delta \mathbf{v}^\alpha) - \mathbf{v} \rho^\alpha e^\alpha) \\ &\quad + \nabla_{\mathbf{y}^\alpha} \cdot (\mathbf{j}^\alpha + \boldsymbol{\tau}^\alpha \cdot (\mathbf{v} + \Delta \mathbf{v}^\alpha) - \Delta \mathbf{v}^\alpha \rho^\alpha e^\alpha) + \mathbf{f}^\alpha \cdot (\mathbf{v} + \Delta \mathbf{v}^\alpha) \end{aligned} \quad (3-16)$$

where \mathbf{t}^α and $\boldsymbol{\tau}^\alpha$ are the homogeneous and inhomogeneous parts of moment flux;

\mathbf{q}^α and \mathbf{j}^α are the homogeneous and inhomogeneous parts of heat flux, respectively.

They are defined as

$$\begin{aligned} \mathbf{t}^\alpha &= \sum_{k=1}^N \tilde{\mathbf{V}}^k \otimes (m^\alpha \tilde{\mathbf{V}}^{k\alpha}) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\alpha} - \mathbf{y}^\alpha) \\ &\quad - \frac{1}{2} \sum_{k,l=1}^N \sum_{\xi,\eta=1}^n (\mathbf{R}^k - \mathbf{R}^l) \otimes \mathbf{F}^{k\xi} B(k, \xi, l, \eta, \mathbf{x}, \mathbf{y}^\alpha) \end{aligned} \quad (3-17)$$

$$\begin{aligned}\boldsymbol{\tau}^\alpha &= \sum_{k=1}^N \Delta \tilde{\mathbf{v}}^{k\alpha} \otimes (m^\alpha \tilde{\mathbf{V}}^{k\alpha}) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\alpha} - \mathbf{y}^\alpha) \\ &\quad - \frac{1}{2} \sum_{k,l=1}^N \sum_{\xi,\eta=1}^n (\Delta \mathbf{r}^{k\xi} - \Delta \mathbf{r}^{l\eta}) \otimes \mathbf{F}^{k\xi l\eta} B(k, \xi, l, \eta, \mathbf{x}, \mathbf{y}^\alpha)\end{aligned}\quad (3-18)$$

$$\begin{aligned}\mathbf{q}^\alpha &= \sum_{k=1}^N \tilde{\mathbf{V}}^k \left(\frac{1}{2} m^\alpha (\tilde{\mathbf{V}}^{k\alpha})^2 + U^{k\alpha} \right) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\alpha} - \mathbf{y}^\alpha) \\ &\quad - \frac{1}{2} \sum_{k,l=1}^N \sum_{\xi,\eta=1}^n (\mathbf{R}^k - \mathbf{R}^l) \tilde{\mathbf{V}}^{k\xi} \cdot \mathbf{F}^{k\xi l\eta} B(k, \xi, l, \eta, \mathbf{x}, \mathbf{y}^\alpha)\end{aligned}\quad (3-19)$$

$$\begin{aligned}\mathbf{j}^\alpha &= \sum_{k=1}^N \Delta \tilde{\mathbf{v}}^{k\alpha} \left(\frac{1}{2} m^\alpha (\tilde{\mathbf{V}}^{k\alpha})^2 + U^{k\alpha} \right) \delta(\mathbf{R}^k - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\alpha} - \mathbf{y}^\alpha) \\ &\quad - \frac{1}{2} \sum_{k,l=1}^N \sum_{\xi,\eta=1}^n (\Delta \mathbf{r}^{k\xi} - \Delta \mathbf{r}^{l\eta}) \tilde{\mathbf{V}}^{k\xi} \cdot \mathbf{F}^{k\xi l\eta} B(k, \xi, l, \eta, \mathbf{x}, \mathbf{y}^\alpha)\end{aligned}\quad (3-20)$$

where $\tilde{\mathbf{V}}^{k\alpha} = \mathbf{V}^{k\alpha} - \mathbf{v} - \Delta \mathbf{v}^\alpha$, $\tilde{\mathbf{V}}^k = \mathbf{V}^k - \mathbf{v}$ and $\Delta \tilde{\mathbf{v}}^{k\alpha} = \Delta \mathbf{v}^{k\alpha} - \Delta \mathbf{v}^\alpha$ are the difference between

phase space velocities and physical space velocities; for pair potential $\mathbf{F}^{k\xi l\eta} = \partial U^{k\xi l\eta} / \partial \mathbf{R}^{k\xi}$

is the internal force acted on atom $k\xi$; \mathbf{f}^α is the external force; and

$$B(k, \xi, l, \eta, \mathbf{x}, \mathbf{y}^\alpha) = \int_0^1 \delta(\mathbf{R}^k \lambda + \mathbf{R}^l (1-\lambda) - \mathbf{x}) \tilde{\delta}(\Delta \mathbf{r}^{k\xi} \lambda + \Delta \mathbf{r}^{l\eta} (1-\lambda) - \mathbf{y}^\alpha) d\lambda \quad (3-21)$$

From Equation (3-17) ~ (3-20), it is seen that \mathbf{t}^α and \mathbf{q}^α are respectively the

momentum flux and heat flux due to the motion and deformation of the lattice. They are consistent with the stress (Cauchy stress) and heat flux concepts in the mechanics of homogenized continua. $\boldsymbol{\tau}^\alpha$ and \mathbf{j}^α are respectively the momentum flux and heat flux due to the reorganization of atoms within the lattice cells. They describe the contribution to stress and heat flux due to atomic-scale inhomogeneity.

An instantaneous expression for the temperature in classical many-body dynamics is expressed in terms of the mean-squared velocity relative to the local velocity field (Hoover, 1986,1991)

$$T(\mathbf{x}) = \frac{V(\mathbf{x})}{3k_B} \sum_{k=1}^N M(\mathbf{V}^k - \mathbf{v})^2 \delta(\mathbf{R}^k - \mathbf{x}) = \frac{V(\mathbf{x})}{3k_B} \sum_{k=1}^N M(\tilde{\mathbf{V}}^k)^2 \delta(\mathbf{R}^k - \mathbf{x}) \quad (3-22)$$

where $V(\mathbf{x})$ is the volume of the unit cell located at \mathbf{x} , $M = \sum_{\alpha=1}^n m^\alpha$ is the total mass of the atoms in the unit cell, k_B is Boltzmann constant.

With this temperature definition and also the internal force density, the balance equation of linear momentum (Equation (3-15)) can be rewritten as

$$\rho^\alpha \frac{d}{dt} (\mathbf{v} + \Delta \mathbf{v}^\alpha) = \mathbf{f}_{\text{int}}^\alpha + \mathbf{f}_{\text{ext}}^\alpha - \frac{\gamma^\alpha k_B}{\Delta V} \nabla_{\mathbf{x}} T \quad (3-23)$$

or

$$\rho^\alpha \ddot{\mathbf{u}}^\alpha(\mathbf{x}) = \mathbf{f}_{\text{int}}^\alpha(\mathbf{x}) + \mathbf{f}_{\text{ext}}^\alpha - \frac{\gamma^\alpha k_B}{V(\mathbf{x})} \nabla_{\mathbf{x}} T \quad (3-24)$$

where $u^\alpha(\mathbf{x})$ is the displacement of the α -th atom embedded at the material point at \mathbf{x} ;

$\gamma^\alpha = m^\alpha / M$ is the fraction of contribution of the α -th atom to the temperature.

Generally, the internal force density

$$\mathbf{f}_{\text{int}}^\alpha = \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^n \mathbf{f}(\mathbf{u}^\alpha(\mathbf{x}) - \mathbf{u}^\beta(\mathbf{x}')) d\mathbf{x}' \quad (3-25)$$

is a nonlocal and nonlinear function of atomic displacements, and can be obtained through fitting to experimental measurements or first principle calculations. Equation (3-25) serves as the constitutive equation which links the displacement field to the internal force.

It is seen from Equation (3-25) that the internal force may depend on the high order gradient of displacement field which corresponds to the strain and strain gradient in classical continuum mechanics. Recall that the Equation (3-14) ~ (3-16) have the similar form as the conservation of mass, balance of linear momentum and conservation of energy equations in classical continuum mechanics, respectively. For this reason, we can treat AFT as a generalized continuum theory.

Difference between Particles Dynamics and Continuous Field Theories

Unlike many-body dynamics, a continuous field theory is only concerned with local

densities, such as mass density, stress tensor, heat flux vector and energy density. The evolutions of local densities in space and time are governed by balance equations. Supplemented by constitutive relations, the balance equations completely determine the dynamic behavior of a material system under a given external field and properly imposed boundary and initial conditions.

With the local densities defined in the continuous field theory, balance equations, always serve as the governing equations, can be derived and used to describe the motion of the system. While for particle dynamics, classical mechanics serves the governing equation for many-body discrete systems.

AFT is a continuous field theory which is based on local densities and balance equations. Different from atomistic simulation methods which are based on discrete many-body dynamics, temperature is defined as an independent variable in AFT. This variable can be directly solved from the conservation of energy equation. It is interested to find that the balance equations in AFT will be equivalent to many-body dynamics for MD simulation when focusing on the motion of every atom in the system.

The finite element implementation of AFT

Weak Form of the Governing Equation

From the previous section, it is seen that AFT is a nonlinear and nonlocal field theory which is able to describe the motion of materials at any length scale. The governing equations of AFT have been numerically solved by meshfree method to investigate the dynamic fracture of a model composite material (Deng et al. 2009). In this dissertation, a finite element implementation of AFT is introduced. It is seen that, for materials under a given temperature field, Equation (3-24) supplemented by Equation (3-25) completely govern the evolution of the atomic-scale displacements in space and

time. Substituting Equation (3-25) into Equation (3-24), we have the following governing equation

$$\rho^\alpha \ddot{\mathbf{u}}^\alpha(\mathbf{x}) = \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^n \mathbf{f}(\mathbf{u}^\alpha(\mathbf{x}) - \mathbf{u}^\beta(\mathbf{x}')) d\mathbf{x}' + \mathbf{f}_{\text{ext}}^\alpha \quad (3-26)$$

here we omit the temperature term since we are considering a homogeneous or a given temperature field. Under these special situations, the temperature term which has been omitted here is either zero or a constant. Therefore only Equation (3-26) is needed as the governing equation. The conservation of mass equation and the conservation of energy equation (Equation (3-14), (3-16)) are automatically satisfied. There are $3n$ equations in (3-26) need to be solved for determining exactly the same number of unknowns (\mathbf{u}^α , $\alpha=1, 2, \dots, n$).

In this section, the finite element method is used to numerically solve Equation (3-26). Notice that Equation (3-26) has the similar form as the governing equation for classical continuum theory. Different from the classical continuum theory which explicitly involves the spatial derivatives of displacement field in evaluating the internal force field, Equation (3-26) employs the integration over $\Omega(\mathbf{x}')$, the domain covers the whole system, to determine the internal force field. Benefit from the avoiding the spatial derivatives, we can easily overcome the difficulties in solving the high gradient or local discontinuity problems with AFT.

Firstly, we approximate the displacement field within each element by

$$\hat{\mathbf{u}}^\alpha(\mathbf{x}) = \Phi_\xi(\mathbf{x}) \mathbf{U}_\xi^\alpha, \quad \text{or} \quad \hat{u}_i^\alpha(\mathbf{x}) = \Phi_\xi(\mathbf{x}) U_{i\xi}^\alpha, \quad \xi=1, 2, \dots, n, \quad i=1, 2, 3 \quad (3-27)$$

where $\hat{\mathbf{u}}^\alpha(\mathbf{x})$ stands for the approximation of atomic displacement field within the element, $\Phi_\xi(\mathbf{x})$ is the shape function, and $U_{i\xi}^\alpha$ is the i -th component of the displacement of the α -th atom within the ξ -th node. Here, as shown in Figure 3-3, each element

node is associated with a unit cell which contains n atoms.

Then the weak form of Equation (3-26) can then be written as

$$\begin{aligned}
& \int_{\Omega(\mathbf{x})} \rho^\alpha \Phi_\eta(\mathbf{x}) \Phi_\xi(\mathbf{x}) d\mathbf{x} \ddot{U}_\xi^\alpha \\
& - \int_{\Omega(\mathbf{x})} \Phi_\eta(\mathbf{x}) \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^n \mathbf{f} \left[\Phi_\xi(\mathbf{x}) U_\xi^\alpha - \Phi_\xi(\mathbf{x}') U_\xi^\beta \right] d\mathbf{x}' d\mathbf{x} \\
& - \int_{\Omega(\mathbf{x})} \Phi_\eta(\mathbf{x}) (\mathbf{f}_{\text{ext}}^\alpha) d\mathbf{x} \\
& = 0
\end{aligned} \tag{3-28}$$

where both $\Omega(\mathbf{x})$ and $\Omega'(\mathbf{x})$ represent the domain of the whole system. And on the left hand side of this equation, the first term is the inertial term; the second term is the internal force term; the third term is the external force term.

Numerical Evaluation of Each Term in Weak Form Equation

To calculate the inertial term, the lump mass assumption is used. The lumped mass matrix can be defined as the following form:

$$\mathbf{M}^\alpha \triangleq \int_{\Omega_I(\mathbf{x})} \rho^\alpha \Phi_\eta(\mathbf{x}) \Phi_\xi(\mathbf{x}) d\mathbf{x} \approx \begin{bmatrix} m_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & m_N \end{bmatrix} \tag{3-29}$$

The internal force term can be obtained with numerical integration strategies such as Gauss integration or nodal integration. For nodal integration, the discretized internal force term can be written in the following form:

$$\begin{aligned}
\mathbf{F}_{\text{int}}^\alpha &= \int_{\Omega(\mathbf{x})} \Phi_\eta(\mathbf{x}) \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^v \mathbf{f} \left[\Phi_\xi(\mathbf{x}) U_\xi^\alpha - \Phi_\xi(\mathbf{x}') U_\xi^\beta \right] d\mathbf{x}' d\mathbf{x} \\
&= \sum_{J=1}^{N_p} w_J \Phi_\eta(\mathbf{x}_J) \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^v \mathbf{f} \left[\Phi_\xi(\mathbf{x}_J) U_\xi^\alpha - \Phi_\xi(\mathbf{x}') U_\xi^\beta \right] d\mathbf{x}'
\end{aligned} \tag{3-30}$$

where $\sum_{J=1}^{N_p}$ is the summation over all the nodes; w_J is the weight associated with each

node. It is seen from the above equation that the internal force term is a function of the displacements of all the nodes.

It is easy to see that the integration of the internal force over each element is contributed by two different parts: internal force on the boundary and in the bulk region within each element. The reason for this difference is that the internal force in the bulk region of an element only depends on the nodal displacement of this element, while the internal force on the boundary of an element not only depends on the nodal displacements of this element but also depends on the nodal displacements of its neighboring elements as well. In other word, the bulk region part only locally depends on the displacement field within the element, while the boundary part also depends on the displacement field of the neighboring elements. For homogeneous deformation, the displacement field varies smoothly with space, so there are few differences between interior and boundary regions. However in some critical situation, such as dislocation or crack nucleation and propagation, the displacement field may have high gradient in some critical regions. At this time, it is necessary to distinguish the interior and boundary regions.

$$\begin{aligned}
\mathbf{F}_{\text{int}}^{\alpha} &= \int_{\Omega(\mathbf{x})} \Phi_{\eta}(\mathbf{x}) \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^{\nu} \mathbf{f} \left[\Phi_{\xi}(\mathbf{x}) \mathbf{U}_{\xi}^{\alpha} - \Phi_{\xi}(\mathbf{x}') \mathbf{U}_{\xi}^{\beta} \right] d\mathbf{x}' d\mathbf{x} \\
&= \sum_{J=1}^{N_p} \Phi_{\eta}(\mathbf{x}_J) \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^{\nu} \mathbf{f} \left[\Phi_{\xi}(\mathbf{x}_J) \mathbf{U}_{\xi}^{\alpha} - \Phi_{\xi}(\mathbf{x}') \mathbf{U}_{\xi}^{\beta} \right] d\mathbf{x}' \\
&\quad + \sum_{K=1}^{N_s} w_S \Phi_{\eta}(\mathbf{x}_K) \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^{\nu} \mathbf{f} \left[\Phi_{\xi}(\mathbf{x}_K) \mathbf{U}_{\xi}^{\alpha} - \Phi_{\xi}(\mathbf{x}') \mathbf{U}_{\xi}^{\beta} \right] d\mathbf{x}' \\
&\quad + \sum_{L=1}^{N_B} w_B \Phi_{\eta}(\mathbf{x}_L) \int_{\Omega'(\mathbf{x})} \sum_{\beta=1}^{\nu} \mathbf{f} \left[\Phi_{\xi}(\mathbf{x}_L) \mathbf{U}_{\xi}^{\alpha} - \Phi_{\xi}(\mathbf{x}') \mathbf{U}_{\xi}^{\beta} \right] d\mathbf{x}'
\end{aligned} \tag{3-31}$$

The above Equation (3-31) is for the numerically evaluation of internal force using Gauss integration strategy. The integration of internal force over the whole domain is divided into three parts: the nodal part, the boundary part and the bulk region part. The node part includes all the internal forces on all the element nodes in the system.

$\sum_{J=1}^{N_P}$ means the summation over all element nodes. The boundary part includes the internal forces on the boundaries of all elements. $\sum_{K=1}^{N_S}$ means the summation over all the gauss points on the surface. The bulk region part includes the internal forces inside each element, the internal force field in this region should only depend on the nodal displacement of the element where the region located. $\sum_{L=1}^{N_B}$ means the summation over all the gauss points in this region. The notations w_S and w_B represent the weights for the surface gauss points and bulk gauss points respectively. It is worthwhile to mention that the weight for the nodal part summation is 1 instead of w_J in nodal integration version.

Compared with the Gauss integration procedure, nodal integration is significant more efficient, easy to implement, but less accurate and usually leads to instability. In order to stabilize the nodal integration strategy, the nodal integration form can be rewritten, using Newton's third law $\mathbf{f}[\Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta] = -\mathbf{f}[\Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta - \Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha]$, into the following form

$$\begin{aligned}
\mathbf{F}_{\text{int}}^\alpha &= \frac{1}{2} \int_{\Omega} \Phi_\eta(\mathbf{x}) \int_{\Omega'} \sum_{\beta=1}^n \mathbf{f}[\Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta] d\mathbf{x}' d\mathbf{x} \\
&\quad - \frac{1}{2} \int_{\Omega} \Phi_\eta(\mathbf{x}) \int_{\Omega'} \sum_{\beta=1}^n \mathbf{f}[\Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta - \Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha] d\mathbf{x}' d\mathbf{x} \\
&= \frac{1}{2} \sum_{\beta=1}^n \left\{ \int_{\Omega} \Phi_\eta(\mathbf{x}) \int_{\Omega'} \mathbf{f}[\Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta] d\mathbf{x}' d\mathbf{x} \right. \\
&\quad \left. - \int_{\Omega} \Phi_\eta(\mathbf{x}) \int_{\Omega'} \mathbf{f}[\Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta - \Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha] d\mathbf{x}' d\mathbf{x} \right\}
\end{aligned} \tag{3-32}$$

In the above Equation (3-32), since the integration domains Ω and Ω' both indicate the region that cover the whole system, so there is no difference to exchange the notations with ' with those without ' just like exchanging the damming indexes in tensor

analysis. So Equation (3-32) can be further rewritten as

$$\begin{aligned}
\mathbf{F}_{\text{int}}^\alpha &= \frac{1}{2} \sum_{\beta=1}^n \left\{ \int_{\Omega} \Phi_\eta(\mathbf{x}) \int_{\Omega'} \mathbf{f}[\Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta] d\mathbf{x}' d\mathbf{x} \right. \\
&\quad \left. - \int_{\Omega} \Phi_\eta(\mathbf{x}) \int_{\Omega'} \mathbf{f}[\Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta - \Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha] d\mathbf{x}' d\mathbf{x} \right\} \\
&= \frac{1}{2} \sum_{\beta=1}^n \left\{ \int_{\Omega} \Phi_\eta(\mathbf{x}) \int_{\Omega'} \mathbf{f}[\Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta] d\mathbf{x}' d\mathbf{x} \right. \\
&\quad \left. - \int_{\Omega'} \Phi_\eta(\mathbf{x}') \int_{\Omega} \mathbf{f}[\Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\beta - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\alpha] d\mathbf{x} d\mathbf{x}' \right\} \\
&= \frac{1}{2} \sum_{\beta=1}^n \left\{ \int_{\Omega} \int_{\Omega'} \Phi_\eta(\mathbf{x}) \mathbf{f}[\Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\alpha - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta] d\mathbf{x}' d\mathbf{x} \right. \\
&\quad \left. - \int_{\Omega'} \int_{\Omega} \Phi_\eta(\mathbf{x}') \mathbf{f}[\Phi_\xi(\mathbf{x})\mathbf{U}_\xi^\beta - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\alpha] d\mathbf{x} d\mathbf{x}' \right\} \\
&= \frac{1}{2} \sum_{J=1}^{N_p} w_J \Phi_\eta(\mathbf{x}_J) \sum_{\beta=1}^n \left\{ \int_{\Omega'} \Phi_\eta(\mathbf{x}_J) \mathbf{f}[\Phi_\xi(\mathbf{x}_J)\mathbf{U}_\xi^\alpha - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\beta] d\mathbf{x}' \right. \\
&\quad \left. - \int_{\Omega'} \Phi_\eta(\mathbf{x}') \mathbf{f}[\Phi_\xi(\mathbf{x}_J)\mathbf{U}_\xi^\beta - \Phi_\xi(\mathbf{x}')\mathbf{U}_\xi^\alpha] d\mathbf{x}' \right\}
\end{aligned} \tag{3-33}$$

In summary, Equation (3-33) is employed to calculate the internal force on the α -th atoms of the η -th node in nodal integration version. While in the Gauss integration version, Equation (3-31) is used for the computation of internal force. Figure 3-4 shows the differences of these two numerical integration strategies. In the pictures, the solid dark points indicate the nodal points where both the internal forces and the displacement need to be determined and the star points indicate the Gauss points where only the internal forces need to be determined.

Since the external force field is always explicitly given, so the external force term can be analytically derived out. It is also worthwhile to mention that the temperature term is omitted in this work because of the temperature field is always assumed to be given in our current consideration. If the temperature field is not explicitly given, the thermal-mechanical coupling problem needs to be solved. At this time, the equation of conservation of energy (Equation (3-16)) will be involved to solve the additional field

variables, temperature.

Finally, the discretized governing equation

$$\mathbf{M}^\alpha \ddot{\mathbf{U}} - \mathbf{F}_{ext}^\alpha - \mathbf{F}_{int}^\alpha = 0 \quad (3-34)$$

is derived by combining Equation (3-28) ~ (3-33). The central difference method is used for solving the discretized governing Equation (3-33). Some details of the central difference algorithm in solving this equation will be addressed later in Chapter 4.

Mesh Strategies

Since there is no space derivatives involved in the formulation, the continuum finite elements do not have to be connected. This provides us with more freedom in designing the mesh strategy. As shown in Figure 3-5, there are two options for the mesh strategy: with and without connectivity. Of course the latter is suitable for simulating, with coarse mesh, the nucleation and propagation of discontinuities throughout the whole region. But since the coarse mesh is used instead of the finest mesh, some of the high frequency waves might be suppressed when using low order interpolation functions. This unphysical suppressing can be avoided by using physically based shape functions. In this work, our goal is to simulate the artificial crack propagation in material. So the coarse mesh without connectivity strategy is employed here. And since the element size is very small, only linear shape functions are used.

It is worthwhile to mention that since there is no connectivity between neighboring elements, each node only belongs to one element. Then the integration over the whole domain can be divided into the integrations over the elements in the system. Thus the summation of all nodes or Gauss points in the system can be simply divided into the summation of the nodes or Gauss points in each element. In other word, no assembling is needed.

Since the single crystalline material is anisotropic, its fracture behaviors are direction depended. For these reason, those planes which are easier to be separated should be chosen as the surfaces of element. Different from in the cohesive zone model, these potential paths for fracture evolutions is physically selected. This is also one of the significant advantages of the new CG method over the classical finite element method. In the following, two different finite elements based on material structures are introduced.

2D triangular element

In the study of 2D brittle fracture problems, 2D triangular lattice is the most frequently used model (Holian and Ravelo 1995, Zhou et al. 1996, Abraham et al. 1994, Buehler et al. 2003, Buehler and Gao 2006). This lattice model is simple and useful. The model represents the (1 1 1) plane of the face center cubic (FCC) crystalline material. Since the triangular unit formed by the neighboring lattice points is equilateral triangle, this lattice model is isotropic.

For a triangular lattice shown in the left side picture of Figure 3-6, the density of bonds along the red dash lines are less than the density of bonds along the blue dot dash line. This means that the crack is easier to propagate along the red dash lines than the blue dot dash line. Actually, the bond densities along these three directions are the lowest in all the possible directions. So these red lines outline the three possible directions of weakest planes. The right side picture of Figure 3-6 shows the shape of a finite element. This element is in the shape of equilateral triangle to maintain the isotropy of the material.

As shown in Figure 3-6, area coordinates are used as the shape functions for this 2D triangular element. The red stars in the triangular element indicate the Gauss points for edge and area integrations. Then following the traditional procedure for treating the

triangular element, the shape functions are substituted into Equation (3-34) to get the equations for solving the nodal displacements.

3D hybrid element

It is widely accepted that FCC crystals seldom show brittle behaviors. In most of the case, this kind of materials display ductile properties with dislocation dominant its fracture. Zhou et al. (1999), through MD simulation, showed that in FCC crystal the crack propagation is close related to the dislocation. They also observed that the crack always kinks to the directions along the slip planes. Motivated by this finding, a hybrid element composed of two tetrahedrons and one octahedron is introduced here to study the fracture of FCC crystal.

The thick red lines and blue lines in the left picture of Figure 3-7 outline a primitive cell of the FCC crystal. a_1 , a_2 , and a_3 are three lattice vectors. As shown in Figure 3-7, this primitive cell can be divided into two tetrahedrons and one octahedron. For FCC crystal, every slip planes should be parallel to one of the four surfaces of the tetrahedron shows in Figure 3-7.

Obviously it is too cost to embed all slip planes in a CG model. However, all these slip planes can be sorted by their orientation into four classes. Within each class, the slip planes are parallel to each other. The tetrahedral element in Figure 3-8 shows the four classes of planes. Every slip plane in this material should be parallel to one of the four faces of the tetrahedral element.

As shown in Figure 3-8, a hybrid element for FCC single crystals is composed of two tetrahedral sub-elements and two pyramid sub-elements. The shape of this hybrid element is the same as a rhombohedral primitive unit cell of the FCC single crystal. By introducing two internal surfaces between the tetrahedral and the pyramid elements, the

hybrid finite element, with 16 faces, shows all of the four classes of slip planes. Note that the 16 faces are divided into four groups and each group, with four parallel faces, corresponds to a class of slip planes. Furthermore the total areas of each group of faces are the same. So the discontinuity can propagate in all the possible directions with equal possibility.

Within each hybrid element, the conventional linear shape functions are used to interpolate the displacement field in each sub-element. Between neighboring hybrid elements, there is a gap of $\frac{\sqrt{3}}{3}a$ (a is the length of a lattice constant) to separate different elements as well as the gap separates tetrahedral and pyramid sub-elements. Since the displacement field across the gap is not necessary to be continuous, the discontinuities, such as dislocations or cracks, can propagate along the space within or between elements.

The shape functions for

$$N_i(\mathbf{x}) = L_i(\mathbf{x}), (i=1 \sim 4) \quad (3-35)$$

where L_i is the i -th component of the volume coordinate at point \mathbf{x} .

Summary

In this chapter, a new coarse grained method based on AFT is presented in details. One of the most significant features of the new developed method is its outstanding ability in simulating dynamic fracture problems without losing the atomic information. As a field theory, AFT unifies the governing equations in both atomic and continuum regions. Under the framework of AFT, the interatomic potential can be directly served as the constitutive relations for solving the balance equations. The coarse grained method developed in this work is a finite element implementation of AFT. Instead of stress-strain

relation, in the new coarse grained method, interatomic potential is used to calculate the internal nodal forces. It is worthwhile to mention that not only the interatomic potential but also the atomic structure of the material are embedded in the coarse grained model. Each finite element should correspond to a specified material. In this chapter, two kinds of finite elements are developed for two different materials. A 2D triangular element is designed for 2D triangular lattice. A 3D hybrid element is designed for 3D single crystal with rhombohedral primitive cell. One of the advantages of this treatment is that the faces of the element are always the weakest planes where the failure is most likely to occur on them. This CG method has been successfully used to investigate the dislocation and dynamic crack branching in material (Xiong et al. 2011; Deng et al. 2010). In Chapter 5 and 6, some details on the application of this CG method for dynamic fracture will be presented.

It is worthwhile to mention that, for the present method, only the balance equation of linear momentum is solved since only given temperature cases are under the consideration here. When crack is propagating, the temperature field near the crack tip could become very complex. To take into account the thermo-mechanical effect, we need to solve the conservation of energy equation along with the balance equation of linear momentum.

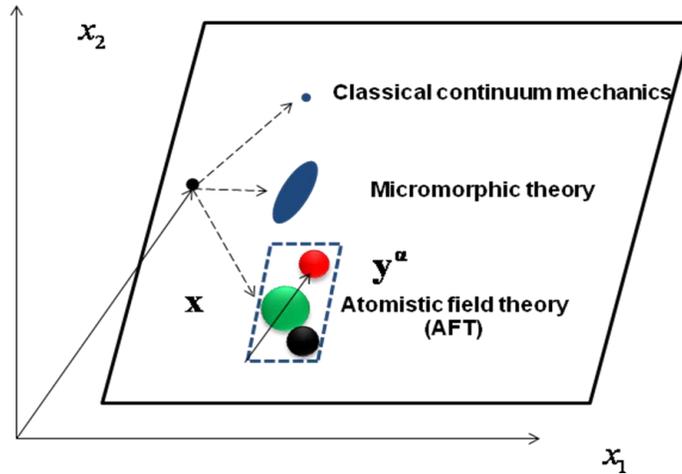


Figure 3-1. Relationship between AFT and continuum mechanics

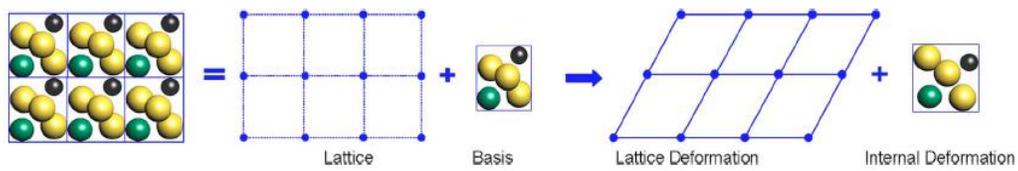


Figure 3-2. Decomposition of atomic positions

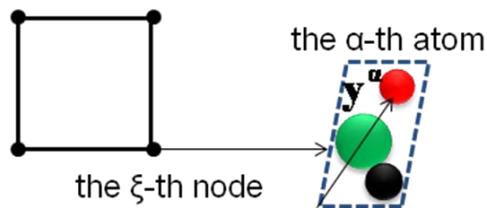
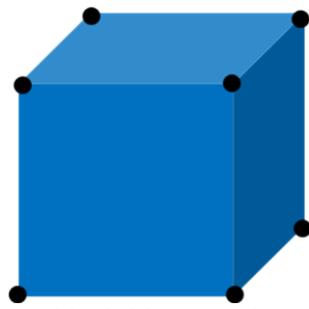
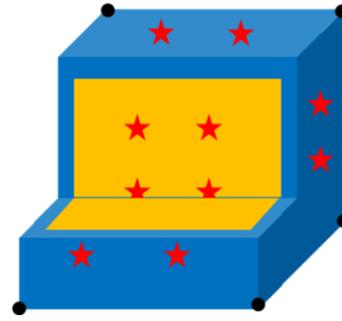


Figure 3-3. Schematic scratch for finite elements in the FE implementation of AFT

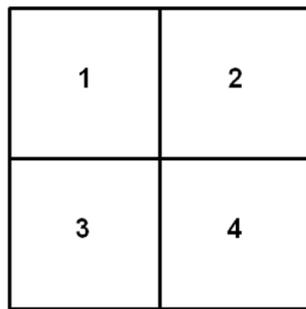


(a) Nodal integration

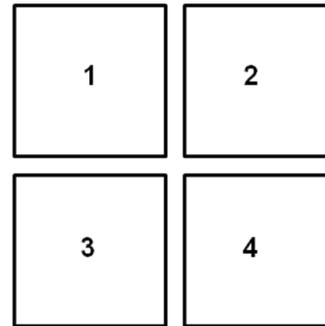


(b) Gauss integration

Figure 3-4. Nodal integration and Gauss integration



(a) with connectivity



(b) without connectivity

Figure 3-5. Two options of the mesh strategy

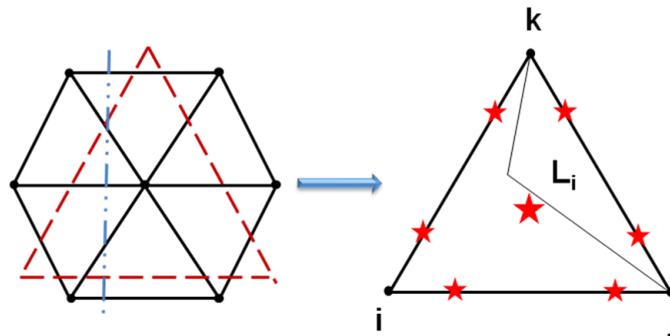


Figure 3-6. The triangular element which shows the weakest planes

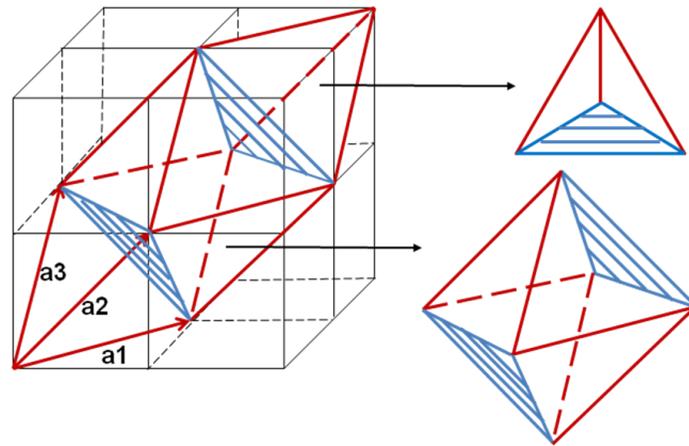


Figure 3-7. The primitive unit cell and slip planes of FCC crystal

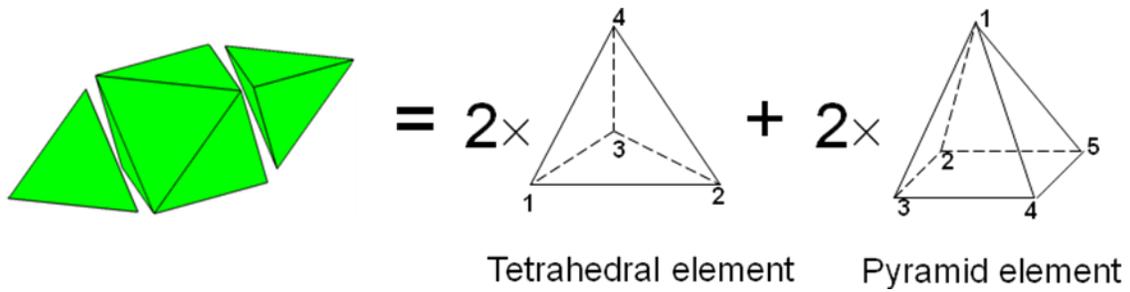


Figure 3-8. Sketch of one hybrid element

CHAPTER 4 PARALLEL IMPLEMENTATION OF CG-AFT METHOD

Introduction

The fast development of high performance computers has provided us a great opportunity for massive computation. Using multi-processor platforms, not only the computational time but also the memory can be largely reduced. These two advantages are also critical to the scale up of the simulation using the new CG method. For most supercomputers, the memory allocated to each core is less than 8GB. So the total number of elements that can be handled by a series code should not exceed 4000. With the parallel code, the total number of elements in a simulation model can be much larger than this critical value. In this chapter, the structure and the implementation of the parallel code is introduced in details.

The performance of this parallel code is tested via stability and scalability analyses. To test the stability of the code, the same problem is computed with different numbers of processors. For a stable simulation, these results should be identical to each other. The scalability of a parallel code reflects how the parallel code performs with different numbers of processors. In this work, the effects of the size of the simulation system on the performance of the parallel code are also investigated.

Four design spaces for the parallel code

There are four design spaces in the design of a parallel code: finding concurrency, algorithm structure, supporting structures, and implementation mechanisms. In this section, some details of these four design spaces and their specifications for the studied code will be presented.

In the finding concurrency design space, the tasks and data are decomposed first.

Then the decomposed sub-tasks are grouped and ordered according to their functions and sequences. Finally, the data sharing for the sub-tasks is analyzed. This design space, including decomposition and dependency analysis, is an important step in implementing a parallel program. After decomposing and grouping the key tasks, we have the following five groups of tasks:

1. a group of tasks to read input data
2. a group of tasks to construct the neighbor list
3. a group of tasks to calculate the nodal forces
4. a group of tasks to update the position and velocities of each node
5. a group of tasks to output the results

There are three groups of key data:

1. the coordinates and velocities for each node
2. the forces for each node
3. the neighbor list for each node

Data group 1 is initiated by task group 1. Then it is used by task groups 2 and 3.

Task group 4 updates the data in data group 1. Data group 2 is generated by task group 2 and then is used by task group 4 in the updating. Data group 3 is generated or updated by task group 2 and then used by task group 3. The data groups used by task group 5 depend on what information needs to be output.

The goal of the second design space, algorithm structure, is to decide which pattern or patterns are most appropriate for the problem. For different problems, different algorithm structure patterns are needed. The criteria for choosing the pattern is to best exploit the concurrency. The program is composed of a large loop of time evolutions. The result of each time step is an input for the next step. Within each time step, the nodal forces and positions are updated independently. So we can use different processors to handle these jobs concurrently. For this problem, either task parallelism or

a geometric decomposition algorithm could be used. The former is easy to implement, but needs more time in communication between processors; the latter is more difficult in implementation, but could save more communication time, especially for very large systems. In this work, to enable our program to handle an extremely large model, we use the geometric decomposition algorithm.

The third design space, supporting structures, is more close to the programming. In this space, the program structure and data structure are determined. In this work, the SPMD (single program, multiple data) pattern is used for the tasks within one time step. The distributed array pattern is used to partition the large arrays between different processors. Since the geometric decomposition algorithm is used, the arrays are partitioned according to the geometry of the nodes.

The fourth space, implementation mechanisms, includes the details of the the parallel program implementing. The three core subsets of these mechanisms are: process or thread management, synchronization, and communication. As was mentioned before, MPI is used for the parallel program implementation. MPI is fundamentally based on processes. An MPI program with an executable named foo is launched through the command: "mpirun -np N foo". The processes are destroyed when the programs running on the nodes of the parallel computer exit. There are a large number of operations in MPI, the most useful of which are in the following three groups: (1) basic point-to-point message passing; (2) collective operations; (3) advanced point-to-point message passing. The details of these operations will not be listed here because they can be easily found in any MPI manual such as: *MPI—The Complete Reference* by Marc Snir et al.

After the above analysis, the parallel strategy for this problem is given by the following steps:

1. Read the information from the input file. This should be done by every processor at the beginning of the computation. This is done by subroutine **INPUT**.
2. Divide the whole domain into P sub-domains (P is the number of processors) and find neighbors for each sub-domain. Instead of restoring the information for every node in the system, each processor only restores the nodal information, such as positions, velocities and forces, of its own sub-domain. Each processor also maintains a local list of the local indices and global indices of the nodes and elements in this sub-domain. This is done by subroutine **GETMAP**.
3. For every few time steps, construct the neighbor list with the link-cell strategy. Each processor only constructs the neighbor list of the nodes in its own sub-domain. The radius of this neighbor list is r_b . This is done by subroutines **ELIST**.
4. For each time step, update the positions of nodes of boundary elements by inter-processor communications. This is done by subroutine **EXCHANGE_U**.
5. For each time step, compute the forces of each node within the sub-domain. This is done by subroutine **GETFORCE**.
6. For each time step, update the positions of nodes within the sub-domain. This is done by subroutine **CENTRAL**.
7. Output the results every N steps; N is set by the user of the program. This is done by subroutine **OUTPUT**.

The structure of the parallel code

The flow chart shown in Figure 4-1 depicts the structure of the current parallel code. In this figure, t_GMAP , t_ELIST , t_EXC , t_CFR , and t_CEN represent the CPU time for the key steps GETMAP, ELIST, EXCHANGE_U, CFORCE, and CENTRAL, respectively. EXCHANGE_U, CFORCE and CENTRAL are the three steps which need to be executed every time step. The average number of time steps for a calculation is on the order of several tens of thousands. The time t_SOLVE equals to the sum of t_EXC , t_CFR , and t_CEN . In the following, these key steps will be introduced separately.

Domain Decomposition, GETMAP

GETMAP is used to divide the whole system into several domains with no overlaps. Each domain is allocated to a single processor. The more evenly assigned the load is among the processors, the more efficient the parallel computing. Therefore, in this subroutine, the whole system is divided into domains with almost the same number of finite elements. To divide the model shown in Figure 4-2 (a), first a mapping protocol is used to transfer the model into the cubic shown in Figure 4-2 (b). Then the cubic is evenly divided into many domains, as shown in Figure 4-2(c). In current CG-AFT method, solving the nodal positions of each domain involves not only the nodes within this domain, but also the nodes of its neighboring domains.

Figure 4-3 shows the interfaces between domains. The interactions across these interfaces involve the neighboring domains. Thus the other objective of this subroutine is to record the neighbors of each domain which will be used in the future for data communication between neighboring domains.

Construct Neighborlist, ELIST

As is presented in Chapter 3, AFT is a nonlocal theory and the internal force on each atom depends on its neighbors through Equation (3-25). In the proposed coarse grained method, only the forces on Gauss points are needed for numerically evaluating the integration. It is therefore necessary to construct a list of neighbors, called 'neighborlist', for each Gauss point before the calculation of its internal force. ELIST is a subroutine designed for the construction of neighborlist. Following the numerical integration scheme introduced in Section 3.2, there are Gauss points associated with node, edge, surface, and body. ELIST will construct the neighborlist for all the Gauss points. ELIST is divided into two main steps. The first step is to construct the element

neighborlist for each element. The second step is to find the neighborlist of the Gauss points of each element only for its neighboring elements. Here the assumption is made that the finite element is large enough so that all the neighbors of a Gauss point are located in its own element or the neighboring elements. Considering a system with m finite elements, if there are n unit cells and g Gauss points in each element, then the computation complexity of ELIST is of the order $m^2 + 27m \cdot n \cdot g$. Here the factor 27 indicates that each element has 27 neighboring elements in a 3D system. For a non-uniform mesh, this number might vary slightly. For a parallel scheme, each processor only deals with the Gauss points within its own domain. If there are no processors used, then the computation complexity for each processor is of the order $27\left(\frac{m}{np}\right)^2 \left(1 + \frac{n \cdot g \cdot np}{m}\right)$.

When m is very large, the term $\frac{n \cdot g \cdot np}{m}$ is much smaller than the number 1. Then this

step involves $\left(\frac{m}{np}\right)^2$ order calculations. However, it is not necessary to call this

subroutine every time step. In most cases, it is called every few hundred or thousand steps.

Communication Between Processors, EXCHANGE_U

EXCHANGE_U and GETMAP are two of the most important steps in parallel implementation. GETMAP provides the parameters for communication and EXCHANGE_U performs the communication with the parameters provided by GETMAP. As shown in Figure 4-4, EXCHANGE_U communicates the updated nodal displacements between neighboring domains. Figure 4-4 shows an example of 3D communication. This communication is divided into three steps, and in each step the

communication is performed in one dimension (x direction, y direction, or z direction). After these three steps, the information of each domain is shared by all of its neighboring domains.

Calculation of Nodal Forces, CFORCE

The step of calculating the nodal force, CFORCE, is the implementation of numerical integration. Using Equation (3-31) or (3-33), the Gauss integration or nodal integration strategies are used to calculate the nodal force numerically. In this step, the nodal force of a finite element is calculated based on the internal forces on all Gauss points belonging to this finite element. Each processor only needs to handle the elements assigned to it. This is the reason why the computational time and memory can be largely saved by using the proposed parallel algorithm. The efficiency of this step determines the overall efficiency of the parallel code. This will be discussed in detail later in the following sections.

Time Integrations, CENTRAL

The introduced new CG method involves both spacial and time integrations. The spacial integration has been discussed above. The time integration is numerically achieved using CENTRAL. In this work, the explicit numerical integration scheme, the central difference method, is applied for this time integration. Like in CFORCE, each processor only needs to handle the nodes within its own domain. Figure 4-5 shows the main steps for the central difference method used here.

Performance of the parallel code

Stability

Stability is one of the most important features of a computational code. For parallel code, the computational results are always different with different numbers of

processors. The computation results from an unstable code are useless. Unstable computation always results in nonphysical or infinitely large values, so the stability test is the first step in testing the performance of our parallel code.

As the first example, the stress wave propagation in a face center cubic (FCC) single crystal bar is simulated. As shown in Figure 4-6, 6400 ($4 \times 4 \times 40$) hybrid element are used to discretize the bar. The Lennard-Jones pair potential is used to describe the interactions between atoms. The bar is clamped at one end and a force perturbation is applied to the other end.

Figure 4-7 shows the variation of the average stress with time when the stress waves propagate in the bar. From the curves, it is seen that the parallel code is stable. We also compare the current simulation results with those of MD simulation. In Figure 4-7, it is shown that the wave propagation predicted by our method is almost identical to that predicted by MD simulation. Remember that the degree of freedom here in our FE model is only 1.4% of that of the corresponding MD model.

For a successful parallel code with consistency, the computation results should be independent on the number of processors. To test the independence of the simulation results on the number of processors, the same computation model is calculated with different numbers of processors (1, 4, 8, 16, 32, 64 processors). Figure 4-8 shows that for a varied number of processors, the results are exactly the same. Both Figure 4-7 and 4-8 demonstrate the stability of the parallel code.

Scalability

Scalability is a measurement that describes how the parallel code will perform with a varied number of processors. The scalability test is conducted on the supercomputer Trestle of SDSC (San Diego Supercomputer Center). The supercomputer consists of

324 computer nodes. Each node contains 32 AMD- Magny-Cours processors. The clock speed of each processor is 2.4GHz.

In this section, two different testing cases are considered:

- (1) Scaled size test (ST): In this test, the number of elements per processor is kept to be 640. With the increasing of number of processors, the size of the whole model is increased proportionally. The total number of elements and the corresponding number of processors are listed in Table 4-1.
- (2) Fixed size test (FT): As shown in Table 4-1, in this test, the total number of elements is kept to be 5120. With an increased number of processors, the number of elements assigned to each processor is reduced.

From Figure 4-1, it is known that the total computational time is the sum of the time for steps INPUT, GETMAP, ELIST, EXCHANGE_U, CFORCE, CENTRAL and OUTPUT. Among these steps, EXCHANGE_U, CFORCE and CENTRAL are executed every time step. The union of these three steps is called SOLVE here. So the total computational time can be calculated using the following equation:

$$T_{total} = T_{INP} + T_{GETM} + T_{ELIST} + N \times T_{SOLV} + T_{OUT} \quad (4-1)$$

where N is the number of time steps.

The serial fraction γ is the ratio of computational time for the serial part to the total computational part when using a single processor. For this code, the serial fraction is

$$\gamma = \frac{T_{INP} + T_{GETM} + T_{PUT}}{T_{total}(1)} \quad (4-2)$$

The speedup, which describes how fast the computation would be by using multiple processors, is the ratio of the total computational time by a single processor to the computational time by multiple processors. The expression for speedup, in an ideal situation with no overhead in the parallel part, is

$$S(P) = \frac{T_{total}(1)}{T_{total}(P)} = \frac{1}{\gamma - \frac{1-\gamma}{P}} \quad (4-3)$$

The efficiency, which describes how efficient the code is when using multiple processors, is defined as the speedup normalized by the number of processors

$$E(P) = \frac{S(P)}{P} = \frac{T_{total}(1)}{PT_{total}(P)} \quad (4-4)$$

In general, most of the simulations involve several tens of thousands of time steps. In this work, only one thousand time steps are computed, since the speedup and efficiency converged for time steps greater than this number. For all these tests, the computational time for the key steps are listed in Tables 4-2 and 4-3.

Using Table 4-2, Table 4-3, and Equations (4-1) - (4-4), the speedup and efficiency for both FT and ST are computed and plotted in Figure 4-9 and Figure 4-10. It is seen from Figure 4-9 that the speedup for both FT and ST are very close to ideal speedup when the number of processors is less than 20. As the number of processors continues to increase, the speedup becomes slightly lower than the ideal one. It is worthwhile to mention that the number of elements is another issue which affects the speedup. Fewer elements per processor always leads to higher parallel efficiency. For ST, the number of elements assigned to each processor is fixed to be 640. For FT, the number of elements assigned to each processor decreases with the number of processors used. When 64 processors are used in the FT case, there are only 80 elements assigned to each processor. At this time, the communication and other overhead cost for FT is less than that of ST. Therefore, the speedup for FT is higher than for that of ST.

As shown in Figure 4-10, the efficiency for both FT and ST drops with an increasing number of processors. They are both very close to 1 when the number of processors is less than 8. Then as the number of processors increases, the efficiency

becomes lower and lower. Fortunately, the rate at which the efficiency drops becomes lower and lower as the number of processors increases further. Like speedup, the efficiency is also affected by the number of elements assigned to each processor. The efficiency of FT is higher than that of ST.

Summary

In this chapter, the parallel implementation of the new CG method was reviewed in detail. At first, the four design spaces of the parallel code were discussed. Then in the implementation of the code, the overall structure of the parallel code was introduced. The details of each key step of the parallel code were also introduced. Finally, the performance of the parallel code was tested on the supercomputer Trestle of SDSC (San Diego Supercomputer Center). A stress wave propagation example was used for the test. It was found that the CG simulation results were almost the same as those of the corresponding MD simulation. The simulation results were independent on the number of processors used. It was also found that the parallel code showed high efficiency in the FT and ST tests. As a conclusion, we found that the efficiency of the parallel code is related to the number of elements assigned to each processor. A large number of elements per processor always leads to slightly lower efficiency.

Table 4-1. Number of elements and processors used in ST and FT

Number of Processors	ST	FT
	Total number of elements	Total number of elements
4	2560	5120
8	5120	5120
16	10240	5120
32	20480	5120
64	40960	5120

Table 4-2. FT results (unit: second)

Procs.	INP	GETM	ELIST	EXC	SOLV	OUT
4	2.428	0.179	362.16	0.002	1.647	2.488
8	2.511	0.04	176.05	0.006	0.843	2.493
16	2.538	0.04	88.602	0.006	0.426	2.502
32	2.568	0.05	44.170	0.006	0.220	2.573
64	2.510	0.05	20.071	0.006	0.110	2.489

Table 4-3. ST results (unit: second)

Procs.	INP	GETM	ELIST	EXC	SOLV	OUT
4	1.226	0.04	163.130	0.002	0.814	1.247
8	2.511	0.04	176.052	0.006	0.843	2.493
16	5.068	0.05	184.924	0.006	0.906	5.062
32	10.021	0.05	188.210	0.007	0.988	10.108
64	19.009	0.05	193.587	0.006	1.076	19.684

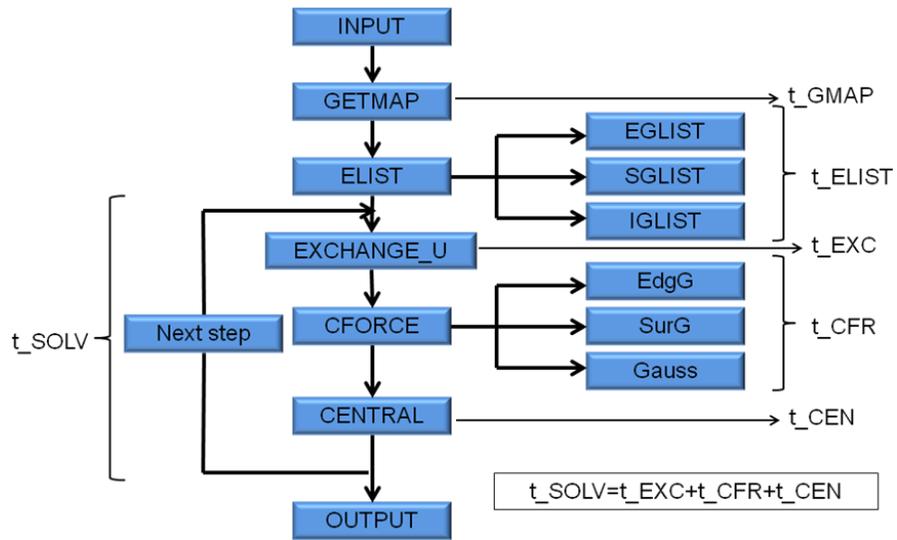


Figure 4-1. Structure of the parallel code

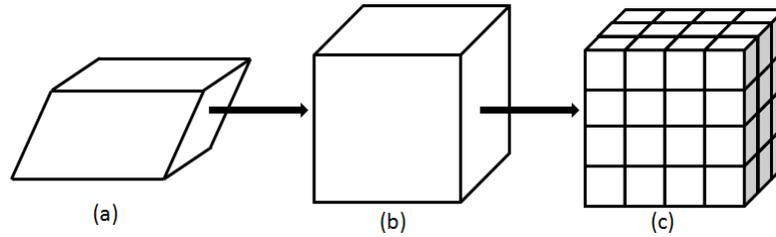


Figure 4-2. Procedures for domain decomposition (GETMAP)

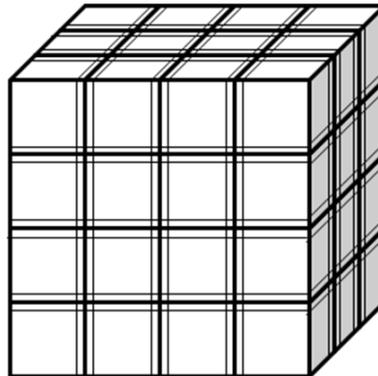


Figure 4-3. Interfaces between domains

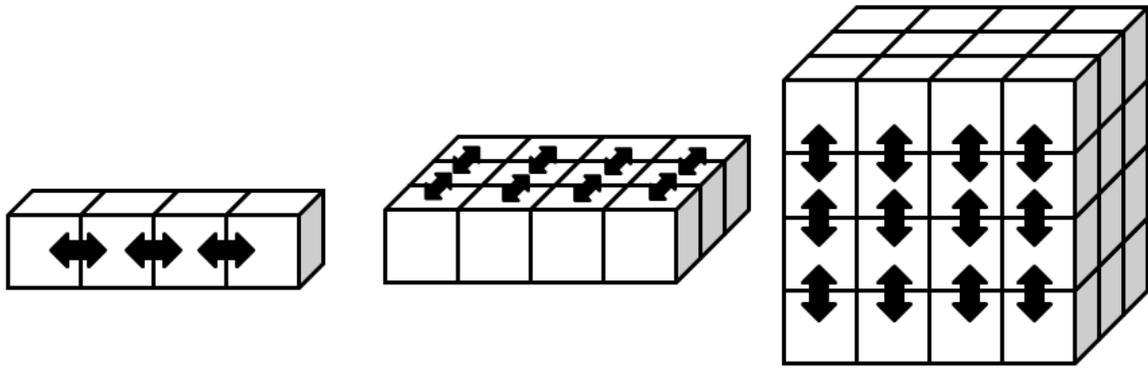


Figure 4-4. Communication between neighboring domains (EXCHANGE_U)

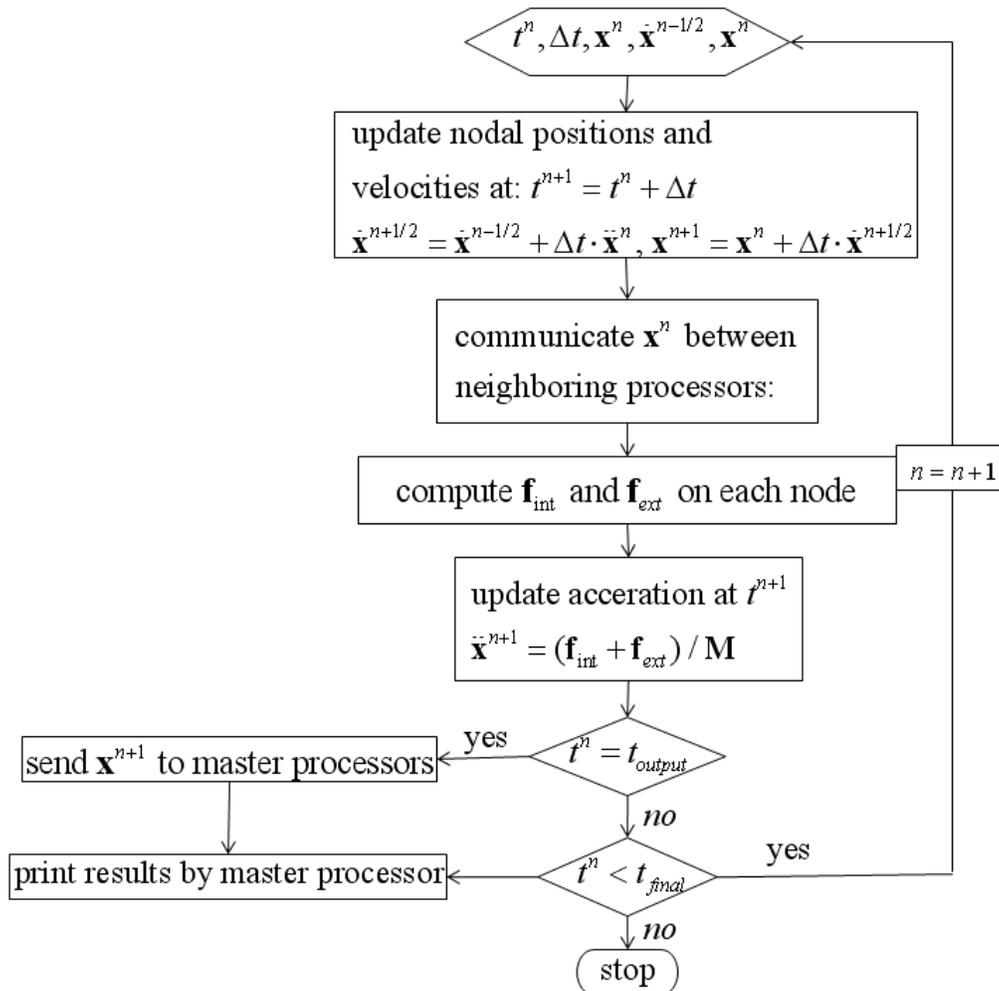


Figure 4-5. Central difference method

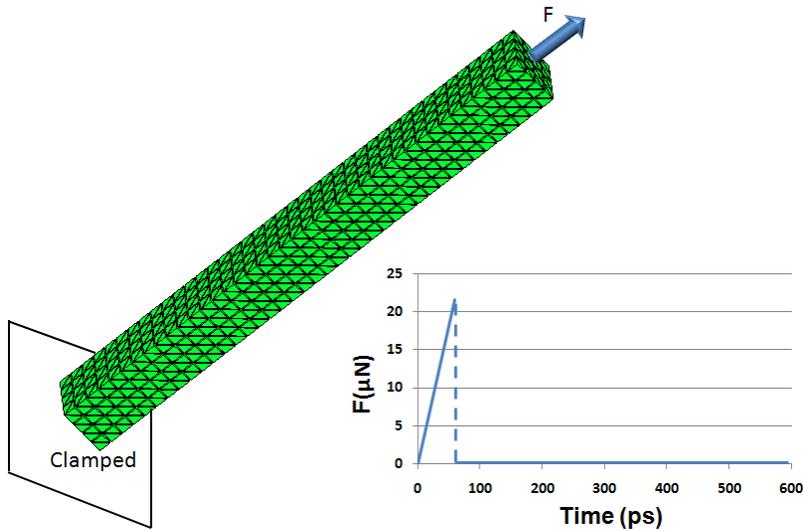


Figure 4-6. Setup of the model for the force perturbation test

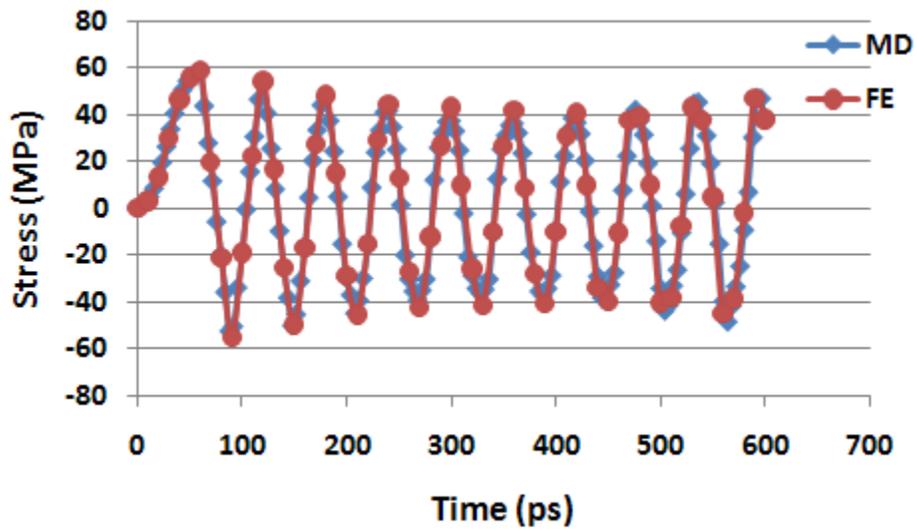


Figure 4-7. Vibration of a FCC bar predicted by the two different methods

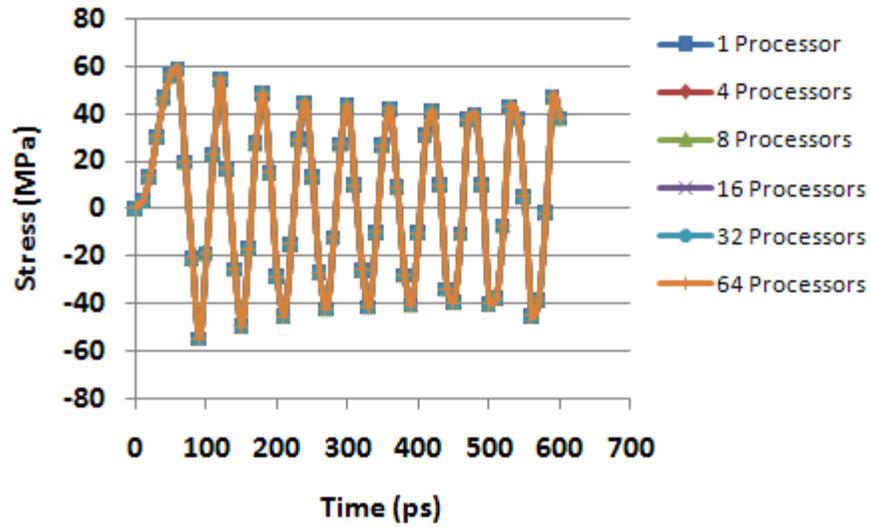


Figure 4-8. Vibration of the FCC bar predicted with varies number of processors

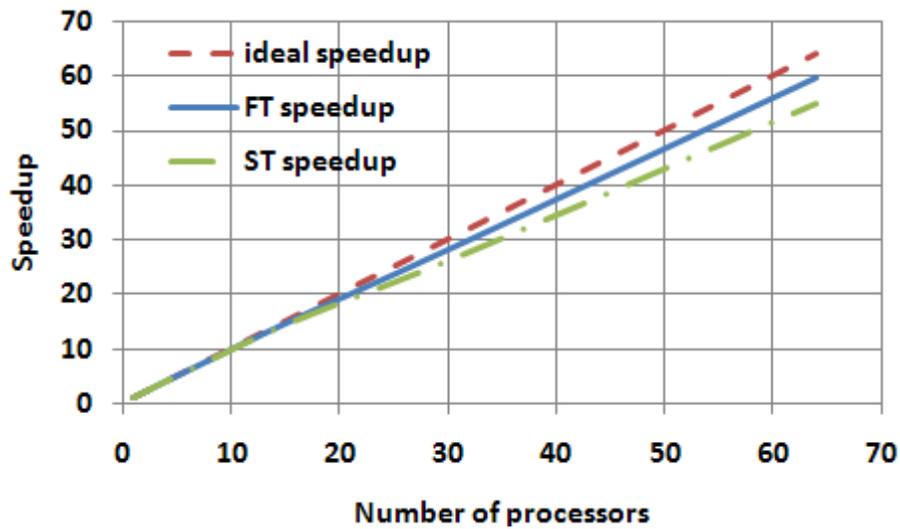


Figure 4-9. Speedup curves for FT and ST

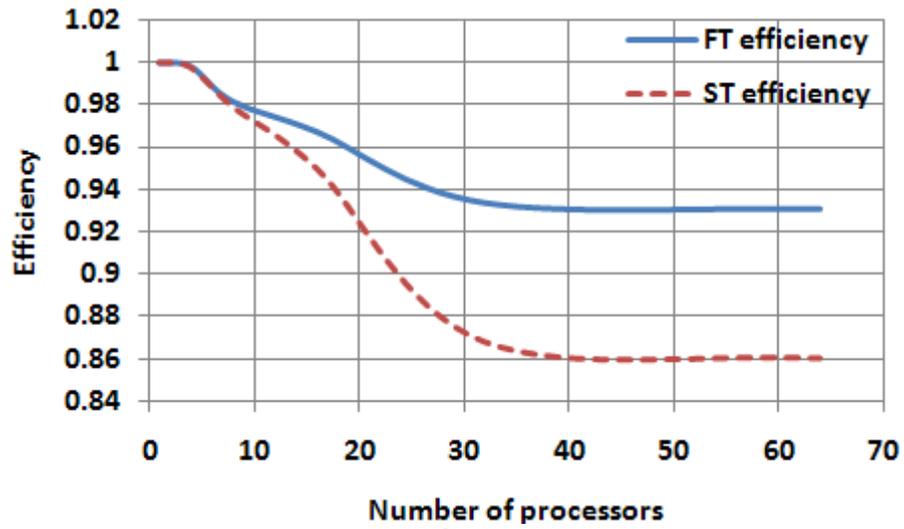


Figure 4-10. Efficiency curves for FT and ST

CHAPTER 5 SIMULATIONS OF 2D DYNAMIC BRITTLE FRACTURE

Introduction

In this chapter, the proposed coarse grained method (CG-AFT) is used to simulate the dynamic crack propagation in 2D brittle material. Firstly, the 2D triangular lattice model which has been widely used in the study of 2D brittle fracture is introduced. Then the crack propagation and branching is simulated using both CG and MD method. The comparison of their results are conducted and discussed. At last, as an application of the proposed CG-AFT method, five models with different size and loading rate are employed to investigate the interaction of stress waves and crack propagation at the atomic scale. As we have mentioned in Chapter 4, the CG-AFT method is fundamentally different from MD simulation approaches. So the exactly identifying the simulation results by the two different approaches is impractical. But we still can compare their simulation results because both of them are proposed to describe the nature of material at atomic scale.

The computational model

The first significant MD simulation of crack propagation in a 2D triangular lattice was described by Arhurst and Hoover (1976) in a landmark paper using about 1000 atoms. Large-scale MD simulations in parallel computers were performed about 20 years later, for examples, by Holian and Ravelo (1995), Zhou et al. (1996), Abraham et al. (1994, 1996, 1997, 1998, 2000), Buehler et al. (2003), Buehler and Gao (2006). In all of those simulations, the 2D triangular lattice model was again employed, with the computer models containing atoms from thousands to millions of atoms. Harmonic potentials or Lennard-Jones potential function were employed in most of the simulations

to describe the interaction between atoms. A common goal of all these simulations is to model dynamic fracture behavior with general features common to a large class of real physical systems rather than to a particular material.

The main objective of this chapter is to demonstrate the application of the new formulation and its finite element implementation. Therefore, in this work, we also use the 2D triangular lattice model in both the FE and the MD simulations. We employ the 2D computer model also out of our consideration of the influence of reflecting wave from the boundaries of the specimen. With the available computer power and the capability of our sequential computer code, we wish to generate a computer model that has the dimensions as large as possible so that waves emanated from the crack tip take some time before being reflected back by the boundaries of the specimen to interfere with the forward propagation waves.

The well-tested and well tuned Lennard-Jones (LJ) potential used in above-mentioned MD simulations is also employed in this work to represent an idealized “brittle” material. The LJ potential has a simple analytical form:

$$\phi_{LJ}(r) = 4\varepsilon((\sigma/r)^{12} - (\sigma/r)^6) \quad (5-1)$$

where r is the separation distance between two atoms, ε is the depth of the potential energy well and σ is the value of r where the potential become zero. In this work, we employ the normalized parameter system which has been used by Abraham et al. in 1996 and also Holian and Ravelo. By using this parameter system, all energy quantities are normalized by ε and all distance quantities are normalized by σ . For the sake of elabration, in the rest of this chapter, we may convert the quantities from this unit system to the physical unit system somewhere,

using $\varepsilon = 0.415\text{eV}$, $\sigma = 0.2277\text{nm}$ and $t_0 \approx 2\text{fs}$. Only the nearest neighbors of an atom is considered for computing its potential energy to mimic the “brittle” material.

The geometry of the computer model is shown in Figure 5-1. The rectangular slab of the 2D triangular lattice has the slab length and width L and W , respectively. A crack with length a is cut in the middle of the left boundary. The slab is loaded in model I in the direction perpendicular to the crack with a constant stain rate.

As shown in Figure 5-2, the 2D triangular element introduced in Chapter 3 is used to discretize the model. There are 32,060 triangular 2D elements and 96,180 nodes in the FE model shown in the figure. Each element contains 55 unit cells. The model corresponds to 1,767,200 atoms (~1.8 million atoms). The degrees of freedom to be solved in the FE model are about 5% of that in the MD model.

Simulation details and results

Comparison of CG And MD Simulations

In this section, two test cases for crack propagation are simulated using both the proposed FE method and MD simulation. Our goal of this section is to compare the results predicted through these two methods. As we have mentioned before, MD simulation has been widely used by the pioneers to investigate the brittle crack propagation and branching at atomic scale. Through this comparison, we are trying to show the reliability of the results predicted with the proposed new formula and its finite element implementation.

In case 1, the model described in Figure 5-2 is simulated using both of the two methods. The strain loading with the rate of about 10^8 per second (approximately converted from dimensionless unit system to physical unit system) is applied on both

the top and the bottom of the slab. The results are shown in Figure 5-3 and Figure 5-4. As predicted in both simulations, at about 1% of the applied strain, the crack begins to open up at the weakest spot, namely the lower tip of the crack or the right corner of the notch, where the stress is most concentrated. The kink observed in both of the two simulations shows that the crack is unstable at the beginning of its propagation. With the further increasing of the loading, a daughter crack is formed near the main crack. It is seen that the pattern of the crack propagation in the FE simulation does not exactly resemble that in MD simulation. Nevertheless, the FE model does have reproduced the strains for crack initiation and branching similar to that observed in MD simulation. Being unable to output the crack velocity history from LAMMPS, we do not output crack speed in this case. However, from the temporal sequence of the snapshot pictures of the propagating crack and the distance it traveled, we see that the crack propagation velocities in the FE model are close to that of the MD model.

The local stress distribution is also plotted in Figure 5-3. The local stress is calculated using virial stress formula. We understand that the virial stress formula is not appropriate for the description of local stress because it is not consistent with the concept of Cauchy stress in continuum mechanics and does not satisfy the continuum mechanics balance equations (Chen 2006). However, since LAMMPS output only local virial stress, for the purpose of comparison we also output the local virial stress in the FE simulation. It is seen from Figure 5-3 that the distribution of the local virial stress are similar in both simulations, but the maximum local stress obtained in the FE simulation is lower than that obtained in the MD simulation. Noticed that MD model contains about 1.8 millions of atoms and the stress is calculated per atom, while the stress in the FE

model is calculated per element and the specimen contains only 32060 elements. Therefore, the MD model has a resolution that is significantly higher than that of the FE model. As a result, the pictures taken from MD simulation have much higher resolution and the maximum stresses are also higher.

Figure 5-4 shows the average stress–strain curve of the specimen during the processes of tensile loading obtained by the FE and the MD simulations. It is seen that the magnitude of the stress obtained by the FE simulations agrees well with that obtained by MD simulation. However, there is noticeable discrepancy between the two stress-strain curves, especially between the areas that each curve covers. This implies that the FE model is more brittle than the MD model and underestimates the ductility and the work of fracture of the material. We believe that this considerable discrepancy is caused by the assumption of the linear shape function made in the finite element implementation.

In case 2, we use the same geometric model as in case 1. However, the cutoff for interatomic interaction is changed from 0.57nm to 0.34nm. The material is thus more brittle than that in case 1. We also reduce the applied strain rate from 10^8 per second in case 1 to 1.8×10^7 per second in case 2. A temporal sequence of snapshot pictures for the crack dynamics as well as local stress distribution by MD and FE are shown in Figure 5-5. Note that we use fixed color bars for both MD and FE results. It is found, this time, that with the more brittle material and slower loading rate the crack starts with stable, straight ahead propagating at the beginning. As the applied strain loading approaching 1.54%, the crack starts to branch. It is worthwhile to mention that in both of the two cases the crack always starts to propagate at a critical strain about 1%. This

means that the loading rate effect is not so significant to the start of crack propagation as to the start of crack branching.

Figure 5-6 shows the local stress distributions in front of the crack tip at the time when the crack starts to propagate. It is seen that the stress distributions in these two models are very close. The MD curve consists of 132 points, while there is only 16 points for the FE curve. As a result of the different resolutions in the MD model and the FE model, the stress at the crack tip is lower in the FE model.

Large-Scale FE Simulations

We present in this section a series of FE simulations with relatively larger computer models with the same idealized brittle material described in previous section. The goal is to investigate the effect of the interaction of reflected waves with the propagating crack. We simulate the dynamic responses of five specimens that have different size, shape, crack length, and loading rate. The details of the 5 models are given in Table 5-1.

Figure 5-7 shows the local stress distributions in front of the crack tip for the 5 models measured in the time interval during which the cracks start to propagate. We see that the 5 models predict almost the same critical stress value for crack initiation independent of the specimen size and the crack length. We also calculate the stress intensity factor, K_{IC} , for each model. The results are shown in Table 5-2, from which we find a nearly identical K_{IC} value for all the 5 models.

In Figure 5-8, we present the temporal sequence of snapshot pictures of the dynamic process of crack initiation, propagation, and branching for model E, the largest specimen (1.127 μm by 0.708 μm). We see in Figure 5-8(a) that the stress waves are

emitting from the crack tip when the crack starts to propagate, and that the waves are undisturbed. However, in Figure 5-8(b), it is seen that the stress waves have been reflected back from the horizontal boundaries and have interfered with the forward propagating crack. As a result, the stress waves are disturbed, and the crack starts to branch. We see in Figure 5-8(c) that stress waves are more disturbed and the crack has branched into two cracks. In Figure 5-8(d) we observe that each of the two branched cracks further branches into two cracks, which seems to be also resulted from the interaction of propagating crack with stress waves being reflected back from the horizontal boundaries. Two newly branched cracks soon arrest, leaving two main cracks continue to propagate. This is why we see kinks in Figure 5-8(e) and (f), and the two main branches propagate to the right boundary of the specimen in a zigzag pattern.

We plot the time history of the crack speed in Figure 5-9 for the period of crack initiation and first branching, in which c_R is the Rayleigh wave speed. It is seen that the first sharp increase in crack speed occurs when the reflected waves meet with the forward propagating stress waves at the strain around 0.58% (cf. Figure 5-8(b)). Then, we observe a sharp drop of the crack speed after the branching (cf. Figure 5-9). Point A and B correspond to the crack initiation and the first crack branching. We see before and after branching events, the crack speed oscillates at about 0.4 Rayleigh speed. From Figure 5-8 and Figure 5-9, we find that the sudden increase of crack speed is caused by the interaction of forward stress waves with reflected stress waves, which in turn cause the crack to branch, namely the dynamic instability of brittle fracture.

Figure 5-10 presents the snapshot pictures taken at approximately the time of the first crack branching for all the five models. The left are zoomed in pictures at the crack

tips. Again, it is seen that cracks, in all the five models, branch as a direct result of the interaction between the reflected stress waves and the forward propagating cracks.

Table 5-3 shows the time for different waves to emit from and travel back to the crack tip and also the time interval (the output time interval in the FE simulation) during which the crack branching occurs. Here, t_{left}^l , t_{left}^s and t_{left}^R stands for the time for longitudinal waves, shear waves, and Rayleigh waves to emit from and travel back to crack tip, respectively. It is found that in all of the five models, the longitudinal waves always arrive at the crack tip slightly before the crack branching. The shear waves and Rayleigh waves arrive at the crack tip after the crack branching, and in all the five models it is the longitudinal waves that causes the crack speed to rapid increase and subsequently the cracks to branch.

Summary

We have presented in this chapter a series of 2D simulation of the dynamic fracture. Using the 2D triangular linear finite element introduced in Section 3.2, a 2D computational model with about 5% degrees of freedom of the corresponding MD model was constructed. By comparing the coarse grained and MD simulation results, we have showed that the coarse grained method based on the atomistic formulation can predict the dynamic evolution of the crack tip stress, crack propagation speed, crack path and branching, qualitatively and quantitatively similar with atomic-level molecular dynamics simulations. Form the larger-scale model (~7 millions of atoms), we have found that the interaction of stress waves emitting from the propagating crack tip and the stress waves reflected back form the boundaries has a strongly influence on the crack speed, crack path and crack branching,

Besides the given temperature limitation, the linear shape function interpolation used here is also a factor that affects the accuracy of the computations. As we know, at the crack tip the displacement field always has very high gradient. Thus, some more sophisticated shape functions are needed to improve the accuracy of the computation.

Table 5-1. Geometric parameters and loading rates for the six models

Models	A	B	C	D	E
Specimen length (Lx) (nm)	1127	1127	1127	566	1127
Specimen width (Ly) (nm)	354	354	354	354	708
Crack length (a) (nm)	56	225	450	225	450
Number of nodes	384,480	384,480	384,480	192,480	768,960
Number of elements	128,160	128,160	128,160	64,160	256,320
Number of atoms	7,048,800	7,048,800	7,048,800	3,528,800	14,097,600
Applied strain rate (1/s)	4.6×10^6	4.6×10^6	4.6×10^6	9.2×10^6	4.6×10^6

Table 5-2. K_{IC} for models A~E

Models	A	B	C	D	E
$K_{IC} (MPa\sqrt{m})$	1.77	1.71	1.71	1.73	1.65

Table 5-3. Time interval during which crack branching occurs and time of arrival of reflected stress waves

Models	A	B	C	D	E
Time interval for branching (ps)	80~120	160~200	160~200	160~200	320~360
Time of arrival of Longitudinal wave	t_{left}^l (ps)	243	464	249	828
	$t_{up-down}^l$ (ps)	176	176	176	352
Time of arrival of Shear wave	t_{left}^s (ps)	421	803	431	1433
	$t_{up-down}^s$ (ps)	305	305	305	609
Time of arrival of Rayleigh wave	t_{left}^R (ps)	453	863	463	1541
	$t_{up-down}^R$ (ps)	328	328	328	655

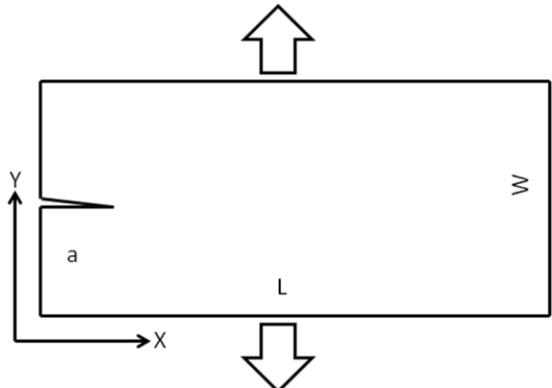


Figure 5-1. Simulation Model and Boundary Conditions

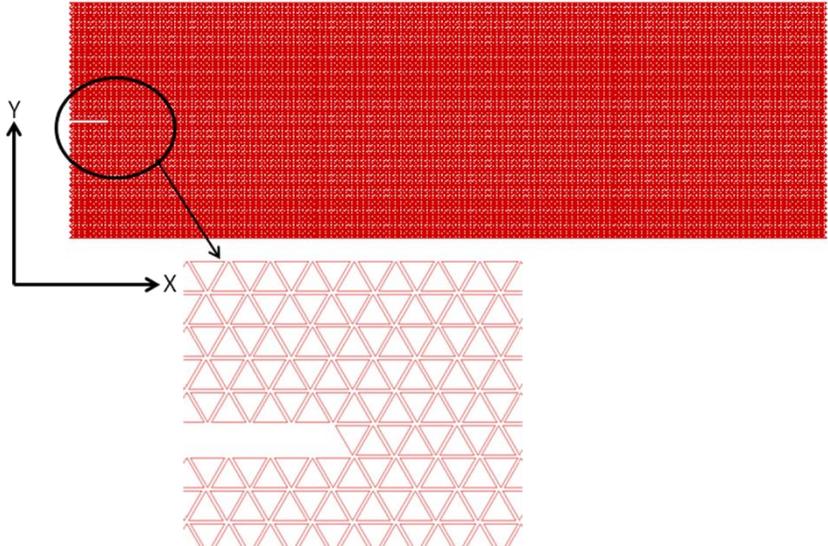


Figure 5-2. The finite element mesh of the computer model

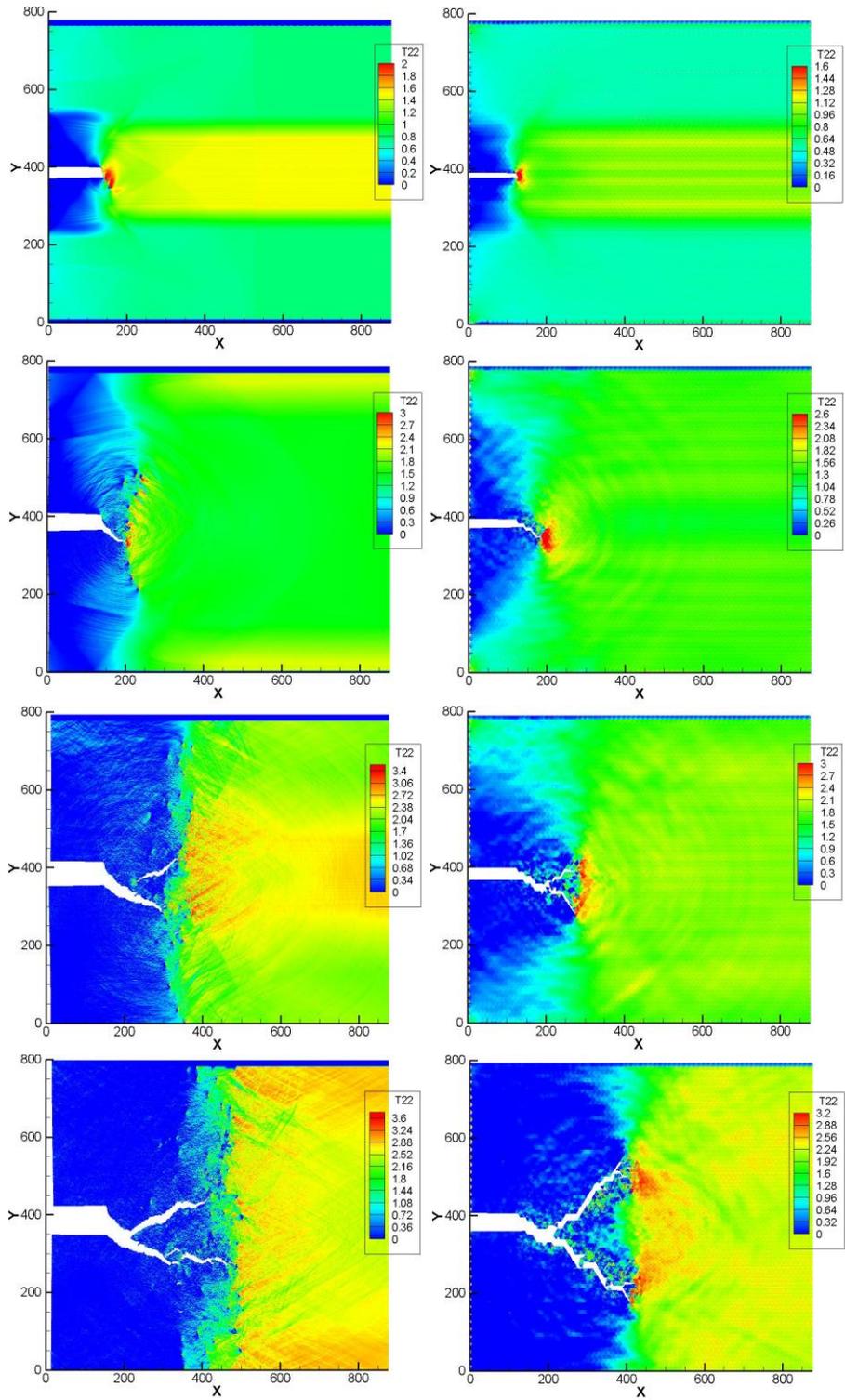


Figure 5-3. Comparison between FE and MD simulation results (deformation and stress distribution) for case 1. Left: MD simulation results (~1.8 million atoms), Right: FE simulation results (~32K elements)

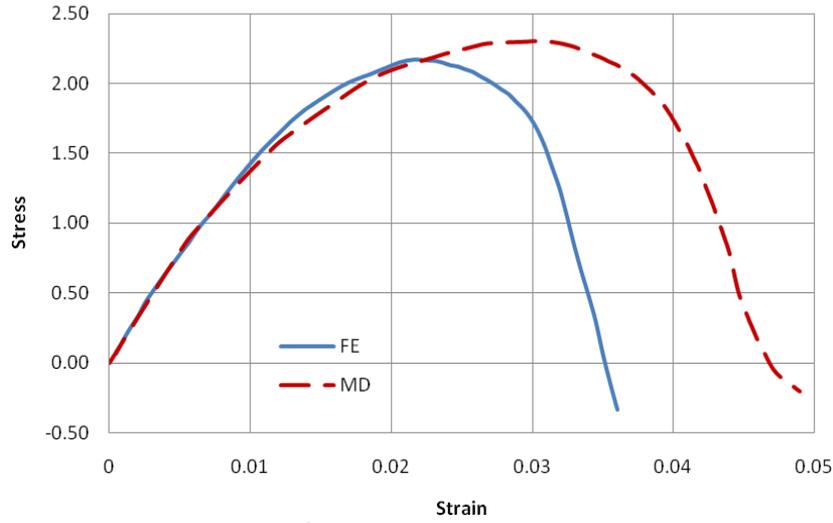


Figure 5-4. Average stress vs. strain curves

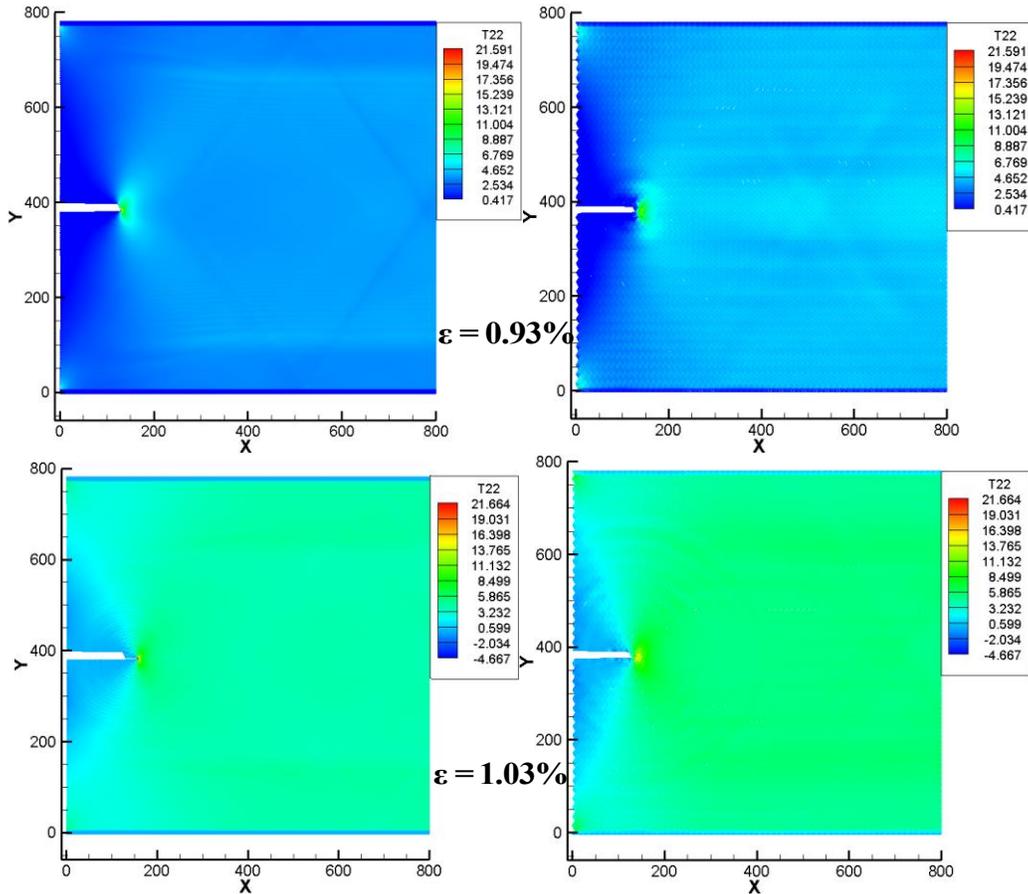


Figure 5-5. Comparison of FE and MD simulation results (deformation and stress distribution) for case 2 Left: MD simulation results (~1.8 million atoms), Right: FE simulation results (~32K elements)

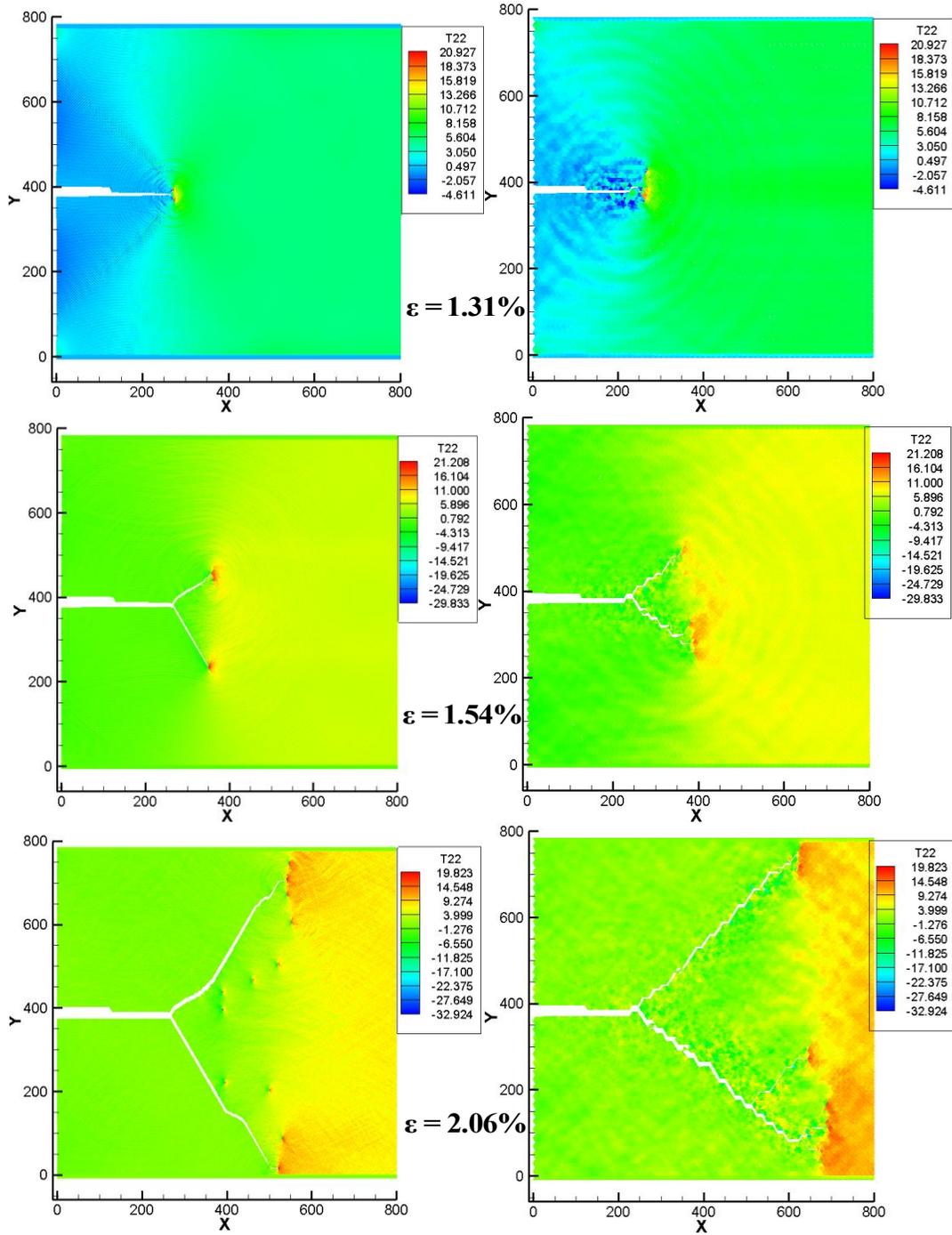


Figure 5-5. Continued

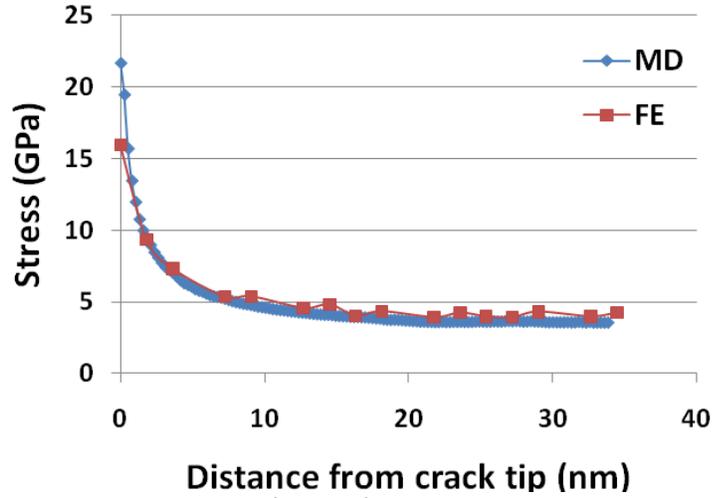


Figure 5-6. Stress distribution in front of crack tip when crack start to propagate

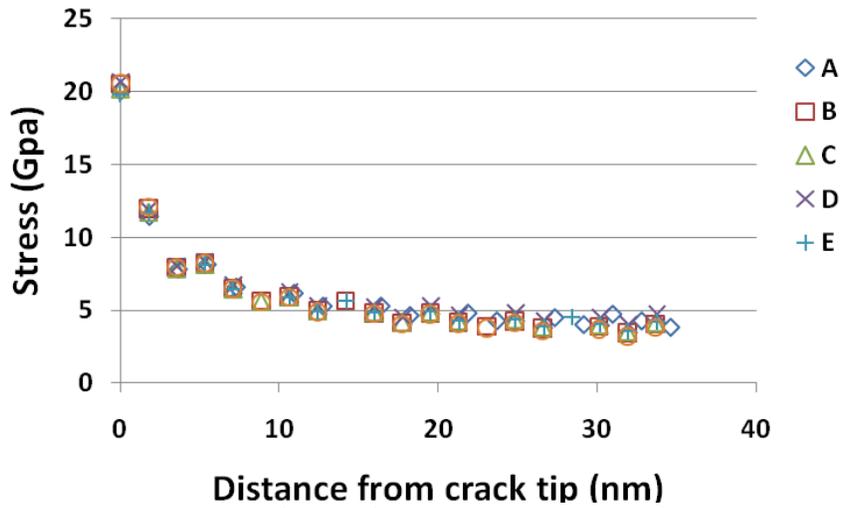


Figure 5-7. Stress distribution in front of crack tip when crack start to propagate

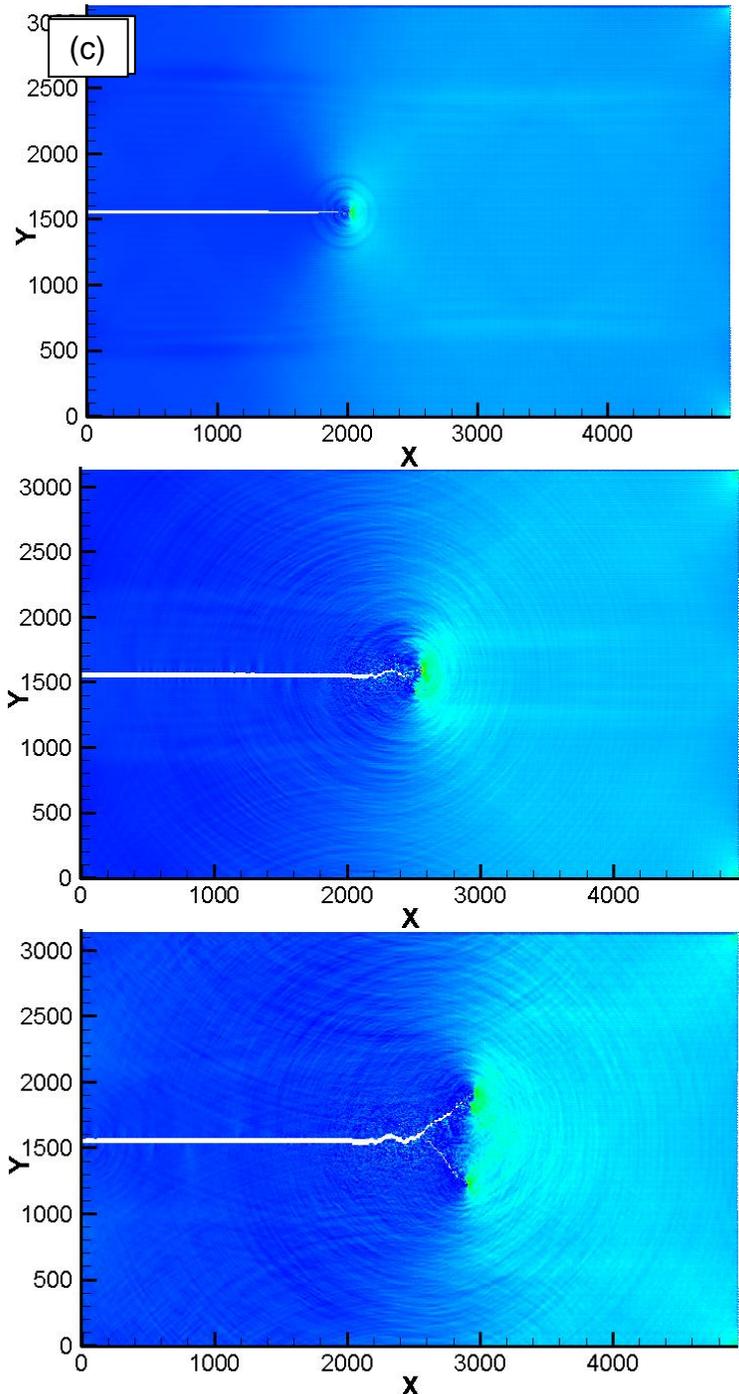


Figure 5-8. FE results of dynamic crack propagation. The pictures are taken at strain (a) 0.49%, (b) 0.58%, (c) 0.75%, (d) 0.97%, (e) 1.07%, (f) 1.18%

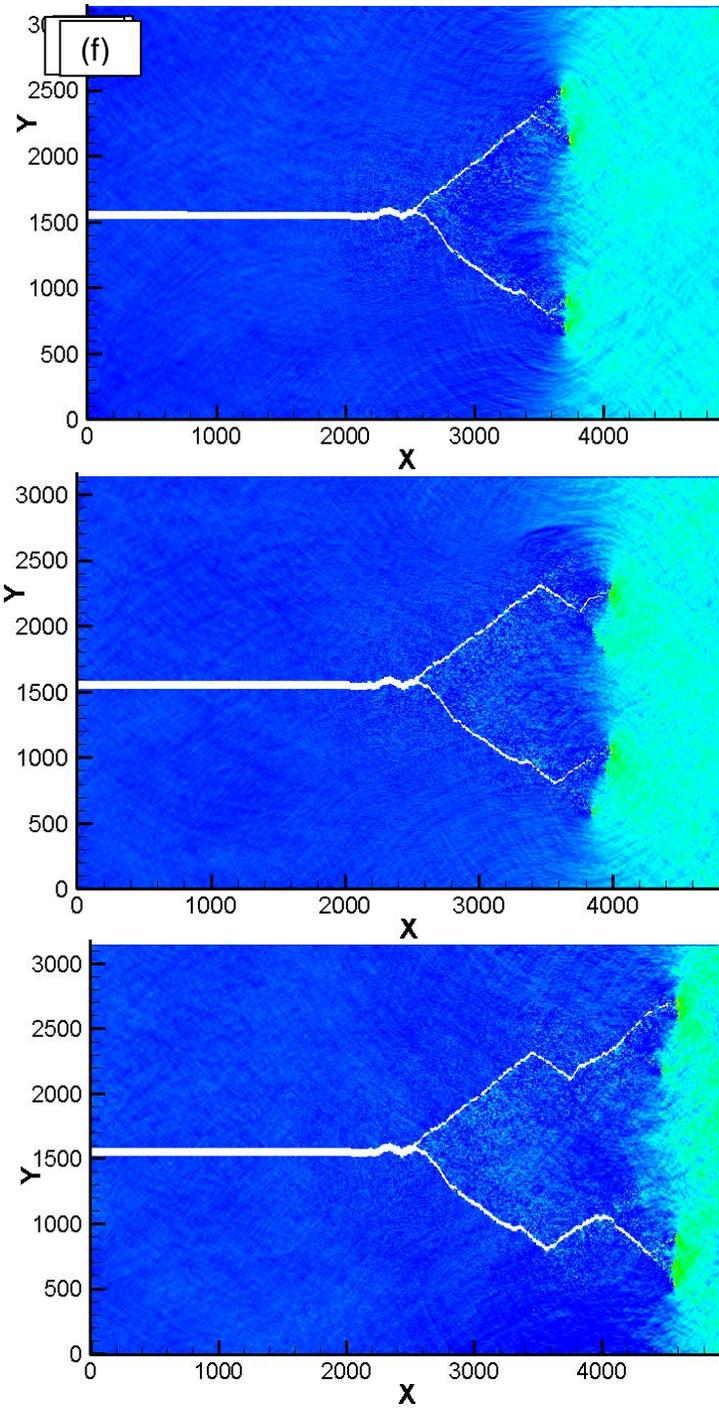


Figure 5-8. Continued

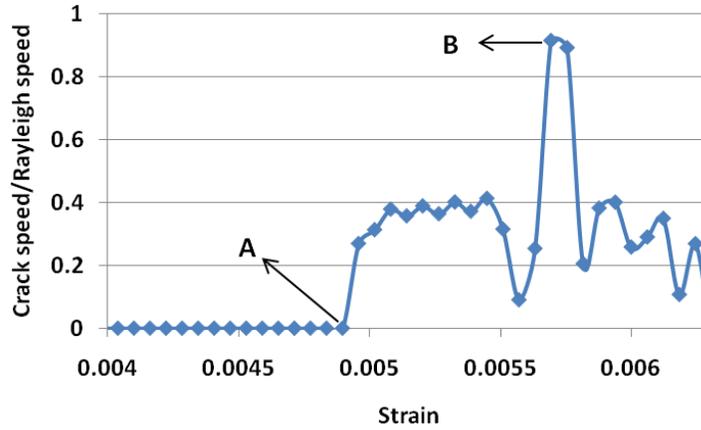


Figure 5-9. Crack speed measured for model E during the period of crack initiation and first branching

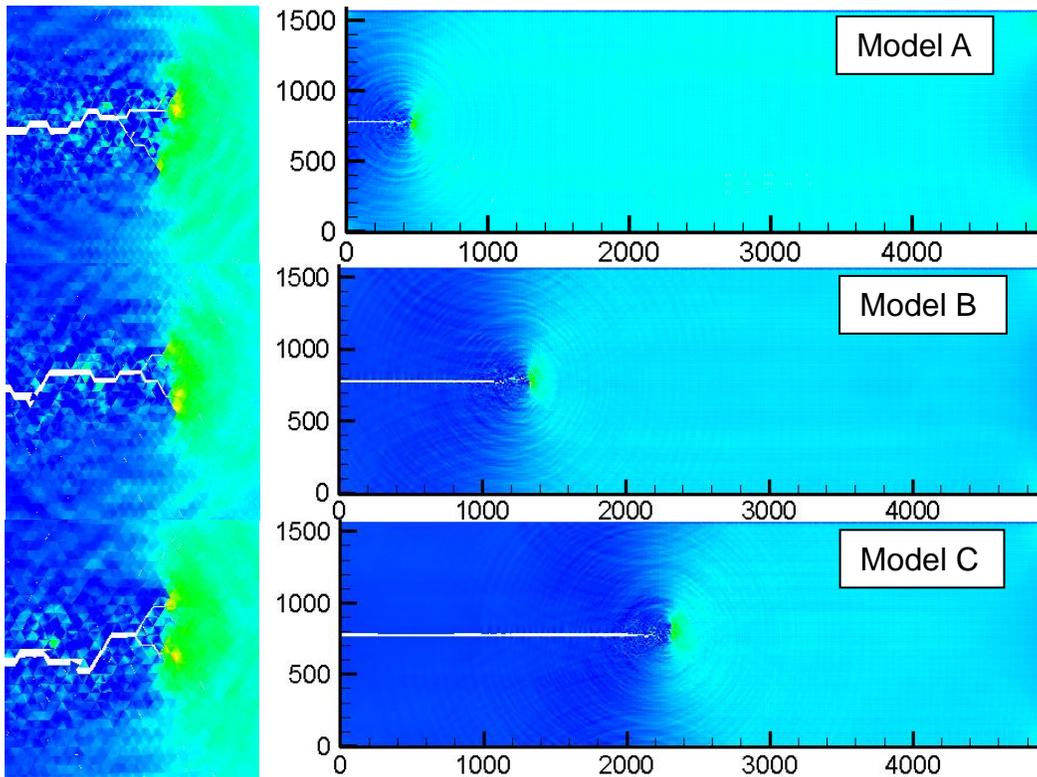


Figure 5-10. Stress waves at the time of crack branching in models A-E (Left: zoomed in the crack tip).

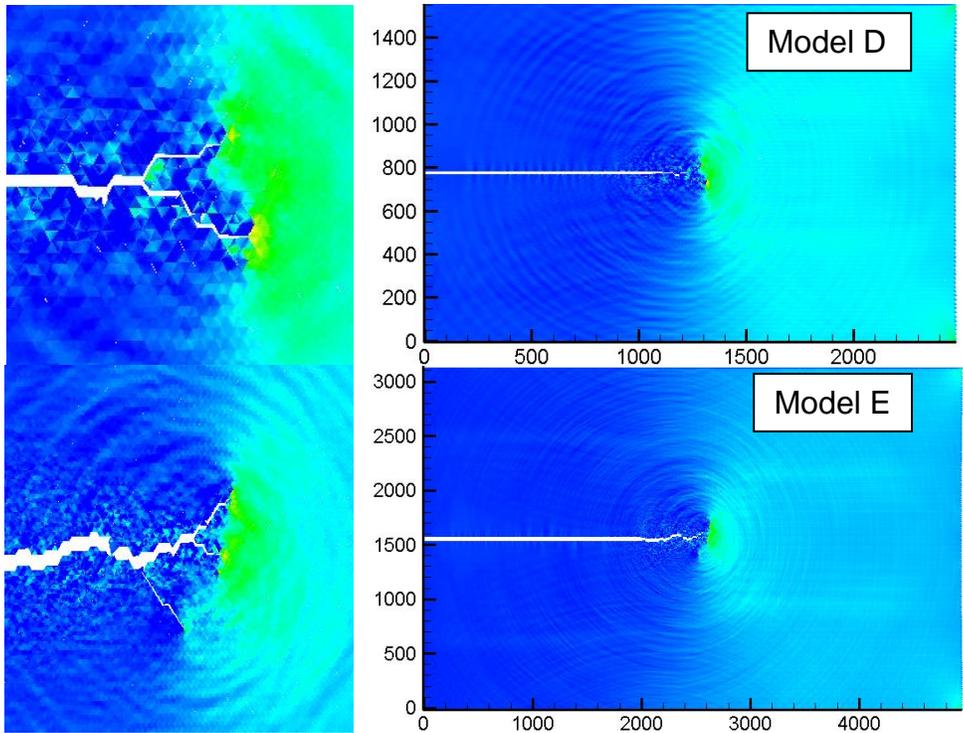


Figure 5-10. Continued

CHAPTER 6 SIMULATIONS OF 3D DYNAMIC BRITTLE FRACTURE

Introduction

Three dimensional dynamic fractures are much more complex than two dimensional cases not only because of the addition of one dimension, but also because of the new phenomena that would be involved with the addition of the dimension. On the other hand, the analytical solution for 3D dynamic fracture problems is always too difficult to derive. Thus a numerical simulation tool for 3D dynamic fracture is necessary. In the previous chapter, we used the new CG method to study 2D dynamic fracture problems. Because of the differences between 2D and 3D problems, we still need to evaluate the performance of the new CG method in simulating 3D dynamic fracture.

Firstly, we compare the results of CG and MD simulations to show that the results computed by the CG method are reasonable physically. It is worthwhile to mention that there is a difference between the treatments of the temperature field in the CG and the MD simulations. In the CG simulation, temperature is an independent variable which can be determined by solving the conservation of energy equation. In this work, as a special case, temperature is kept to be 0 K throughout the CG simulation. While in the MD simulation under the NVE ensemble, temperature is coupled with the velocity of each atom in the system. Even if we start with a homogeneous 0 K temperature, the temperature could be very high and inhomogeneously distributed after the simulation. The NVT or NPT ensemble is always employed for the sake of controlling the average temperature. However, neither the NVT nor NPT algorithm is suitable for the comparison here due to the numerical dumping that they would bring. This dumping can significantly change the wave speed in the model. Therefore, in this chapter, we chose

the NVE ensemble in the MD simulation. When the system temperature in the MD simulation is low, the two simulation results are comparable.

Next, a series of CG simulations, including 3D crack branching, impact tests, and 3D crack surface evolution, are performed. Through these simulations, we want to show that the new CG method can predict the arbitrary dynamic crack propagation in crystals. In the last example, 512 processors is used for a system composed of 0.11 million elements. With this example, the performance of the parallel code in handling large-scale simulations is also demonstrated.

Comparison of CG and MD simulations

Crack propagation is associated with the loss of material stability which ushers in a host of numerical challenges. In this section, the 3D crack propagation in an FCC single crystal film is simulated using both CG and MD methods. Through the comparison of their results, we want to show the reliability of the CG method in predicting 3D dynamic fracture.

In this example, the robustness of the CG method is demonstrated through the simulation of 3D dynamic fracture. As shown in Figure 6-1, the computational model is a notched thin film which is discretized into 1590 hybrid elements. The thin film is fixed at the bottom and a constant velocity, 20m/s, is applied to the top surface in the direction shown by the arrow. For the purpose of comparison, an MD model with the same size, shape and boundary conditions is computed using LAMMPS. Since each hybrid element used here is equivalent to 1000 atoms, the MD model contains 1,590,000 atoms.

Figure 6-2 (a) and (b) show local stress (σ_{yy}) distributions for the two models

when the average strain ε_{yy} is 1.9%. At this stage, there is no crack propagation and the material stability is still maintained. Both of the two pictures (Figure 6-2 (a) and (b)) show that there are two stress concentration regions. One is at the crack tip; the other is at the top-right corner of the model. The comparison of these two pictures shows that, before crack propagation, the CG model predicts quite similar local material behavior to the corresponding MD model.

When the crack starts to propagate, because of the stress concentration, the crack path will bend up to the top right corner of the model. As mentioned in section 3, the crack propagation always prefers the weak surfaces in crystals. The results shown in Figure 6-2 (c) and (d) confirm this point. Both the CG result (Figure 6-2 (c)) and the MD result (Figure 6-2 (d)) show that the crack path bends up to the top right region of the model with an angle of 60° . This 60° angle corresponds to one of the slip planes of the material.

Note that the differences between the CG result and the MD result are caused by the different treatment of temperature. As mentioned before, the system temperature is kept to 0 K throughout the CG simulation. Before crack propagation ($\varepsilon_{yy} \leq 1.9\%$) the temperature of the MD model is less than 10 K. It is reasonable that the MD result is similar to the CG result at such a low temperature. After crack propagation, the temperature of the MD model increases suddenly. At the stage of $\varepsilon_{yy} = 2.9\%$, the temperature of the MD model is over 100k. So the CG result is different from the MD result as shown in Figure 6-2 (c) and (d).

Figure 6-3 shows the average stress-strain curves calculated by the CG and the MD simulations. It is found that the two curves are very close to each other before they

reach their respective peaks corresponding to the beginning of crack propagation. The staircases of these two curves are caused by stress wave propagation in the specimen. After crack propagation, the temperature of the MD model is much higher than that of the CG model. Thus the fracture in the CG model is more brittle because of the lower temperature. This is the reason why the CG curve drops earlier than the MD curve

It is worthwhile to mention that the crack propagation, especially when it becomes unstable, is very sensitive to temperature. Since the CG and the MD simulations are performed under different thermodynamic conditions, the crack paths predicted by the two methods are not exactly the same. To obtain closer results, temperature should not be assumed to be a constant in the CG simulation. This will lead to the addition of an unknown to the computation. To solve this unknown, the conservation of energy equation needs to be solved in combination with the balance of linear momentum equations.

Application of the CG method in 3D brittle fracture

3D Crack Branching Under Mixed-Mode Loading

In this section, the crack branching in an FCC single crystal is simulated. The computational model is shown in Figure 6-4 (a) - (c). The thin film is discretized into 8,944 hybrid elements, which is equivalent to 8.944 million atoms. The dimensions of the thin film are 325.4nm×108.5nm×1.8nm. Figure 6-4 (a) shows the front view of the model. A notch with the length of 130.2nm (40% of the length of the thin film) is cut in the middle of the film. $\sigma = 1.6GPa$ is the constant distributed force applied to the top and the bottom surfaces. Figure 6-4 (b) is the side view of the model. Figure 6-4 (c) is the magnified view of (b). From Figure 6-4 (b) and (c), it is seen that the loading will cause mixed-mode fracture in the specimen. The CG simulation is performed using 16

processors.

For this CG model, it is assumed that the crack propagation and branching always prefer the slip planes. This assumption is reasonable according to Dr. Gumbsch and his co-workers' findings. They found that the crack branching in FCC single crystals always follows the path of dislocations emitted from the crack tip (Gumbsch et al. 1997). They also explained that the dislocation nucleation distorts the arrangement of the atoms near the slip plane at the crack tip and creates a weak path for the crack to follow. In my work, an MD simulation is performed to demonstrate the validity of this conclusion for a mixed mode loading case. Because of the high cost of the MD simulation, the model is reduced to about one eighth of the CG model. Each dimension of the MD model is about half of that of the CG model, so there are about 1million atoms in the MD model. The MD simulation (Figure 6-5) shows that the crack starts to propagate stably at the beginning. Then it branches into the directions of $+60^\circ$ and -60° relative to the original path. These directions correspond to the preferred directions of dislocations. This result shows that the conclusion of Dr. Gumbsch and his co-workers is also valid for mixed-mode fracture problems.

As shown in Figure 6-6, the simulation result of the larger CG model is similar to the MD result shown in Figure 6-5. Although the size of the CG model and the MD model are different, the similar crack propagation pattern and the crack branching angle have revealed the nature of the crack branching in this problem. The dislocations emitted from the crack tip determine the paths for crack branching in this material.

Impact Fracture of a Plate

In this case, a more complex fracture problem is computed using the CG method. The applicability of the method to large deformations, fragmentation and multiple-crack

problems is conducted. The computational model for this impact test is shown in Figure 6-7. A rigid ball with a constant velocity V hits on the FCC single crystal plate. The FCC plate contains 40,000 hybrid elements, which are equivalent to 40,000,000 atoms. This plate is a very thin film with the dimensions of $511\text{nm}\times 511\text{nm}\times 1.8\text{nm}$. The rigid ball has a radius of 50nm . It moves towards the plate with a constant velocity of 100m/s . The simulations in this test were conducted with 64 processors.

The snapshots of this impact simulation are shown in Figure 6-8. The contour represents the displacement field DZ in the deformed body. DZ indicates the displacement normal to the plate surface and opposite to the moving direction of the rigid ball. As shown in Figure 6-8, when the rigid ball hit the plate, the plate fractured with six small cracks. Then the six cracks propagate in six directions with almost the same crack speed. The angle between each two neighboring crack branches is nearly 60° . The bottom-right picture in Figure 6-8 shows the oblique view of the deformed specimen at $t=200\text{ps}$.

Because of the large size of the CG model, the corresponding MD simulation is too computationally expensive to be performed. Therefore, in this work, a similar impact simulation on a much smaller model is conducted with MD. The MD model has the same thickness as the CG model. But the dimensions in the other two dimensions are only one tenth those of the CG model. Correspondingly, the size and the velocity of the rigid ball are also reduced to one tenth of the CG model. Although the size and the shape for these two models are different, their simulation results reveal the nature of the problem. From Figure 6-9, although the number of cracks is reduced to four, the fracture pattern of the plate is still similar to that in the CG result. The four cracks propagate

symmetrically and with almost the same velocity. The reason for the reduction in the number of cracks in the MD simulation is that the ratio of the thickness to the in-plane dimensions is ten times larger than that of the CG model. Since thicker plates dissipate more energy from an impact than do thinner plates, only four cracks are sufficient to dissipate the energy brought by the impact of the rigid ball.

3D Crack Surface Evolution

The objective of this example is two-fold. Firstly, we want to simulate 3D crack propagation in a single crystal block using the CG method. Secondly, we want to test the performance of the parallel code for large-scale simulation. The computational model is shown in Figure 6-10. A notched FCC rhombohedral block is fixed on the bottom and a constant surface loading, $\sigma = 1.6GPa$, is applied to the top of the model. The applied loading is set to be much larger than that for stable crack propagation. This computational model is composed of 1,545,348 nodes and 110,382 hybrid elements, which correspond to over 0.11 billion atoms. 512 processors are employed for the computation to test the performance of the parallel code in handling a large number of processors.

Figure 6-11 shows some snapshots of this complex crack surface propagation process. The contour in the pictures indicates the displacement field DY in the specimen. DY is the displacement along the direction of the applied force and normal to the top surface of the model. The snapshots show the progress of the 3D crack surface evolution under the effects of both stress waves and the crystal structure of the material. It is seen that the crack surface is smooth at the beginning of the crack propagation. Then because of the instability of the crack propagation, the crack surface becomes rougher and rougher. From the snapshots shown in Figure 6-11, it is also found that the

crack surface is smoother when close to the boundary of the specimen and rougher in the middle of the specimen. This different crack propagation pattern indicates the complex stress field within the 3D model.

Summary

We presented in this chapter a series of 3D dynamic fracture simulations using the hybrid element introduced in Chapter 3. As we know, losing part of the atomic information is inevitable for coarse graining. Then the question is how much confidence we would have when ignoring such atomic information. As we have mentioned in the first chapter, the fracture behavior of a single crystal material is closely related to its crystal structure. So it is necessary to maintain as much atomic information as we can in the CG model to obtain the most accurate results. In this chapter, it was shown that the utilization of the hybrid element is a possible way to achieve this goal.

By comparing the CG simulation results with the MD simulation results, we have shown that the new coarse grained method can predict the 3D crack tip stress, stable crack propagation, and unstable crack propagation, both qualitatively and quantitatively similar with the atomic-level MD simulations. It is worthwhile to mention that the difference between coarse grained and MD results comes from the different thermodynamic processes involved in the two different simulations. To obtain closer results for the two methods, the conservation of energy equation needs to be solved along with the balance of linear momentum equations.

As applications of the CG method, 3D crack branching under mixed-mode loading, 3D impact test, and 3D unstable crack surface evolution were simulated to show the robustness of the coarse grained method in dealing with large scale 3D dynamic fracture. In the last case, 3D crack surface evolution under Mode I loading, 512

processors were employed to compute a model with over 0.1 million elements, which is equivalent to over 0.1 billion atoms in the corresponded MD model. Through this case, the ability of the parallel code in handling a large-scale model with a large number of processors was demonstrated.

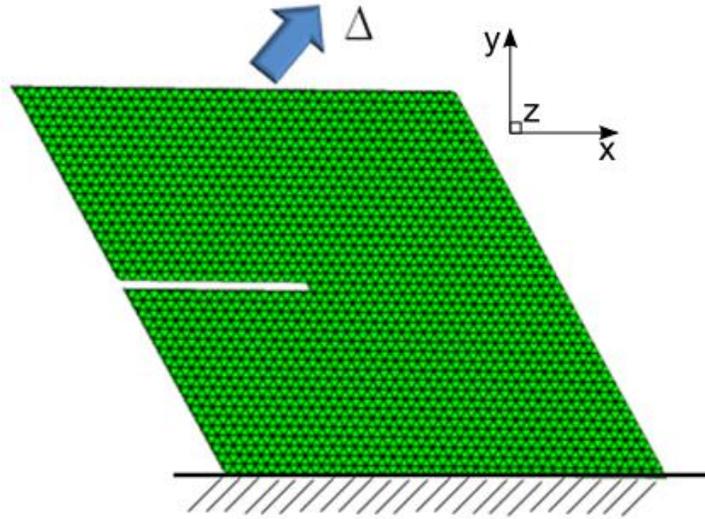


Figure 6-1. Notched film for the first test case

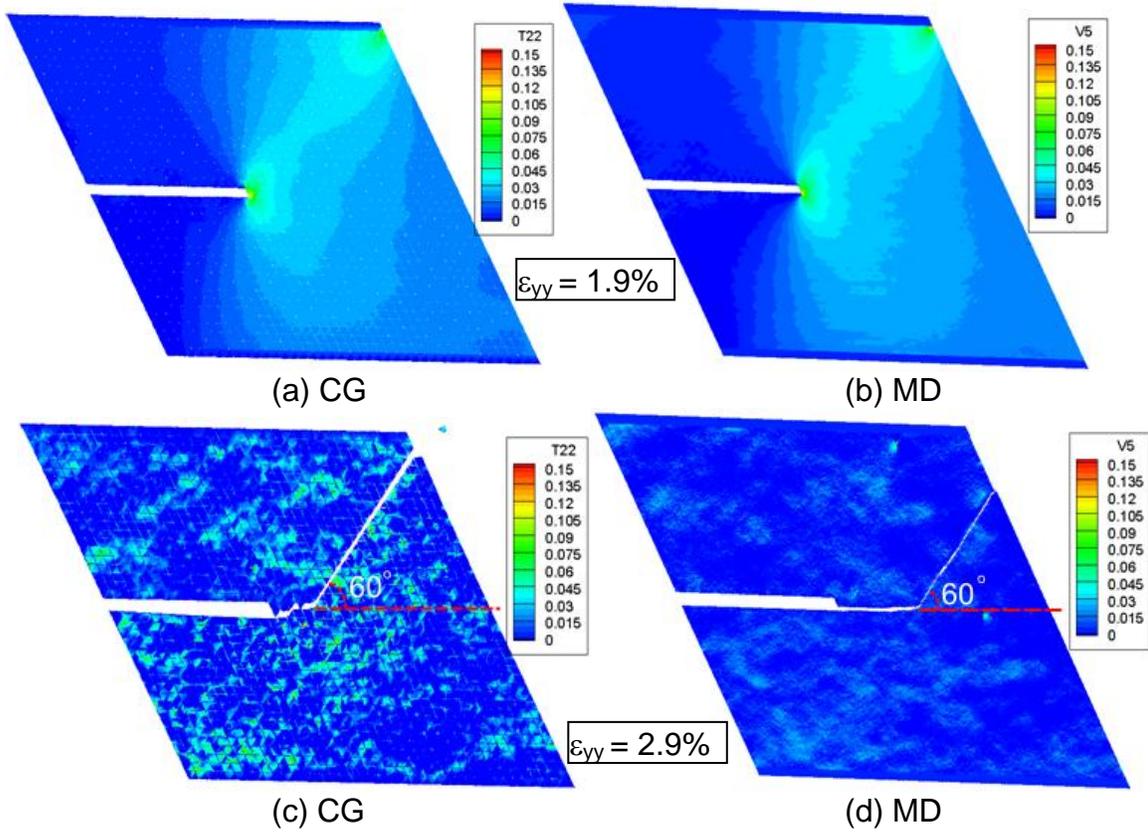


Figure 6-2. Comparison of CG and MD results for the first case

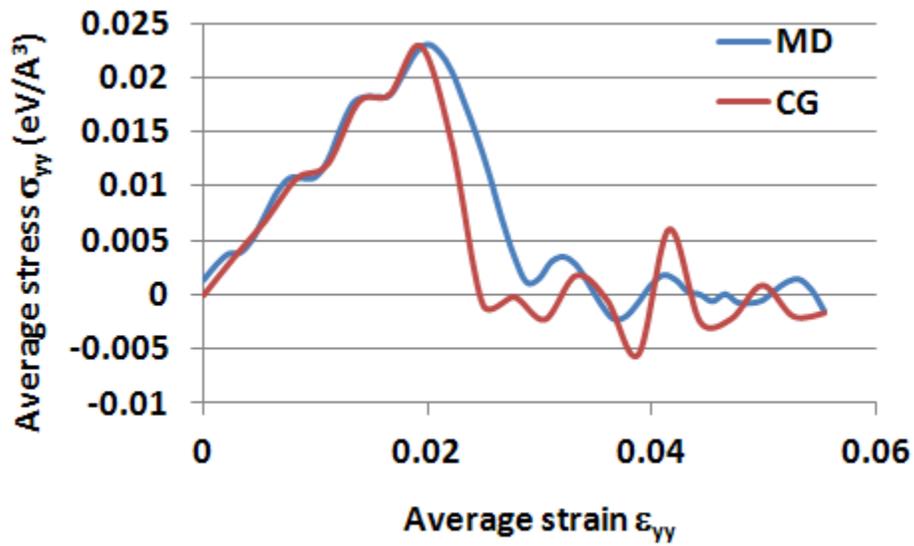


Figure 6-3. Comparison of the average stress - time curves

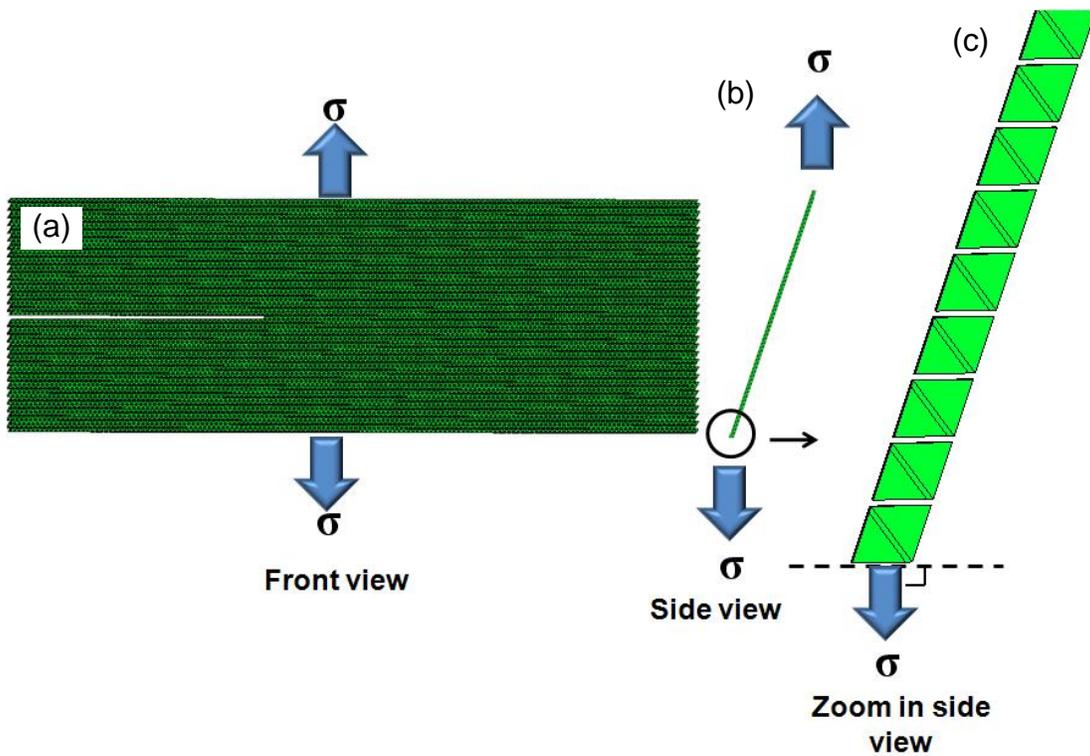


Figure 6-4. Computation model for 3D crack branching

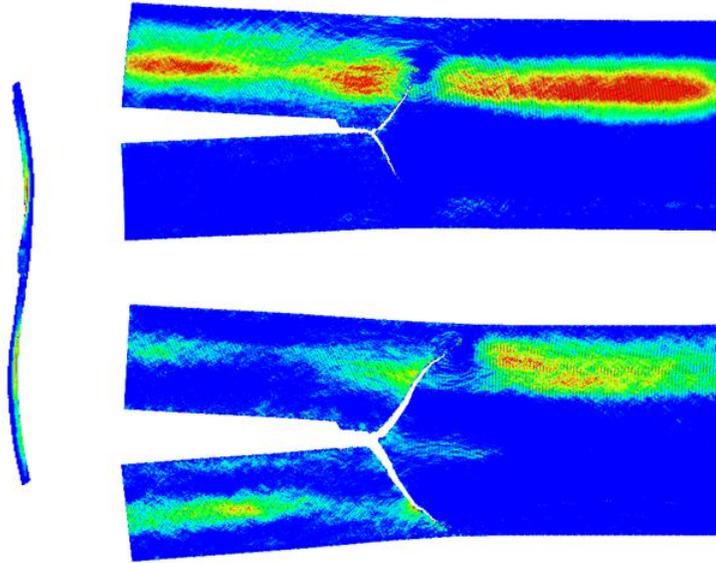


Figure 6-5. MD simulation of crack branching under mixed-mode loading

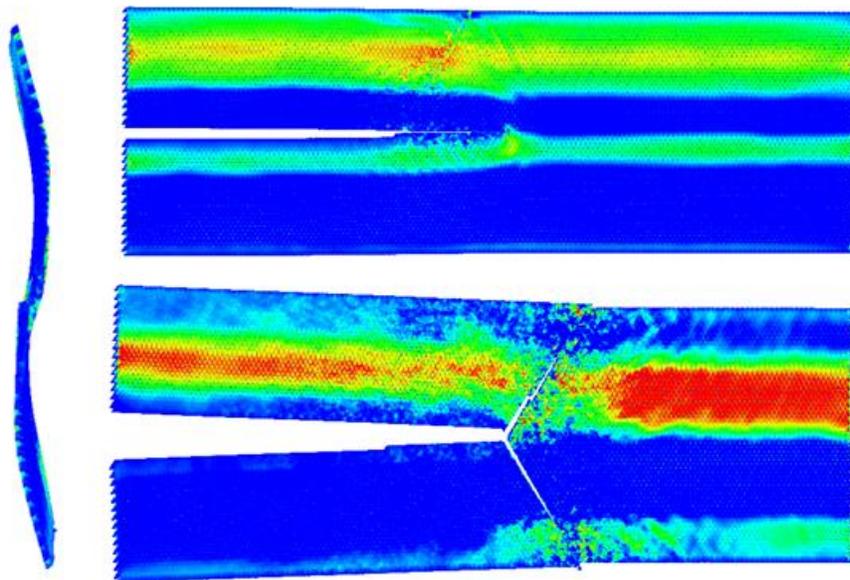


Figure 6-6. CG simulation of crack branching under mixed-mode loading.

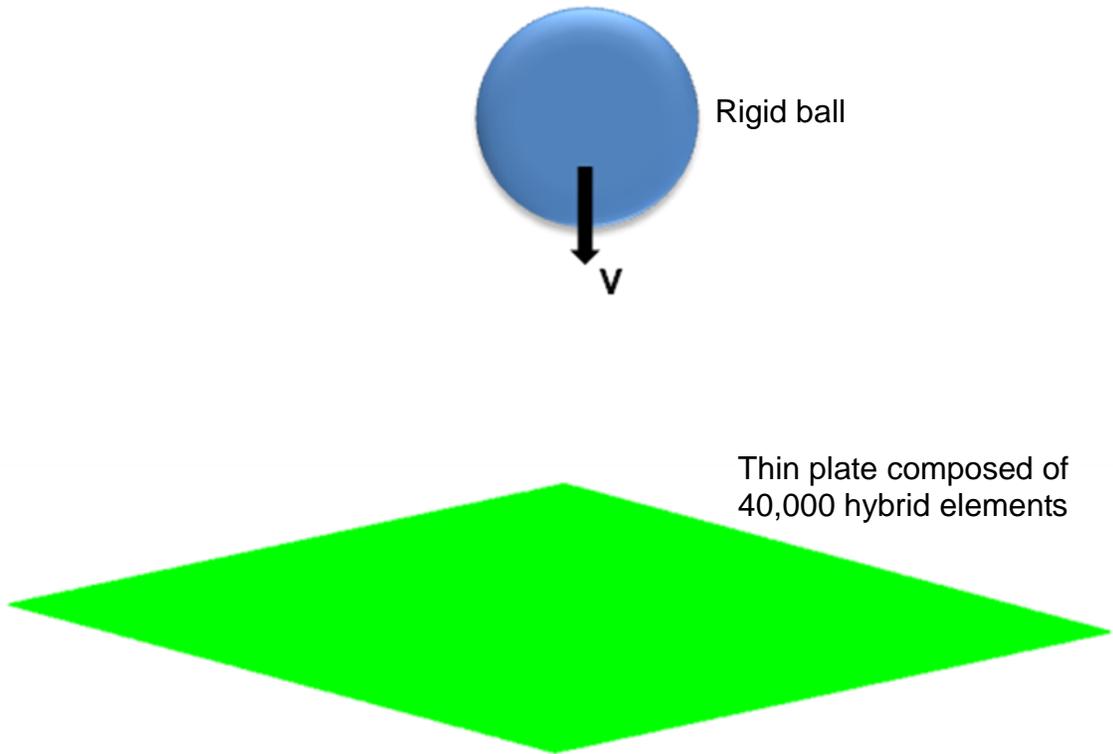


Figure 6-7. Computation model for 3D impact test

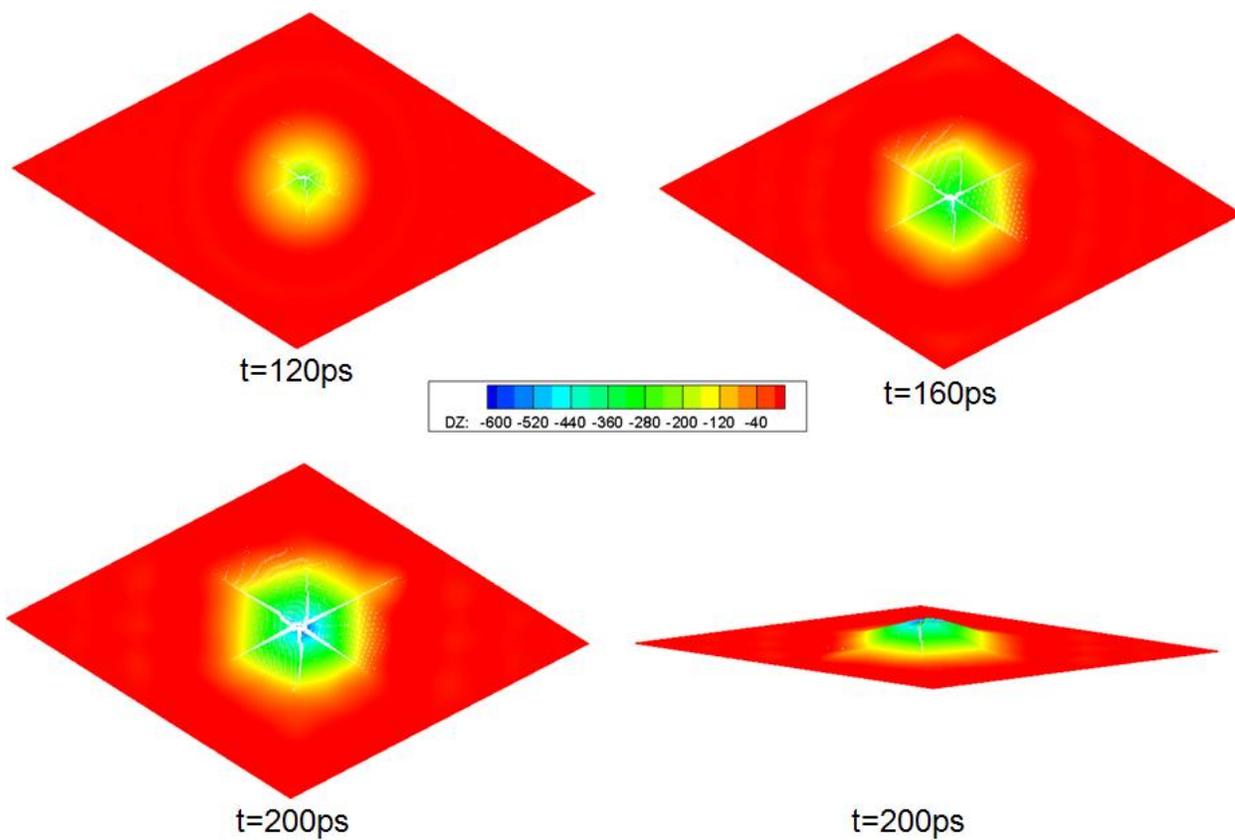


Figure 6-8. Fracture of the plate under the impact of a rigid ball

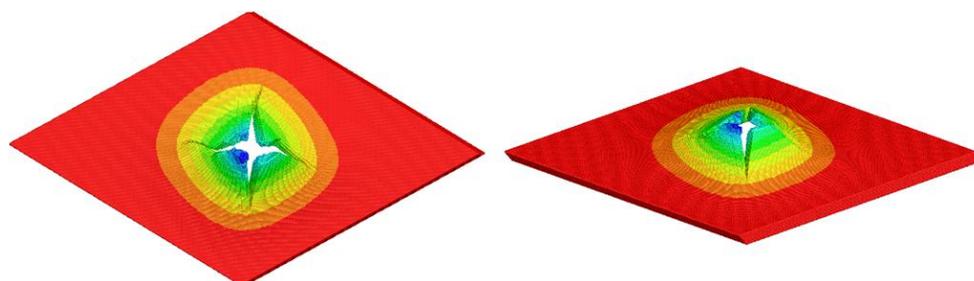


Figure 6-9. Impact test for a smaller plate by MD simulation

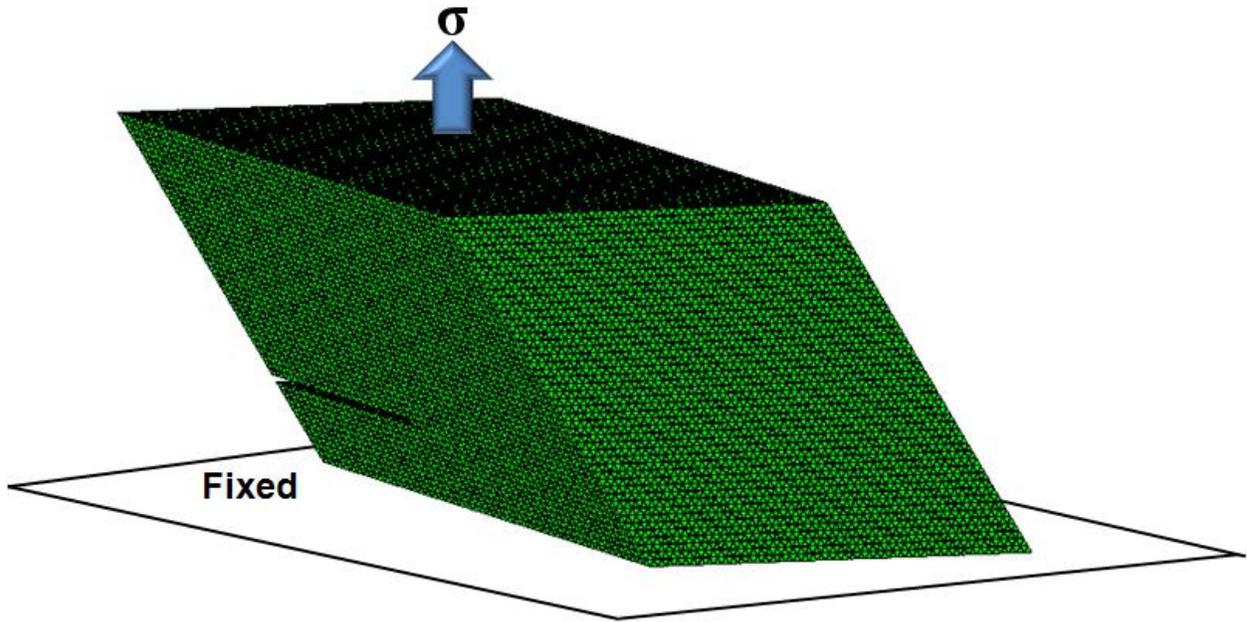


Figure 6-10. Computation model for the bulk material under Mode I loading

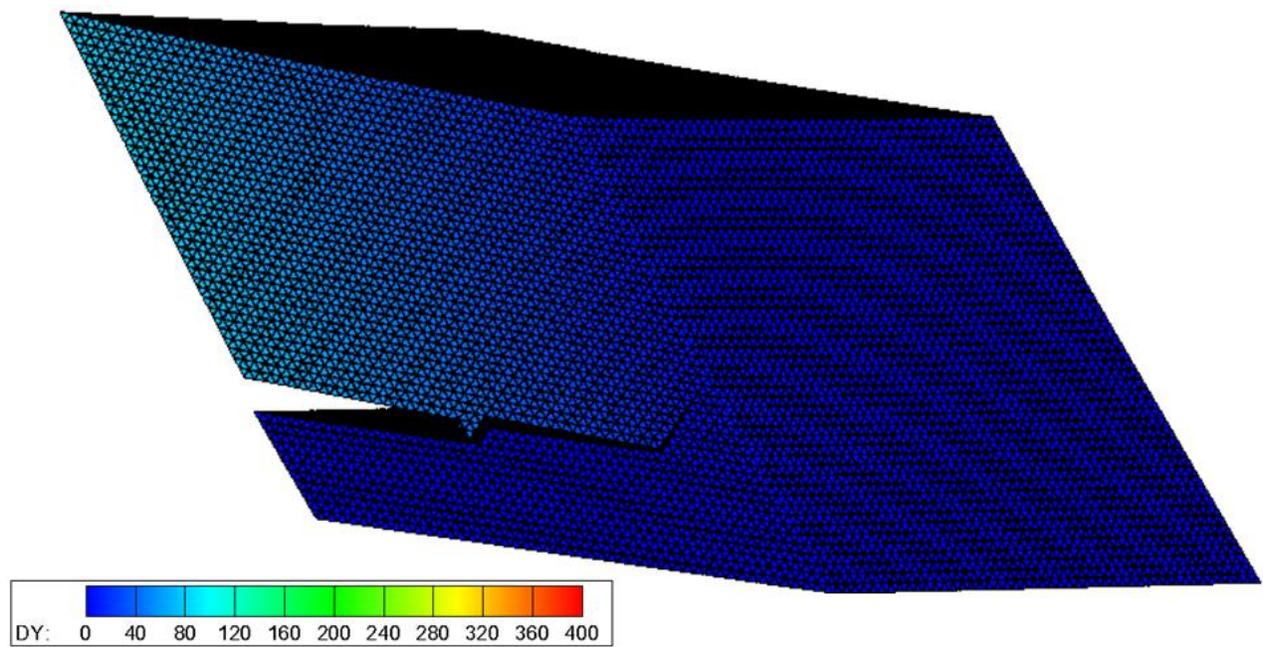


Figure 6-11. Crack surface evolution in bulk material

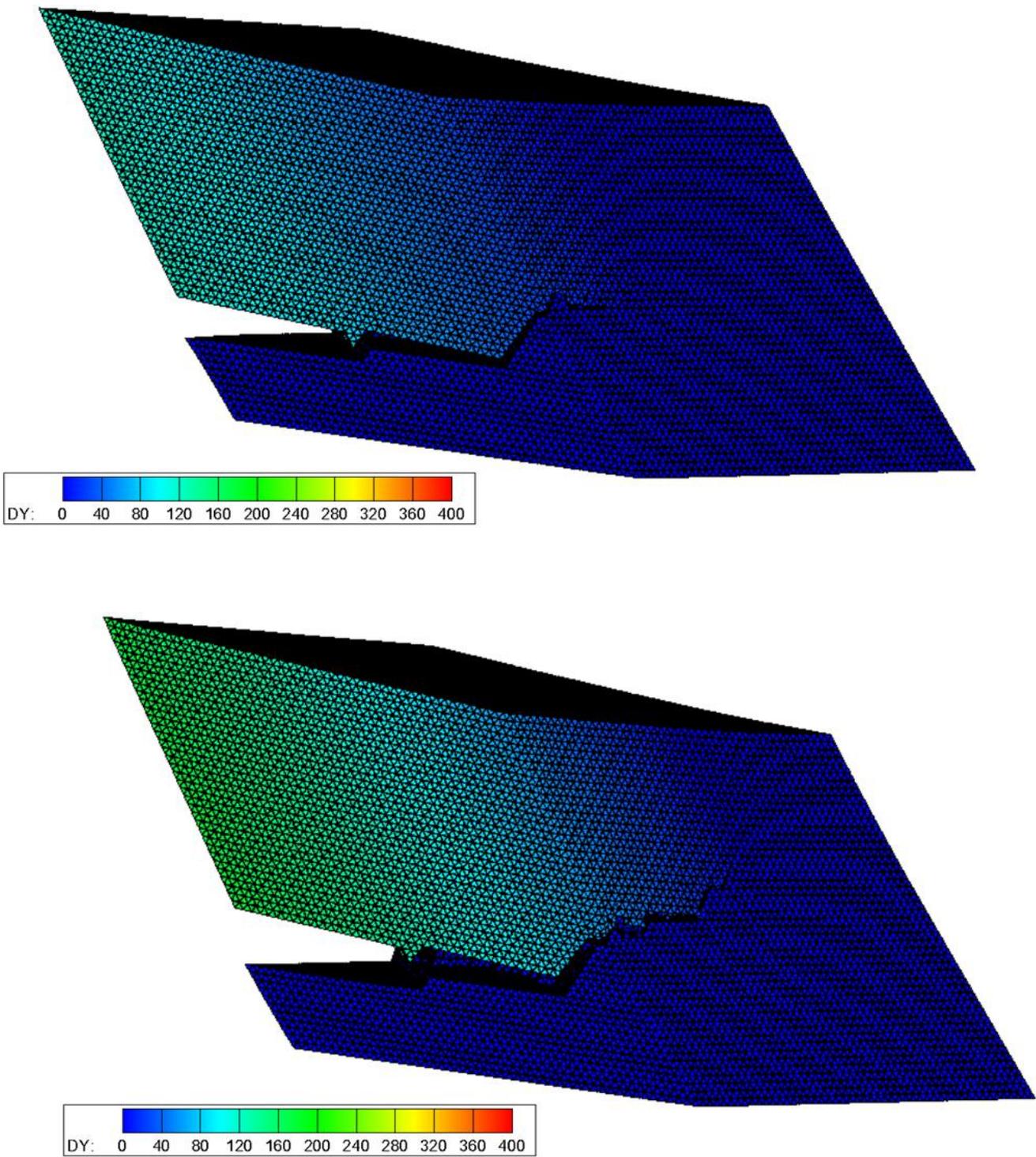


Figure 6-11. Continued

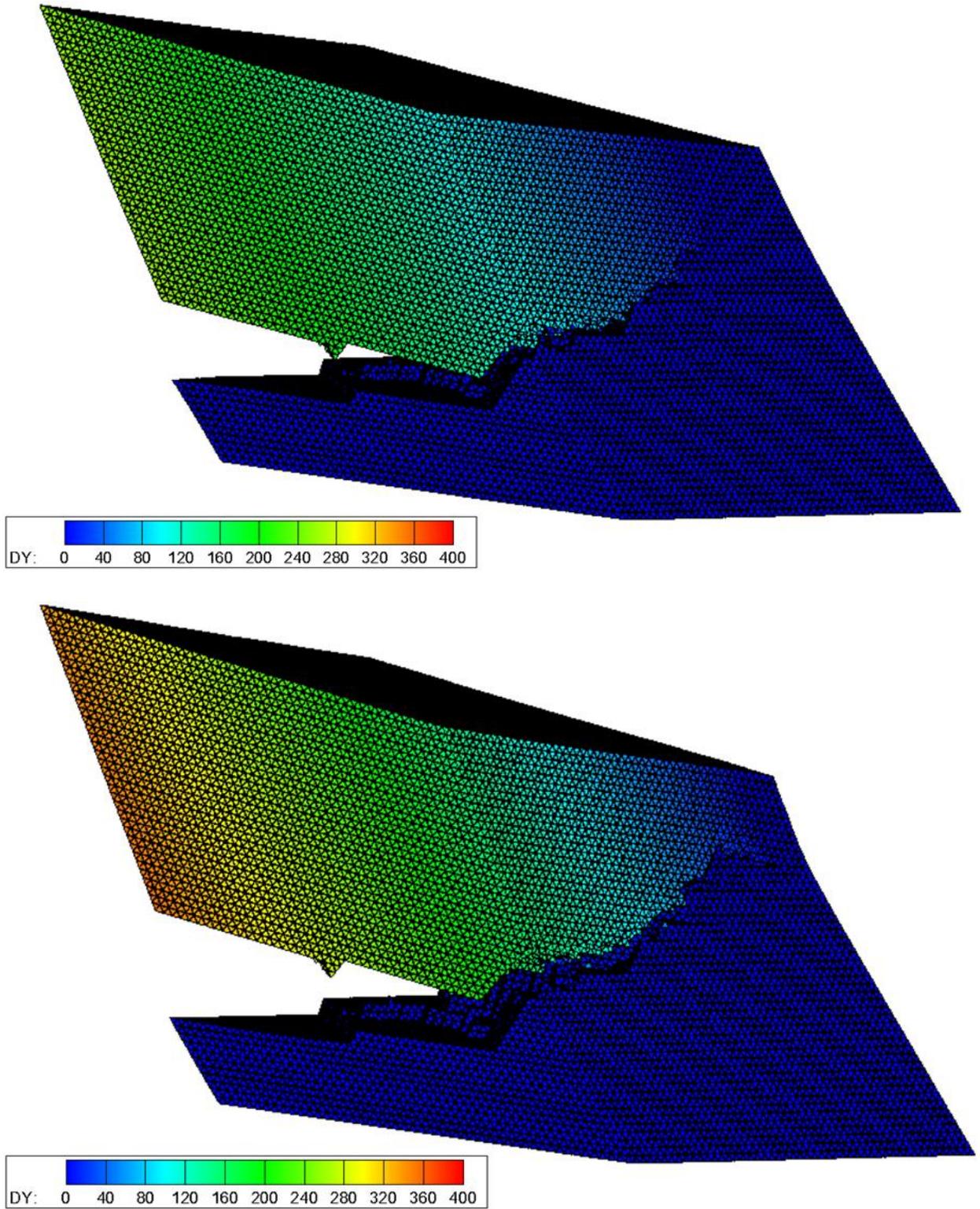


Figure 6-11. Continued

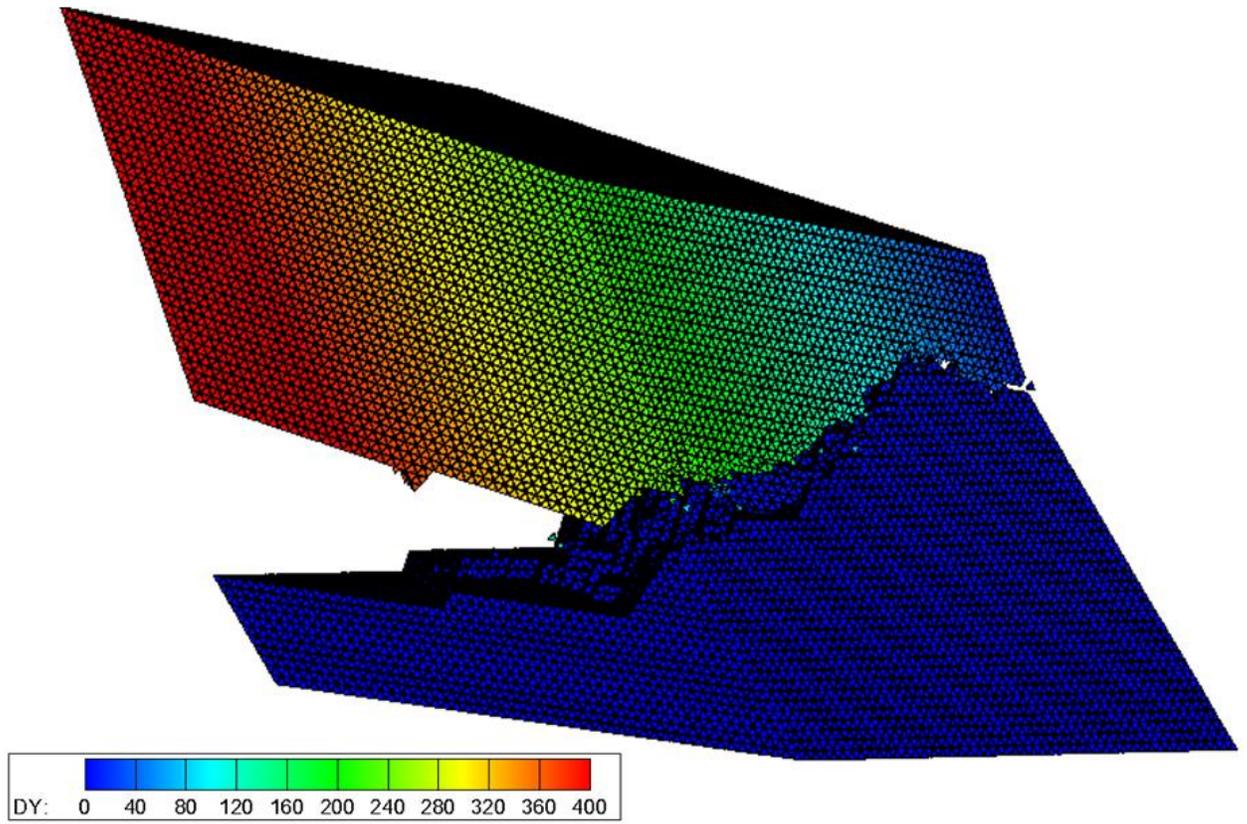


Figure 6-11. Continued

CHAPTER 7 SUMMARY AND FUTURE WORK

Summary and conclusion

The simulation of dynamic fracture is one of the most challenging topics in solid mechanics. On one hand, the existence of a crack or dislocation always brings discontinuities to the problem. On the other hand, fracture, in nature, is a multiscale dynamic process, which involves the breaking of material at different length scales ranging from the macroscale to nanoscale. Molecular dynamics is a numerical approach that successfully meets the two challenges listed above. However, this approach can only handle a model with a very small size because of the large time and memory cost. Multiscale modeling is a promising solution, but currently there is no successful model which deals with large-scale dynamic fracture.

In this work, a new coarse grained method, which incorporates almost all of the critical atomic information embedded into large elements, was presented. It is well-known that losing part of the atomic information is inevitable for coarse graining. Then the question is how much confidence we have when ignoring such atomic information. As we have mentioned in the first chapter, the fracture behavior of a single crystal material is closely related to its crystal structure. Thus not only the interatomic potential but also the key feature of the crystal structure should be embedded in the coarse grained model. Serving as the constitutive relation in AFT, the interatomic potential is naturally embedded in the coarse grained model, but how can the key features of the crystal structure in the coarse grained model be maintained? In this work, the utilization of carefully designed finite elements provides a prospective way to achieve this goal. For a 2D triangular lattice model, a 2D triangular finite element is designed to capture

the main feature of the lattice structure. For a 3D single crystal with a rhombohedral primitive cell, a hybrid finite element aimed to deal with the crack or dislocation propagation is designed.

To improve the efficiency and extend the size of the computation model, a parallel code is implemented using FORTRAN combined with MPI (Message Passing Interface). Domain decomposition strategy is used to divide the entire model into several sub-models. Each sub-model is assigned to one processor, and the communication between neighboring processors is performed every time step to share part of their results. The performance of the parallel code is tested on the supercomputer Trestle of SDSC (San Diego Supercomputer Center). The results show that the parallel code can handle large-scale computation with high efficiency.

To validate the coarse grained method, the comparison of the coarse grained results and the corresponding MD results are conducted. The results show that the new method can predict the crack tip stress, dynamic crack propagation and even crack branching, all of which are qualitatively and quantitatively similar with atomic-level molecular dynamics simulations in both 2D and 3D cases. It is worthwhile to mention that the differences of the results do not solely come from the coarse graining. The temperature treatment is also an important source of the differences. To approximate the reality more accurately, the complex temperature field around the crack tip needs to be considered.

As applications of the developed coarse grained method, crack branching is observed in both 2D and 3D cases. In the 2D case, it is found that the interaction of stress waves emitting from the propagating crack tip and the stress waves reflected

back from the boundaries have a strong influence on the crack speed, crack path, and crack branching. In 3D case, it is found that crack branching is closely related to the dislocations nucleated near the crack tip. These dislocations create two weak paths for the crack to follow.

In the 3D simulation of an impact test on a thin plate, the applicability of the method to large deformation, fragmentation, and the propagation of multiple unstable cracks is demonstrated. In the case of 3D unstable crack surface evolution, the ability of the parallel code in handling large-scale computation with a large number of processors is shown.

Future work

In this work, we only consider the special case where the temperature field in the whole model is given in advance of the computation. Sometimes this is a very crucial condition. For example, the temperature distribution at the crack tip is always very complex, with a high gradient due to the local heat generation during dynamic crack propagation. This means that the dynamic crack propagation phenomenon is always a thermal/mechanical coupling problem. So the conservation of energy equation should be solved in combination with the balance equation of linear momentum.

In AFT, the velocity of each atom is divided into two parts: the stream velocity and the instantaneous velocity. The difference is the velocity which contributes to the temperature of the system. In the presented coarse grained method, the stream velocity field within an element is approximated by linear shape functions. This approximation is perfect for homogeneous deformation. However, when the deformation gradient is large, such as around the crack tip, this approximation may be too simple if using a very large element. To overcome this element size limitation, higher order shape functions or more

sophisticated shape functions should be employed.

The presented methodology is a promising tool for the investigation of the mechanism of dynamic fracture of crystalline materials. The scope of the application should not be confined to some simple crystal such as FCC or BCC single crystals. In our future work, this methodology could be used in coarse graining the atomistic dynamics of many complex materials such as: semiconductor materials, biomineral materials, or even crystalline polymer materials.

LIST OF REFERENCES

- Abraham, F.F., 1996. Dynamics of Brittle Fracture with Variable Elasticity. *Physical Review Letters*. 77, 869.
- Abraham, F.F., Brodbeck, D., Rafeey, R.A., Rudge, W.E., 1994. Instability dynamics of fracture: A computer simulation investigation. *Physical Review Letters*. 73, 272-275.
- Abraham, F.F., Brodbeck, D., Rudge, W.E., Broughton, J.Q., Schneider, D., Land, B., Lifka, D., Gerner, J., Rosenkrantz, M., Skovira, J. Gao, H., 1998. Ab initio dynamics of rapid fracture. *Modelling and Simulation in Materials Science and Engineering*. 6, 639-670.
- Abraham, F.F., Gao, H., 2000. How Fast Can Cracks Propagate? *Physical Review Letters*. 84, 3113.
- Anderson, T.L., 1991. Fracture Mechanics, Fundamentals and Applications. CRC.
- Broughton, J.Q., Abraham, F.F., Bernstein, N., Kaxiras, E., 1999. Concurrent coupling of length scales: methodology and application. *Physical review B*. 60, 2391.
- Buehler, M.J., 2010. Strength in numbers. *Nature Nanotechnology*, 5, 172-174
- Buehler, M.J., Abraham, F.F., Gao, H., 2003. Hyperelasticity governs dynamic fracture at a critical length scale. *Nature*. 426, 141-146.
- Buehler, M.J., Gao, H., 2006. Dynamical fracture instabilities due to local hyperelasticity at crack tips. *Nature*. 439,307-310.
- Camacho, G.T., Ortiz, M., 1996. Computational modeling of impact damage in brittle materials. *International Journal of Solids and Structures*, 33, 2899-938.
- Chen, Y., 2006. Local stress and heat flux in atomistic systems involving three-body forces. *Journal of Chemical Physics*, 124, 054113.
- Chen, Y., 2009. Reformulation of microscopic balance equations for multiscale materials modeling. *Journal of Chemical Physics*, 130, 134706
- Chen, Y. Lee, J.D., 2005. Atomistic Formulation of A Multiscale Theory for Nano/Micro Physics. *Philosophical Magazine*, 85, 4095.
- Chen, Y., Lee, J.D., Lei, Y., Xiong, L., 2006. A Multiscale Field Theory: Nano/Micro Materials, G. C. Sih (Eds), Book chapter (invited contribution) on *Multiscale in Molecular and Continuum Mechanics*, Springer, 23.

- Coffman, V.R., Sethna, J.P., Heber, G., Liu, M., Ingraffea, A., Bailey, N.P., Barker, E.I., 2008. A comparison of finite element and atomistic modeling of fracture. *Modelling and Simulation in Materials Science and Engineering*. 16, 065008.
- Cox, B.N., Gao, H., Gross, D., Rittel, D., 2005. Modern topics and challenges in dynamic fracture. *Journal of Mechanics and Physics of Solids*, 53, 565-596.
- Deng, Q., Xiong, L., Chen, Y., 2010. Coarse-graining atomistic dynamics of brittle fracture by finite element method. *International Journal of Plasticity*. 26, 1402-1414.
- Deng, Q., Chen, Y., Lee, J., 2009. An Investigation of the Microscopic Mechanism of Fracture and Healing Processes in Cortical Bone. *International Journal of Damage Mechanics*. 18, 491-502.
- Eringen, A.C., 1972a. Linear theory of nonlocal elasticity and dispersion of plane waves. *International Journal of Engineering Science*. 10, 425-435.
- Eringen, A.C., 1972b. Nonlocal polar elastic continua. *International Journal of Engineering Science*. 10, 1-16.
- Eringen, A.C., Edelen, D.G.B., 1972. On nonlocal elasticity. *International Journal of Engineering Science*. 10, 233-248.
- Eringen, A.C., Suhubi, E.S., 1964. Nonlinear theory of simple micro-elastic solids—I. *International Journal of Engineering Science*. 2, 189-203.
- Espanol, P., 2004. Statistical mechanics of coarse-graining. *Lecture Notes in Physics*, 640, 69.
- Flores, K.M., Dauskardt, R.H., 1999. Enhanced Toughness Due to Stable Crack Tip Damage Zones in Bulk Metallic Glass. *Scripta Materialia*, 41, 937-943.
- Flynn, M.J., 1972. Some computer organizations and their effectiveness. *IEEE Transactions on Computers*. C-21(9).
- Freund, L.B., 1974. Crack propagation in an elastic solid subject to general loading. IV. Obliquely incident stress pulse. *Journal of Mechanics and Physics of Solids*, 22, 137-146.
- Freund, L.B., 1998. *Dynamic Fracture Mechanics* 2nd edn. Cambridge Univ. Press, Cambridge, UK.
- Fries, T.P., Belytschko, T., 2006. The intrinsic XFEM: a method for arbitrary discontinuities without additional unknowns. *International Journal for Numerical Methods in Engineering*, 68, 1358-1385.

- Gao, H., 1994. Surface roughening and branching instabilities in dynamic fracture. *Journal of Mechanics and Physics of Solids*, 41, 457-486.
- Ghoniem, N.M., Busso, E.P., Kioussis, N., Huang, H., 2003. Multiscale modeling of nanomechanics and micromechanics: an overview. *Philosophy Magazine*. 83, 3475-3528.
- Gibbs, J.W., 1902. Elementary principals in statistic mechanics. Yale University Press.
- Gumbsch, P., Zhou, S.J., Holian, B.L., 1997. Molecular dynamics investigation of dynamic crack stability. *Physical Review B*, 55, 3445-3455.
- Hardy, R.J., 1982. Formulas for determining local properties in molecular dynamics simulation: Shock waves. *J. Chem. Phys.*, 76, 622-628.
- Holian, B.L., Ravelo, R., 1996. Fracture simulations using large-scale molecular dynamics. *Physical Review B*. 5, 11275.
- Hone, T.D., Izvekov, S., Voth, G.A., 2005. Fast centroid molecular dynamics: A force-matching approach for the predetermination of the effective centroid forces. *J. Chem. Phys.*, 122, 54105.
- Hoover, W.G., 1986. Molecular Dynamics. Springer, Berlin.
- Hoover, W.G., 1991. Computational Statistical Mechanics. Elsevier, Amsterdam.
- Huynh, D.B.P., Belytschko, T., 2009. The extended finite element method for fracture in composite materials. *International Journal for Numerical Methods in Engineering*, 77, 214-239.
- Irving, J.H., Kirkwood, J.G., 1950. The statistical mechanics theory of transport process: IV. The equations of hydrodynamics. *J. Chem. Phys.*, 18, 817.
- Irving, J.H., Zwanzig, R.W., 1951. The statistical mechanics theory of transport process: V. The quantum dynamics. *J. Chem. Phys.*, 19, 1173.
- Izvekov, S., Parrinello M., Burnham C.J., Voth G.A., 2004. Effective force fields for condensed phase systems from *ab initio* molecular dynamics simulation: A new method for force-matching. *J. Chem. Phys.*, 120, 10896.
- Izvekov, S., Voth, G.A., 2005. Effective Force Field for Liquid Hydrogen Fluoride from Ab Initio Molecular Dynamics Simulation Using the Force-Matching Method. *J. Chem. Phys. B*, 109, 6573.
- Kirkwood, J.G., 1946. The statistical mechanics theory of transport process: I. General theory. *J. Chem. Phys.*, 14, 180.

- Kirkwood, J.G., 1947. The statistical mechanics theory of transport process: II. Transport in Gasses. *J. Chem. Phys.*, 15, 72.
- Kirkwood, J.G., Buff F.P., Green, M.S., 1949. The statistical mechanics theory of transport process: III. The coefficient of shear and bulk viscosity of liquids. *J. Chem. Phys.*, 17, 988.
- Lyubartsev, A.P., Laaksonen, A., 1995. Calculation of effective interaction potentials from radial distribution functions: A reverse Monte Carlo approach. *Phys. Rev. E*, 52, 3730.
- Lyubartsev, A.P., Laaksonen, A., 1996. Effective potentials for ion-DNA interactions. *J. Chem. Phys.*, 11, 207.
- Mattson, T.G., Sanders, B.A., and Massingill, B.L., 2005. Patterns for Parallel Programming. Pearson Education.
- Miller, R., Ortiz, M., Phillips, R., Shenoy V.B., Tadmor, E.B., 1998. Quasicontinuum models of fracture and plasticity. *Engineering Fracture Mechanics*. 61, 427-444.
- Needleman, A., 1987. A continuum model for void nucleation by inclusion debonding. *ASME Journal of Applied Mechanics*. 54, 525-531.
- Park K., Paulino G.H., Roesler J.R., 2009. A unified potential-based cohesive model of mixed-mode fracture. *Journal of Mechanics and Physics of Solids*, 57, 891-908.
- Pashley, D.W., 1960. A Study of the Deformation and Fracture of Single-Crystal Gold Films of High Strength inside an Electron Microscope. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 255,218-231.
- Pons, A.J., Karma, A., 2010. Helical crack-front instability in mixed-mode fracture. *Nature*, 464, 85-89.
- Remmers, J.J.C., Borst, R., Needleman, A., 2003. A cohesive segments method for the simulation of crack growth. *Computational Mechanics*. 31, 69-77.
- Ross, J., Kirkwood, J.G., 1954. The statistical mechanics theory of transport process: VI. Quantum theory of transport in gasses. *J. Chem. Phys.*, 22, 1094.
- Schultz, R.A., Jensen, M.C., Bradt, R.C., 1994. Single crystal cleavage of brittle materials. *International Journal of Fracture*, 65, 291-312.
- Silling, S.A, 2000. Reformulation of Elasticity Theory for Discontinuities and Long-Range Forces. *Journal of the Mechanics and Physics of Solids*, 48, 175–209.

- Silling, S.A, Ebrahim, A., 2005. A Meshfree Method Based on the Peridynamic Model of Solid Mechanics. *Computers and Structures*. 83, 1526–1535.
- Soper, A.K., 1996. Empirical Monte Carlo simulation of fluid structure. *Chem. Phys.*, 202, 295.
- Soper, A.K., 1997. The quest for the structure of water and aqueous solutions. *J. Phys.: Condens. Matter*, 9, 2717.
- Tadmor, E.B., Ortiz, M., Phillips, R., 1996. Quasicontinuum analysis of defects in solids. *Philosophy Magazine*. 73, 1529-1563.
- Xiao, S.P., Belytschko, T., 2004. A bridging domain method for coupling continua with molecular dynamics. *Computer Methods in Applied Mechanics and Engineering*. 193, 1645-1669.
- Xiong, L., Chen, Y., 2009a. Coarse-grained simulations of single-crystal silicon. *Modelling and Simulation in Materials Science and Engineering*. 17, 035002:1-035002:17.
- Xiong, L., Chen, Y., 2009b. Multiscale modeling and simulation of single-crystal MgO through an atomistic field theory. *International Journal of Solids and Structures*. 46, 1448-1455.
- Xiong, L., Chen, Y., Lee, J., 2008. Simulation of dislocation nucleation and motion in single crystal magnesium oxide by a field theory. *Computational Materials Science*. 42, 168.
- Xiong, L, Tucker, G., McDowell, D.L., Chen, Y., 2011. Coarse-grained atomistic simulation of dislocations. *Journal of the Mechanics and Physics of Solids*. 59, 160-177
- Yamakov, V., Saether, E., Phillips, D.R., Glaessgen, E.H., 2006. Molecular-dynamics simulation-based cohesive zone representation of intergranular fracture processes in aluminum. *Journal of the Mechanics and Physics of Solids*. 54, 1899-1928.
- Yoffe, E.H., 1951. The moving Griffith crack. *Philosophy Magazine*, 42, 739-750.
- Zhou, S.J., Lomdahl, P.S., Thomson, R., Holian, B.L., 1996. Dynamic Crack Processes via Molecular Dynamics. *Physical Review Letters*. 76, 2318.
- Zwanzig, R.W., Kirkwood, J.G., Stripp, K.F., Oppenheim, I., 1953. The statistical mechanics theory of transport process: VI. A calculation of the coefficients of shear and bulk viscosity liquids. *J. Chem. Phys.*, 21, 2050.

Zwanzig, R.W., Kirkwood, J.G., Oppenheim, I., Alder, B.L., 1954. The statistical mechanics theory of transport process: VI. The coefficient of thermo conductivity of monatomic liquids. *J. Chem. Phys.*, 22, 783.

BIOGRAPHICAL SKETCH

Qian Deng was born in Jingzhou, Hubei, P.R. China in 1978. After spending his early years in Jingzhou, Qian moved to Dalian, P.R. China in 1996 to attend the Dalian University of Technology (DUT). He received his B.S. in Engineering Mechanics from DUT in 2000. Then he moved to Wuhan, Hubei, P.R.China to pursue a M.S. in Huazhong University of Science and Technology (HUST). He received his M.S. in Engineering with a major in Solid Mechanics from HUST in 2006. At the same year, Qian joined Dr. Youping Chen's research group to pursue a Ph.D. in solid mechanics. He firstly spend one year to study with the mentorship of Dr. James Lee at the George Washington University (GWU). Then, following the plan, he joined the Atomistic and Multiscale Mechanics lab at the department of Mechanical and Aerospace Engineering, University of Florida. He received his Ph.D. from the University of Florida in the fall of 2011. His current research focuses on developing a multiscale parallel simulation tool for applications in computational material science.