

แบบจำลองเชิงตัวเลขเพื่อจำลองการเปลี่ยนแปลงของความบกพร่องแบบจุดในโครงผลึก



นางสาวขวัญชนก จันทร์สว่าง

ศูนย์วิทยทรัพยากร
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมนิเวศศาสตร์ ภาควิชานิเวศศาสตร์เทคโนโลยี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2552

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

NUMERICAL MODEL FOR SIMULATING POINT DEFECT TRANSIENTS



Miss Kwanchanok Chansawang

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Engineering Program in Nuclear Engineering

Department of Nuclear Technology

Faculty of Engineering

Chulalongkorn University

Academic year 2009

Copyright of Chulalongkorn University

ขวัญชนก จันทร์สว่าง : แบบจำลองเชิงตัวเลขเพื่อจำลองการเปลี่ยนแปลงของความบกพร่องแบบจุดในโครงสร้างผลึก. (NUMERICAL MODEL FOR SIMULATING POINT DEFECT TRANSIENTS) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : รองศาสตราจารย์ ดร. ศัญชัย นิลสุวรรณ โฉมิต, 80หน้า.

การศึกษาการเปลี่ยนแปลงของความบกพร่องแบบจุดใน โครงสร้างผลึกโดยใช้แบบจำลองเชิงตัวเลข เพื่อศึกษาการเปลี่ยนแปลงของความบกพร่องต่อเวลา การศึกษาพบว่า เมื่อเกิดความบกพร่องแบบจุดชนิดที่ว่างเดี่ยวขึ้นใน โครงสร้างผลึกแบบเฟสเซนเตอร์คิวบิกที่ภาวะสมดุล ค่าเฉลี่ยของปริมาณจำเพาะของอะตอมมีค่าเพิ่มขึ้นอย่างทันทีในช่วงแรก หลังจากนั้นมียาลดลงจนกระทั่งเข้าสู่สมดุลโดยมีค่าเฉลี่ยของปริมาณจำเพาะของอะตอมใกล้เคียงกับค่าตั้งต้น ในกรณีของความบกพร่องแบบจุดในบริเวณเดียวกัน โดยมีจำนวนความบกพร่องเพิ่มขึ้น มีจำนวนสูงสุดถึง 5 จุด ซึ่งทำให้เกิดเป็นที่ว่างขนาดใหญ่ขึ้น พบว่าการเปลี่ยนแปลงค่าเฉลี่ยปริมาณจำเพาะของอะตอมอย่างรวดเร็วในช่วงแรกมีค่าใกล้เคียงกัน แต่ระบบที่มีจำนวนความบกพร่องแบบจุดมากกว่าจะเคลื่อนเข้าสู่สมดุลใหม่เร็วกว่า และมีขนาดปริมาณจำเพาะของอะตอมใหญ่กว่ากรณีที่มีจำนวนความบกพร่องแบบจุดน้อยกว่า

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....นิเวศวิทยาเทคโนโลยี

สาขาวิชา.....วิศวกรรมนิเวศวิทยา

ปีการศึกษา.....2552

ลายมือชื่อนิติกร.....ขวัญชนก จันทร์สว่าง

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....

4871802521 : MAJOR NUCLEAR ENGINEERING

KEYWORDS : POINT DEFECT / CRYSTAL STRUCTURE / TRANSIENTS

KWUANCHANOK CHANSAWANG : NUMERICAL MODEL FOR SIMULATING
POINT DEFECT TRANSIENTS. THESIS ADVISOR : ASSOC.PROF.SUNCHAI
NILSUWANKOSIT,Ph.D., 80 pp.

A study on the dynamic of point defects in crystal structure was numerically conducted in order to observe their transients. It was found that when a single vacancy defect occurred in an initially perfect FCC-crystal structure, the average atomic volume was suddenly increased and then gradually decreased down to the value close to the initial value. In case of multiple vacancies at the same position with the number of vacancies up to 5, the results showed that the average atomic volumes at the beginning were very similar. However, the system with the higher number of point defects tended to approach the equilibrium faster and had the larger average atomic volume compared with the system with the lower number of point defects.

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย

Department : Nuclear Technology.....

Student's Signature K. CHANSAWANG

Field of Study Nuclear Engineering.....

Advisor's Signature

Academic Year : 2009.....

Acknowledgements

This dissertation would not have been possible without my advisor, Assoc.Prof.Dr. Sunchai Nilsuwankosit, for invaluable guidance and support.

I am grateful for all knowledge sources particularly the library service at CU, MU and KMUTT.

I would like to thank everyone in NT Department, for their encouragement and support.

Many sincere thanks to Ajarns from Mahidol University, both Kanchanaburi and Phayathai Campus, for their contributions.

Finally, I appreciate to my family, father and mother, my sisters, my brother, my best friends, tantan, ameen, and everyone who have been staying with me all times.



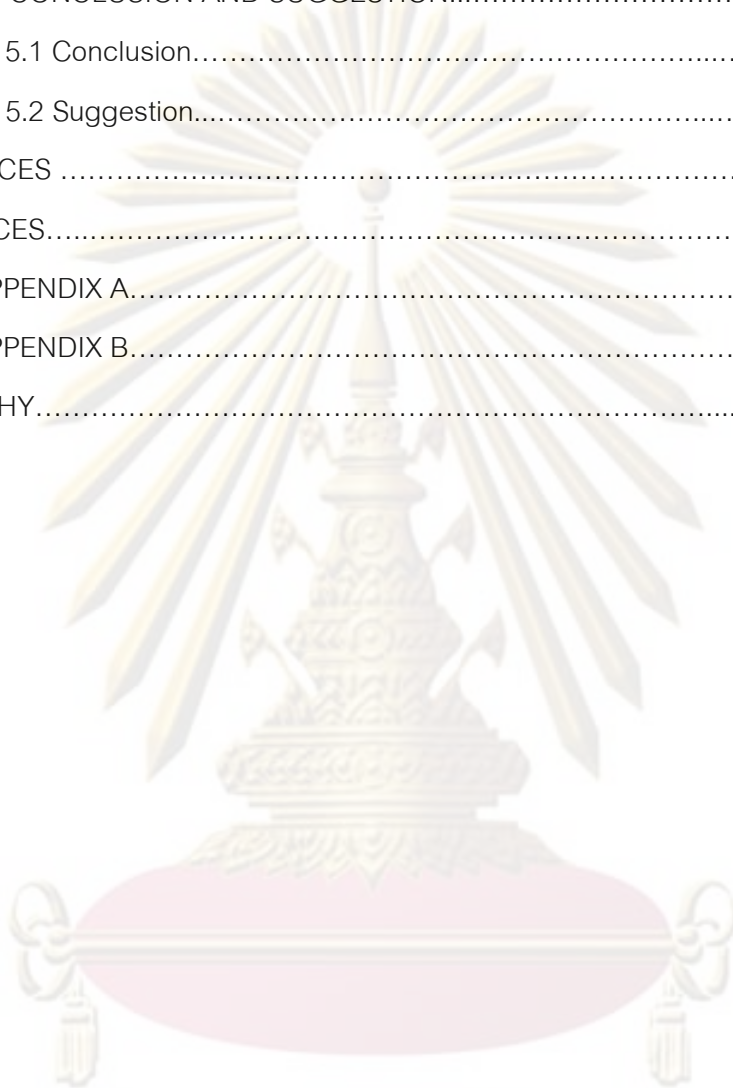
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Contents

	Page
ABSTRACT (THAI)	iv
ABSTRACT (ENGLISH).....	v
ACKNOWLEDGEMENT.....	vi
CONTENTS.....	vii
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER	
I INTRODUCTION.....	1
1.1 Background and Problems of Interest.....	1
1.2 Thesis Objective.....	2
1.3 Scope of Work.....	2
1.4 Expected Benefits.....	3
1.5 Literature Review.....	3
II THEORETICAL BACKGROUND.....	8
2.1 Defect of Crystal Structure.....	8
2.2 Vibrations of Molecules.....	12
2.3 Molecular Dynamic Simulation.....	15
III SPRING-MASS MODEL.....	20
3.1 Descriptions of the Model.....	20
3.2 Case Study on Simple Cubic Crystal.....	26
IV SIMULATION ON POINT DEFECT FOR FCC STRUCTURE.....	30
4.1 Simulation of Point Defect with ALINE.....	30
4.2 Box-Counting Model.....	32
4.3 Average Atomic Volume.....	33
4.4 Variation of Average Atomic Volume with Number of Divisions.....	34
4.5 Transients of Single Vacancy Defects.....	37
4.6 Transients of Multiple Vacancies Defects.....	38

Contents (Continued)

CHAPTER	Page
V CONCLUSION AND SUGGESTION.....	42
5.1 Conclusion.....	42
5.2 Suggestion.....	43
REFERENCES	44
APPENDICES.....	46
APPENDIX A.....	47
APPENDIX B.....	76
BIOGRAPHY.....	80



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

List of Tables

Table		Page
3.1	The description of the system; simple cubic 3x3x3.....	26



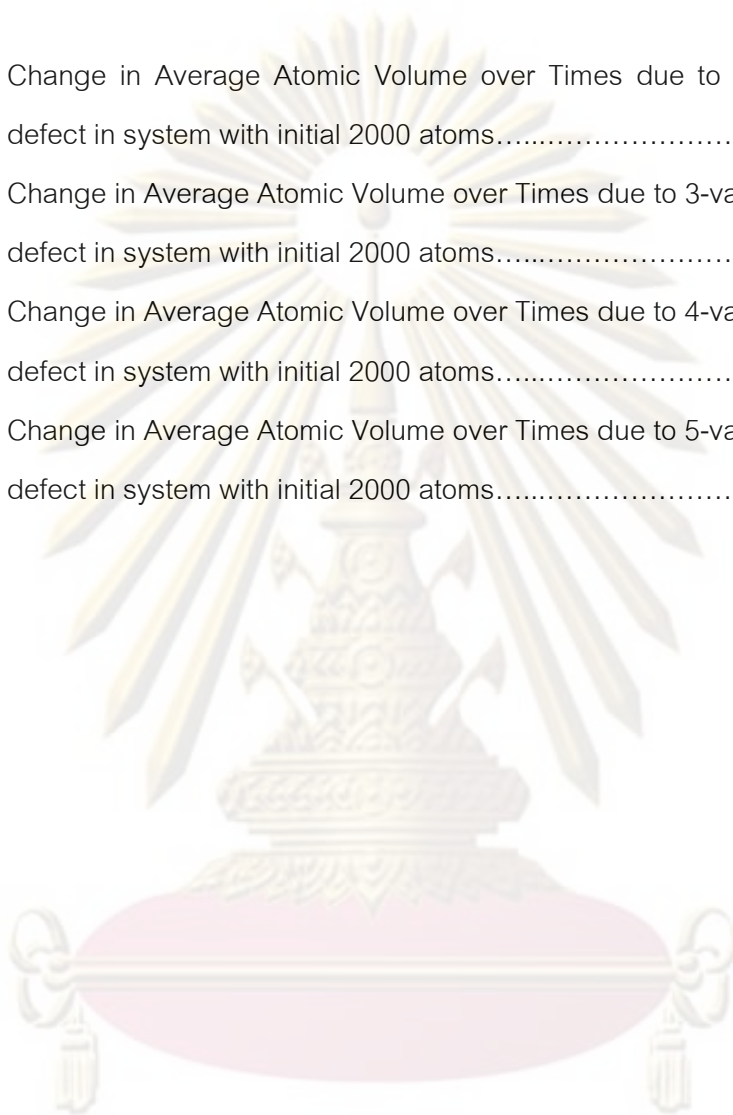
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

List of Figures

	Page
Fig. 2.1 (a) Vacancy and Self-interstitials, (b) and (c) Substitutiona.....	8
Fig. 2.2 Frenkel imperfection and Schottky imperfection.....	10
Fig. 2.3 (a) Screw dislocations, (b) Edge dislocations.....	10
Fig. 2.4 Intrinsic stacking fault.....	12
Fig. 2.5 Pair-wise potentials and the interatomic forces (a) LJ, (b) Morse.....	18
Fig. 3.1 Spring model in three dimensional.....	20
Fig. 3.2 Flow chart of the spring model.....	25
Fig. 3.3 Hamiltonian of the system with time.....	27
Fig. 3.4 Position of atom no.13 in X axis.....	28
Fig. 3.5 Position of atom no.13 in Y axis.....	29
Fig. 3.6 Position of atom no.13 in Z axis.....	29
Fig. 4.1 System of length L_x , width L_y and height L_z was divided in to subsystems, each with the length, the width and the height of B_x , B_y and B_z	33
Fig. 4.2 Variation of Average Atomic Volume with Number of subsystems (n_d) (a) system with initial 2000 atoms.....	35
Fig. 4.3 Variation of Average Atomic Volume with Number of subsystems (n_d) (b) system with initial 6750 atoms.....	36
Fig.4.4 Variation of Average Atomic Volume with Number of subsystems (n_d) (c) system with initial 16000 atoms.....	36
Fig.4.5 Variation of Average Atomic Volume with Number of subsystems (n_d) (d) system with initial 31250 atoms.....	37
Fig.4.6 Change in Average Atomic Volume over Times due to single vacancy defect of system with initial 2000 atoms, 6750 atoms, 16000 atoms and 31250 atoms.....	38
Fig.4.7 Change in Average Atomic Volume over Times due to single vacancy defect in system with initial 2000 atoms.....	39

List of Figures (Continued)

	Page
Fig.4.8 Change in Average Atomic Volume over Times due to 2-vacancies defect in system with initial 2000 atoms.....	40
Fig.4.9 Change in Average Atomic Volume over Times due to 3-vacancies defect in system with initial 2000 atoms.....	40
Fig.4.10 Change in Average Atomic Volume over Times due to 4-vacancies defect in system with initial 2000 atoms.....	41
Fig.4.11 Change in Average Atomic Volume over Times due to 5-vacancies defect in system with initial 2000 atoms.....	41



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER I

INTRODUCTION

1.1 Background and Problems of Interest

Solid materials are generally observed to have the crystal structure. The structure is made up of periodically repeating a structural building block, the unit cell. The pattern of ions or atoms within the crystal is described by the lattice. Atom or group of atoms attached at every lattice point is called the basis. The lattice and basis completely specify the crystal structure. The formation of the crystal is typically controlled by the valence electrons of the atoms that form the crystal. The strength, color and other physical properties are affected by the crystal structure.

Crystal is considered 'perfect crystal' when every atom stays in the correct position. This is the ideal condition which hardly exist. Every crystal contains lattice defects, or imperfections, which locally disrupt the ideal structure of the crystal. The general types of defects are point defects, line defects or dislocations and surface defects. Defects in the crystal structure can have a tremendous effect on a materials behavior, we can modify and improve many of the physical, electrical, magnetic, and optical properties of crystal materials by controlling imperfections in their lattice structure. So, in many fields of science and engineering are largely concerned with how such lattice defects are formed, how they interact with each other, and how they affect macroscopic properties. [1]

Both experiments and simulations are widely developed. Various experimental designs can show the appropriate results which are beneficial in many areas, but have the limitations in some conditions. On the other side, simulation can often come very close to experimental conditions, to the extent that computer results can sometimes be compared directly with experimental results. When this happens, simulation becomes an extremely powerful tool not only to understand and interpret the experiments at the microscopic level, but also to study regions which are not accessible experimentally, or which would imply very expensive experiments, such as under extremely high pressure.

The scale of simulation in computational crystal materials is vast, hierarchical models begin from nano-scale (characteristic length $\sim 10^{-9} - 10^{-7}$ m.), micro-scale (characteristic length $\sim 10^{-8} - 10^{-6}$ m.), meso-scale (characteristic length $\sim 10^{-7} - 10^{-4}$ m.), and macro-scale (characteristic length $\geq 10^{-3}$ m.). Most of simulation must be connected with appropriate scale but these make limitation in them.

The motivation of this dissertation is trying to understand the dynamics of lattice defects, specifically the transient of this system. The first objective aims to create a computer program to calculate the energy term, especially potential energy function. The reason for focusing on potential energy function is the behavior of lattice defects that atoms interact with each other. These interactions originate forces which action of these instantaneous forces. As the atoms move, their relative positions change and forces change as well. Forces are usually obtained as the gradient of a potential energy function, depending on the positions of the particles. The second objective aims to analyse the patterns of lattice defects and predict the possible final states of the transients.

1.2 Thesis Objective

1. To develop a computer program for simulating point defect transients.
2. To analyse the patterns in which the point defects may propagate over time.
3. To predict the possible final states of the transients and/or the time in which a transient may require to reach its final state.

1.3 Scope of Work

1. The study will be concentrated on the system with simple lattice formation in order to minimize the time and resource required for the simulation.
2. Various initial conditions for single and multiple defects will be attempted.

3. The obtained results are to be compared with the available data obtained from the experiments and/or from the theoretical prediction.

1.4 Expected Benefit

Obtain the computer program for simulating the transient of lattice defects. The program will be used as a tool in which further study and the analysis on the formation of the lattice and the propagation of their defects can be conducted.

1.5 Literature Review

1. In 1997 [2], Furio Ercolessi presented 'A molecular dynamics primer' that summarize the skeleton for a short hands-on course on molecular dynamics. These notes showed that in the past, interactions were obtained by two-body potentials with simple analytical form, such as Morse or Lennard-Jones (since 1957). After that, the most accurate potentials contain many-body terms and are determined numerically by reproducing as closely as possible forces predicted by first-principle methods (since 1985). Also explained about limitations of molecular dynamics when deal with the system at the atomistic level, everybody knows this level obey quantum laws rather than classical laws. A simple test of the validity of the classical approximation is based on the de Broglie thermal wavelength, defined as

$$\Lambda = \sqrt{\frac{2\pi\hbar^2}{Mk_B T}}, \quad (1.1)$$

where M is the atomic mass and T is the temperature. The classical approximation is justified if $\Lambda \ll a$, where a is the mean nearest neighbor separation. If one considers for instance liquids at the triple point, Λ/a is of the order of 0.1 for light elements such as Li and Ar, decreasing further for heavier elements. The classical approximation is poor for very light systems such as H_2 , He, Ne. Moreover, quantum effects become important in any system when T is sufficiently low. The drop in the specific heat of crystals below the Debye temperature, or the anomalous behavior of the thermal

expansion coefficient, are well known examples of measurable quantum effects in solids. Molecular dynamics results should be interpreted with caution in these regions.

These notes also emphasized that the realism of the simulation depends on the ability of the potential chosen to reproduce the behavior of the material under the conditions at which the simulation is run.

2. In 2000 [3], J. Merimaa, L.F. Perondi and K. Kaski introduced an interactive simulation program illustrating grain boundary and fracture phenomena in solids. The dynamical behavior of a two-dimensional Lennard-Jones model system under stress, with either a grain boundary or an initial crack, is simulated through a molecular dynamics algorithm. All parameters defining the system and the dynamical load are set through a graphical user interface. A run-time representation of the system is displayed on a graphics window, which has been endowed with magnification and other visualization aid tools. The program runs on a UNIX-X11 Window System platform. The graphical part relies on the MOTIF library. The program has been devised for illustrative purposes. It displays the main elements of an interactive simulation and may be regarded as giving an illustration of the concept of interactivity.

3. In 2002 [4], L.E. Shilkrot, W.A. Curtin and R.E. Miller introduced a method for reducing the degree of freedom in simulations of mechanical behavior of materials without sacrificing important physics. The method essentially combines the quasicontinuum (QC) method with continuum defect models such as the discrete dislocation (DD) method. A procedure for the 'passing' of dislocation defects from the atomistic to the continuum description in 2d problems is also presented. The overall 2d method with dislocation defects is validated by comparing the predictions of the coupled model to 'exact' fully atomistic models for several equilibrium dislocation geometries and a nanoindentation problem in aluminum, and excellent agreement is obtained.

4. In 2004 [5], Mike Finnis proposed an overview of the nature of interatomic forces in materials based on the ideas of density functional theory and the variational principle, without details of the mathematics involved. Also explain that ionic

materials and simple metals, arguments based on electrostatics and linear response have a firm foundation in quantum mechanics, and are key elements for making simplified models. In other materials, he showed that it is possible to simplify the many-body theory to such an extent that useful analytic models emerge.

5. In 2004 [6], W.K. Liu et al. briefly reviewed the essential tools used by nanoscale researchers. These simulation methods include the broad areas of quantum mechanics, molecular dynamics and multiple-scale approaches, based on coupling the atomistic and continuum models. They conclusively demonstrated that the atomistic simulation tools themselves are not sufficient for many of the interesting and fundamental problems that arise in computational mechanics, and that these deficiencies lead to the thrust of multiple-scale methods. They also summarized the strengths and limitations of currently available multiple-scale techniques, where the emphasis is made on the latest perspective approaches, such as the bridging scale method, multi-scale boundary conditions, and multi-scale fluidics.

In detail of this paper showed that nanoscale materials will be used in conjunction with other components that are larger, and have different response times, thus operating at different time and length scales. Single scale methods such as 'ab initio' quantum mechanical methods or molecular dynamics (MD) will have difficulty in analyzing such hybrid structures due to the limitations in terms of the time and length scales that each method is confined to. Because of the availability of accurate interatomic potentials for a range of materials, classical MD simulations have become prominent as a tool for elucidating complex physical phenomena.

The paper's outline is given by the following. Section 2 reviews the fundamentals of classical molecular dynamic simulations, such as the Lagrangian and Hamiltonian formulations, and the structure of interatomic potential functions. Section 3 informs the reader on the relevant quantum mechanical approaches and explains the energetic link between the quantum and classical systems. In Section 4, outline the intrinsic limitations of molecular dynamic simulations and emphasize the necessity in developing the coupled multi-scale methods. Section 5 reviews available multi-scale approaches: hierarchical and concurrent coupling of the atomistic and continuum

simulations (with the emphasis on the bridging scale method), multi-scale boundary conditions and multiple-scale fluidics. Section 6 concludes the paper by discussing future research needs in multiple-scale analysis.

This paper also introduced for such systems, the computer-aided engineering tools must be able to span length scales from nanometers to microns, and time scales from femtoseconds to micro-seconds. Therefore, these systems cannot be modeled by continuum methods alone, because they are too small, or by molecular methods alone, because they are too large. And in supporting the design and qualification of nanostructured materials, a range of simulation tools must be available to designers just as they are today available at the macroscopic scales in general purpose software.

6. In 2004 [7][8][9], Teik-Cheng Lim proposed the study of potential energy function in various papers, in particular the connection among classical interatomic potential functions, these papers showed the relationship among the classical functions of Lennard-Jones, Morse, Rydberg and Buckingham potential, followed by relationships between the classical pair potentials of Morse and Buckingham with the 2-body portion of other empirical potentials, such as Biswas-Hamann and Bauer-Maysenholder-Seeger. The results showed imposing equal potential energy, and their first two derivatives at equilibrium as well as the usage of calculus and Maclaurin series expansions.

7. In 2007 [10], M. Patriarca et al. proposed the study of crystal defects and the complex processes underlying their formation and time evolution has motivated the development of the program ALINE for interactive molecular dynamics experiments. They showed that ALINE couples a molecular dynamics code to a Graphical User Interface and runs on UNIX-X11 Window System platform with the MOTIF library, which is contained in many standard Linux releases. ALINE is written in C, thus giving the user the possibility to modify the source code, and at the same time, provides an effective and user-friendly framework for numerical experiments, in which the main parameters can be interactively varied and the system visualized in various ways. They illustrated the main features of the program through some examples of detection and dynamical

tracking of point-defects, linear defects, and planar defects, such as stacking faults in lattice-mismatched heterostructures.



ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER II

THEORETICAL BACKGROUND

2.1 Defects of Crystal Structure[11]

Every crystal contains lattice defects, or imperfections, which locally disrupt the ideal structure of the crystal. Defects come in several types and are often classified according to the number of spatial dimensions which is required to geometrically describe them. Their presence affects physical properties of the crystal and lead to various phenomena.

2.1.1 Point Defects

In a perfect lattice all atoms are located at specific lattice sites (ignoring thermal motion). There are two possible point defects, namely vacancies and interstitials. A vacancy is an unoccupied lattice point, while an interstitial is an atom located at a non-lattice site. For each fixed temperature there is an equilibrium concentration of vacancies in a crystal. In many common metals the fraction of vacant lattice sites just below the melting temperature is of the order of 10^{-4} to 10^{-3} . The equilibrium concentration of interstitials near the melting temperature is expected to be of the order of magnitude smaller than that of the equilibrium concentration of vacancies. In metals there may be impurities which can also be considered as point defects. There are two type: substitutional (an atom of a kind different from the host atoms occupying a lattice site) and interstitial (an atom of a kind different from the host atoms occupying a non-lattice site.) See Fig. 2.1.

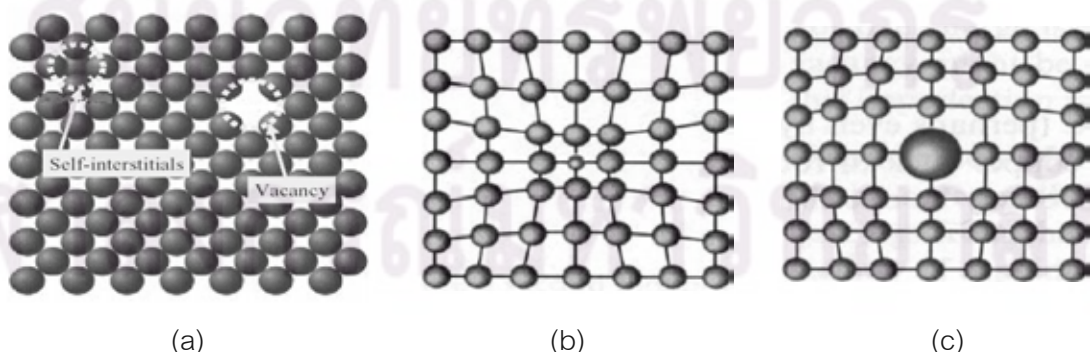


Fig. 2.1 (a) Vacancies and Self-interstitials, (b) and (c) Substitutional [11]

Point defects influence many physical properties of metals. For example, atoms surrounding a vacant lattice site are more loosely bound than those in a perfect lattice. The forces between the atoms thus are weakened (i.e. the vibrational frequencies are lowered) giving an increased vibrational entropy because the lattice is more compressed in the vicinity of an interstitial. Point defect mechanisms dominate in self-diffusion processes. Vacancies cause an increase in the volume of a crystal, and hence contribute to the thermal expansion. The enthalpy and heat capacity increase due to point defects.

The formation of point defects in solids was first predicted by Frenkel in 1926. At high temperatures the thermal vibrations of the atoms may cause an atom to leave its equilibrium lattice site and occupy an interstitial position in the lattice. In this mechanism a vacancy and an interstitial are created simultaneously, and are referred to as a Frenkel defect. In 1930, Wagner and Schottky demonstrated a way for the creation of vacancies: an atom leaves its lattice site and occupies an available position at the surface of the crystal or at internal imperfections in the crystal such as dislocations, grain boundaries, or inclusions, (Fig. 2.2). This mechanism dominates in metals with close-packed structure where the formation energy of vacancies generally is much lower than the formation energy of interstitials. Experimental measurements of vacancy concentrations often rely on the influence of vacancies on thermophysical properties, e.g. thermal expansion and electric resistivity. The problem is to separate the effect of vacancies from other effects, e.g. anharmonicity. Differential dilatometry measures the vacancy concentration directly. In such experiments the linear thermal expansion coefficient and the change in the lattice parameter with temperature are measured simultaneously. From the difference between those quantities the vacancy concentration can be deduced. The equilibrium concentration of interstitials can be determined using elastic constants measurements.

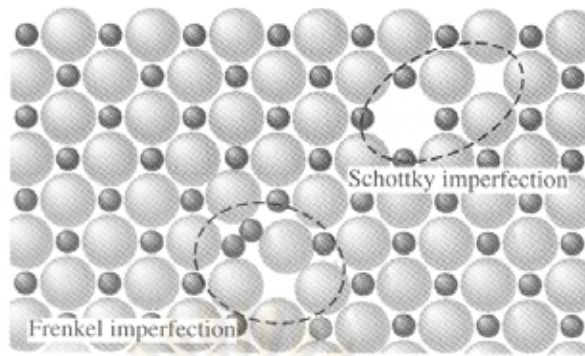


Fig. 2.2 Frenkel imperfection and Schottky imperfection [11]

2.1.2 Line Defects or Dislocations

Imagine that the bonds between the atoms in a simple cubic lattice are represented by flexible springs. If all bonds intersected by a plane which goes through non-atomic sites were instantaneously broken, the lattice would separate along this line if pulled in tension. A screw dislocation is introduced by displacing the two faces of such a cut relative to each other. If instead an extra atomic plane would be placed between the faces, an edge dislocation is created. In a real crystal, dislocations may have characteristics of both edge and screw type, (Fig. 2.3).

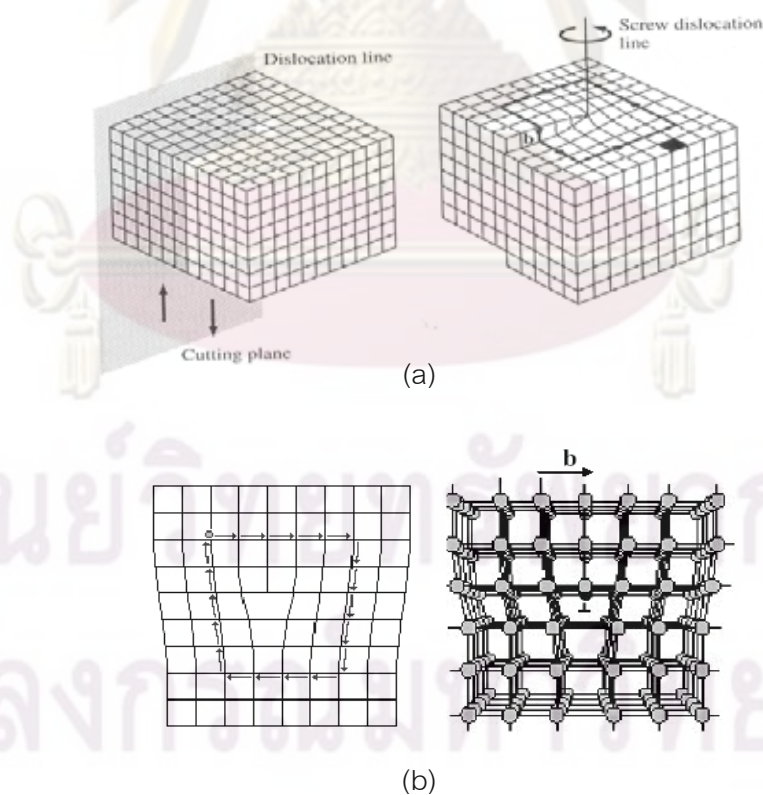


Fig. 2.3 (a) Screw dislocations, (b) Edge dislocations [11]

Dislocations have high formation energy and are not thermally generated at normal temperatures. However, dislocations are introduced through mechanical deformation even at intermediate temperature. The dislocation density is defined as the total length of dislocations per volume. In a carefully treated metallic crystal this value can be as low as 10^{-2} cm^{-2} . It increases rapidly with plastic deformation, and typical value for a heavily deformed metal is $5 \times 10^{11} \text{ cm}^{-2}$. The dislocation density in metallic crystals generally is higher than in non-metallic crystals. Thus, dislocations may become important at high temperatures, and are essential in understanding the deformation of materials. They account for the difference, by many orders of magnitude, between measured critical shear stresses of crystals and theoretical estimations.

The idea of dislocations dates from the end of the 1920s. In 1934 Orowan, Polanyi, and Taylor in simultaneous independent papers laid the foundations of the modern dislocation theory of slip, although the concept of dislocations had been introduced some years earlier. The first major review of the dislocation theory of crystals was written by Seitz and Read (1941), and later by Cottrell (1949). At the end of the 1960s, much of the modern theory of dislocations had been completed. The geometrical and elastic properties of dislocations, the theory of plastic deformation, and the role of dislocation theory in explaining crystal growth, were developed. There are many techniques for the detection of dislocations in crystals, e.g. X-ray diffraction, transmission electron microscopy, and field ion microscopy.

2.1.3 Planar Defects : Stacking faults

Perfect lattices can be described as a stacking of identical atom layers arranged in a regular sequence. There may be one or several ways to position one layer on top of another. In a face-centered cubic lattice there are two ways in which close packed planes (the (111) planes) can be placed on top of each other. The stacking of such planes can schematically be described as the sequence ABCABC... and C layer has been partially removed in the regular stacking sequence, which create and 'intrinsic' stacking fault. (Fig. 2.4)

Stacking faults play an important role in materials science because of their interaction with dislocation. They can be visualized by electron diffraction. Dislocations

often split into partial dislocations (in face-centered cubic metals), which are connected by a stacking fault. The stacking fault energy, γ is the energy required to introduce a stacking fault in a crystal. It is directly related to, e.g. the dissociation of a dislocation into two partials. Calculations of the stacking fault energy have been carried out using classical simulations and ab-initio methods. Using isotropic elasticity theory, the splitting distance, d_{split} between two partial dislocations can be calculated, from which $d_{split} \approx \gamma^{-1}$. The relation between γ and d_{split} may be used in experimental determinations of the stacking fault energy.

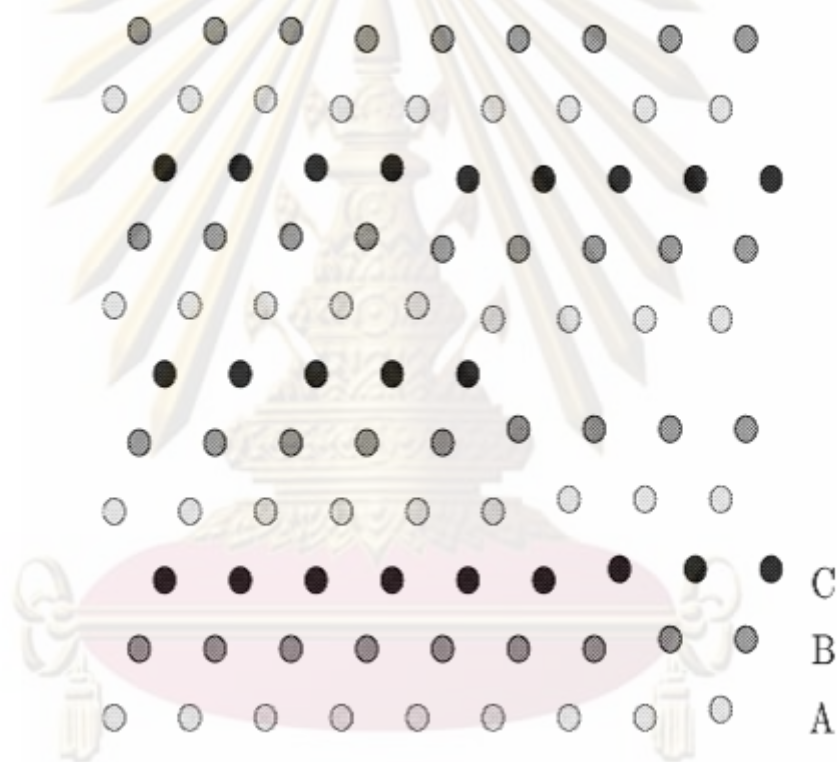


Fig. 2.4 Intrinsic stacking fault [11]

2.2 Vibrations of Molecules[12]

When two atoms are separated from each other by a few atomic diameters, they can exert attractive forces on each other. But if the atoms are so close to each other that their electron shells overlap, the forces between the atoms are repulsive.

Between these limits, there can be an equilibrium separation distance at which two atoms form a molecule. If these atoms are displaced slightly from equilibrium, they will oscillate.

As an example, consider one type of interaction between atoms called the *van der Waals* interaction. Let the center of one atom be at the origin, and let the center of the other atom be a distance r away; the equilibrium distance between centers is $r = R_0$. Experiment shows that the van der Waals interaction can be described by the potential energy function

$$U = U_0 \left[\left(\frac{R_0}{r} \right)^{12} - 2 \left(\frac{R_0}{r} \right)^6 \right], \quad (2.1)$$

where U_0 is a positive constant with units of joules. If two atoms are very far apart, $U = 0$; if they are separated by the equilibrium distance $r = R_0$, $U = -U_0$. The force on the second atom is the negative derivative of Eq. (2.1),

$$\begin{aligned} F &= -\frac{dU}{dr} = U_0 \left[\frac{12R_0^{12}}{r^{13}} - 2 \frac{6R_0^6}{r^7} \right] \\ &= 12 \frac{U_0}{R_0} \left[\left(\frac{R_0}{r} \right)^{13} - \left(\frac{R_0}{r} \right)^7 \right]. \end{aligned} \quad (2.2)$$

The force is positive for $r < R_0$ and negative for $r > R_0$, so this is a restoring force.

To study small-amplitude oscillations around the equilibrium separation $r = R_0$, we introduce the quantity x to represent the displacement from equilibrium:

$$x = r - R_0, \quad \text{so} \quad r = R_0 + x$$

In terms of x , the force F in Eq. (2.2) becomes

$$\begin{aligned} F &= 12 \frac{U_0}{R_0} \left[\left(\frac{R_0}{R_0 + x} \right)^{13} - \left(\frac{R_0}{R_0 + x} \right)^7 \right] \\ &= 12 \frac{U_0}{R_0} \left[\frac{1}{(1 + x/R_0)^{13}} - \frac{1}{(1 + x/R_0)^7} \right]. \end{aligned} \quad (2.3)$$

This looks nothing like Hooke's law, $F = -kx$, so we might be tempted to conclude that molecular oscillations cannot be simple harmonic motion. But let us restrict ourselves to small-amplitude oscillations so that the absolute value of the displacement x will be small in comparison to R_0 and the absolute value of the ratio x/R_0 will be much less than 1. We can then simplify Eq. (2.3) by using the binomial theorem:

$$(1 + u)^n = 1 + nu + \frac{n(n-1)}{2!}u^2 + \frac{n(n-1)(n-2)}{3!}u^3 + \dots \quad (2.4)$$

If $|u|$ is much less than 1, each successive term in Eq. (2.4) is much smaller than the one it follows, and we can safely approximate $(1 + u)^n$ by just the first two terms. In Eq. (2.3), u is replaced by x/R_0 and n equals -13 or -7, so

$$\frac{1}{(1 + x/R_0)^{13}} = (1 + x/R_0)^{-13} \approx 1 + (-13)\frac{x}{R_0},$$

$$\frac{1}{(1 + x/R_0)^7} = (1 + x/R_0)^{-7} \approx 1 + (-7)\frac{x}{R_0},$$

$$F \approx 12\frac{U_0}{R_0} \left[\left(1 + (-13)\frac{x}{R_0}\right) - \left(1 + (-7)\frac{x}{R_0}\right) \right]$$

$$F = -\left(\frac{72U_0}{R_0^2}\right)x. \quad (2.5)$$

This is just Hooke's law, with force constant $k = \frac{72U_0}{R_0^2}$, (Note that k has the correct units, J/m² or N/m.) So oscillations of molecules bounded by the van der Waals interaction can be simple harmonic motion, provided that the amplitude is small in comparison to R_0 so that the approximation $|x/R_0| \ll 1$ used in the derivation of Eq. (2.5) is valid.

We can also show that the potential energy U , Eq. (2.1), can be written as $U \approx \frac{1}{2}kx^2 + C$, where $C = -U_0$ and k is again equal to $\frac{72U_0}{R_0^2}$. Adding a constant to the potential energy has no effect on the physics, so the system of two

atoms is fundamentally no different from a mass attached to a horizontal spring for which $U = \frac{1}{2}kx^2$.

2.3 Molecular Dynamic Simulation [13],[14]

2.3.1 Potential Energy Function in Classical Molecular Dynamics Simulation

One of the objectives of this dissertation is to simulate the transient of lattice defects. In order to do so requires an understanding on the dynamics of lattice defects; the position, the velocity, the acceleration. The dynamics can be explained through the Hamiltonian equations of motion. The equations represent the energy of the physical system, that is, the sum of kinetic and potential energy. An alternative description, the terms of the generalized coordinates and momentum, is also utilized within the Hamiltonian formulation.

The complete differential of the Lagrangian function reads

$$dL = \sum_{i=1}^n \frac{\partial L}{\partial q_i} dq_i + \sum_{i=1}^n \frac{\partial L}{\partial \dot{q}_i} d\dot{q}_i, \quad (2.6)$$

where

L is the Lagrangian function.

q is the generalized coordinates.

We can rewrite this differential as

$$dL = \sum_i \dot{p}_i dq_i + \sum_i p_i d\dot{q}_i, \quad (2.7)$$

where p_i is the generalized momenta defined by

$$p_i = \frac{\partial L}{\partial \dot{q}_i}. \quad (2.8)$$

The right-hand side of Eq. (2.7) can be rearranged as

$$dL = \sum_i \dot{p}_i dq_i + d\left(\sum_i p_i d\dot{q}_i\right) - \sum_i \dot{q}_i dp_i, \quad (2.9)$$

$$d\left(\sum_i p_i \dot{q}_i - L\right) = \sum_i \dot{q}_i dp_i - \sum_i \dot{p}_i dq_i, \quad (2.10)$$

where the function

$$H(p, q, t) = \sum_i p_i \dot{q}_i - L, \quad (2.11)$$

is referred to as the Hamiltonian of the system. The value of the Hamiltonian function is an integral of motion for conservative system, and it is defined to be the total energy of the system in terms of the generalized coordinates and momenta. Thus, we have obtained

$$dH = \sum_i \dot{q}_i dp_i - \sum_i \dot{p}_i dq_i, \quad (2.12)$$

and therefore,

$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q_i}. \quad (2.13)$$

These are the Hamiltonian equations of motion in terms of new variable \mathbf{p} and \mathbf{q} .

For a conservative system of N interacting atoms in a Cartesian coordinate system, the Hamiltonian description acquires the following form:

$$H(r_1, r_2, \dots, r_N, p_1, p_2, \dots, p_N) = \sum_i \frac{p_i^2}{2m_i} + U(r_1, r_2, \dots, r_N), \quad (2.14)$$

$$\dot{r}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial r_i}, \quad (2.15)$$

where the momenta are related to the radius vectors as $\mathbf{p}_i = m_i \dot{\mathbf{r}}_i$.

If the Hamiltonian function and an initial state of the atoms in the system are known, one can compute the instantaneous positions and momentums of the atoms at all successive times that gives the dynamic evolution of the system.

Unlike the positions and momentums which can be computed quite easily from Hamiltonian function, the potential energy terms turns out to be difficult. The potential energy function is an extremely complicated object, when accurately representing the atomic interactions within the simulated system.

In a model for atomic interaction, the form of potential energy function for various systems is produced, the general structure of this function is presented by the following:

$$U(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) = \sum_i V_1(\mathbf{r}_i) + \sum_{i,j>i} V_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j>i,k>j} V_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots, \quad (2.16)$$

where \mathbf{r}_n is the radius vector of the n th particle and function V_m is called the m -body potential. The first term represents the energy due to an external force field, such as gravitational field or electrostatic field into which the system is immersed. The second term shows pair-wise interactions of the particles. The third term gives the three-body components. In practice, the external field term is usually ignored, while all the multi-body effects are incorporated into V_2 in order to reduce the computational expense of the simulations.

Explained in [6], there exist a variety of potential forms in classical molecular dynamics simulation. Among them are these two popular models to describe pair-wise atomic interactions:

Lennard-Jones (LJ) potential, [6],[13], reads as

$$V(\mathbf{r}_i, \mathbf{r}_j) = V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} + \left(\frac{\sigma}{r} \right)^6 \right],$$

$$r = |\mathbf{r}_{ij}| = |\mathbf{r}_i - \mathbf{r}_j|. \quad (2.17)$$

where \mathbf{r}_{ij} is the interatomic radius-vector, σ is the collision diameter, the distance at which $V(r) = 0$, and \mathcal{E} shows the bonding/dislocation energy. The first term of this potential represents atomic repulsion and the second term shows attraction between two atoms.

Morse potential reads as

$$V(r) = \mathcal{E} [e^{2\beta(\rho-r)} - 2e^{\beta(\rho-r)}], \quad (2.18)$$

where ρ and \mathcal{E} are equilibrium bond length and dislocation energy respectively; β is an inverse length scaling factor. The first term of this potential is repulsive whereas the second term is attractive and is interpreted as a representation of bonding. Morse potential has been adapted for modeling atomic interaction in various types of materials and interfaces.

The LJ and Morse potentials are most commonly used in molecular dynamics simulations, in chemistry, physics and engineering. They describe multi-body interaction:

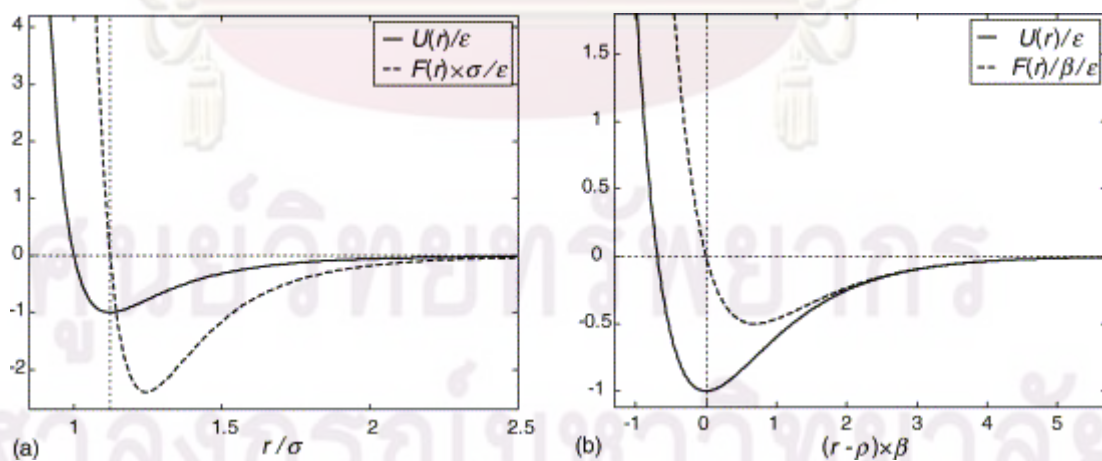


Fig. 2.5 Pair-wise potentials and the interatomic forces (a) LJ, (b) Morse.[6]

Tersoff potential [6],[15] for a class of covalent system reads as

$$V_{ij} = f_c(r_{ij})(Ae^{-\lambda_1 r_{ij}} - B_{ij}e^{-\lambda_2 r_{ij}}), \quad (2.19)$$

where f_c is the cutoff function. If the local bond-order is ignored, so that $B = 2A = \text{const.}$, and $\lambda_1 = 2\lambda_2$, then potential reduces to the Morse model.

Not only used to describe the pair-wise atomic interactions, but also the multi-body interaction, Brenner potential and Reactive Empirical Bond Order (REBO) potential [6],[16] is extended from the Tersoff model to account for different types of chemical bonds that occur in the diamond and graphite phases of the carbon, as well in hydrocarbon molecules.

There is another special form of the multi-body potential:

Embedded atom method (EAM) [6],[10] : for metallic system

$$U = \sum_i G_i \left(\sum_{j \neq i} \rho_j^a(r_{ij}) \right) + \sum_{i,j > i} V_{ij}(r_{ij}). \quad (2.20)$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER III
SPRING-MASS MODEL

3.1 Description of the Model

Using the concept of Hamiltonian, the general structure of potential energy function, and the revision of the vibration of molecules in CHAPTER II, we are ready to investigate the simplest model representing the system of atomic interaction namely the model of spring. (Fig.3.1)

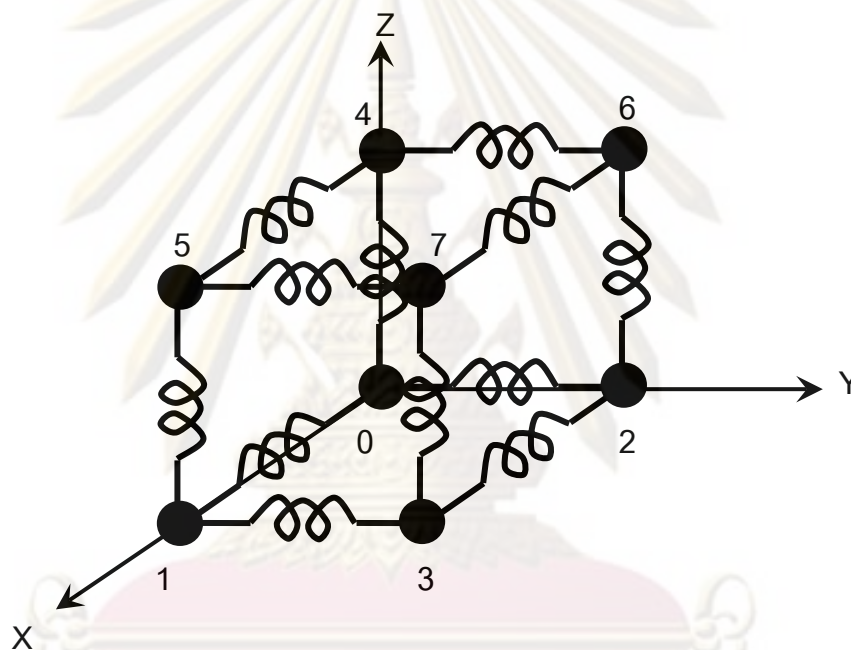


Fig. 3.1 Spring model in three dimensional

From Eq. (2.14), written as the sum of kinetic energy and potential energy, Hamiltonian equation is

$$H = T + V, \quad (3.1)$$

where T is the kinetic energy and V is the potential energy.

The potential energy function of the spring is harmonic potential,

$$V = \sum_{i=1}^m \frac{1}{2} K (\Delta l_i)^2, \quad (3.2)$$

where

$$\Delta l_i = |\vec{r}_\alpha - \vec{r}_\beta| - L, \quad (3.3)$$

and

$$|\vec{r}_\alpha - \vec{r}_\beta| = \sqrt{(x_\alpha - x_\beta)^2 + (y_\alpha - y_\beta)^2 + (z_\alpha - z_\beta)^2}, \quad (3.4)$$

and the kinetic energy of this system is

$$T = \sum_{j=1}^n \frac{1}{2} M (v_j)^2. \quad (3.5)$$

Hence,

$$H = \sum_{j=1}^n \frac{1}{2} M (v_j)^2 + \sum_{i=1}^m \frac{1}{2} K (\Delta l_i)^2 \quad (3.6)$$

where

j is the number of atom

i is the number of spring

M is the mass of atom

v_j is the velocity of atom j

K is the spring constant

Δl_i is the displacement length of spring i

L is the equilibrium length of spring

From Eq. (2.15), the force on atom n in X, Y and Z axis can be written

as

$$F_{nx} = -\frac{\partial H}{\partial x_n}, \quad (3.7)$$

$$F_{ny} = -\frac{\partial H}{\partial y_n}, \quad (3.8)$$

$$F_{nz} = -\frac{\partial H}{\partial z_n}. \quad (3.9)$$

Using formula method, we rewrite Eq. (3.7) as

$$\begin{aligned} F_{nx} &= -\frac{\partial}{\partial x_n} \left\{ \sum_{i=0}^m \frac{1}{2} K (\Delta l_i)^2 \right\}, \\ &= -\frac{\partial}{\partial x_n} \left\{ \sum_{i=0}^m \frac{1}{2} K (|\vec{r}_\alpha - \vec{r}_\beta| - L)^2 \right\}, \\ &= -\sum_{i=0}^m \left\{ \frac{\partial}{\partial x_n} \left(\frac{1}{2} K (|\vec{r}_\alpha - \vec{r}_\beta| - L)^2 \right) \right\}, \\ &= -\sum_{i=0}^m \left\{ \frac{1}{2} K \cdot 2 (|\vec{r}_\alpha - \vec{r}_\beta| - L) \cdot \frac{\partial |\vec{r}_\alpha - \vec{r}_\beta|}{\partial x_n} \right\}, \\ &= -\sum_{i=0}^m \left\{ K (|\vec{r}_\alpha - \vec{r}_\beta| - L) \right. \\ &\quad \cdot \frac{1}{2} [(x_\alpha - x_\beta)^2 + (y_\alpha - y_\beta)^2 + (z_\alpha - z_\beta)^2]^{-\frac{1}{2}} \\ &\quad \left. \cdot \frac{\partial}{\partial x_n} (x_\alpha - x_\beta)^2 \right\}, \\ &= -\sum_{i=0}^m \left\{ \frac{K (|\vec{r}_\alpha - \vec{r}_\beta| - L)}{2 \frac{|\vec{r}_\alpha - \vec{r}_\beta|}{|\vec{r}_\alpha - \vec{r}_\beta|}} \cdot 2(x_\alpha - x_\beta) \cdot \frac{\partial}{\partial x_n} (x_\alpha - x_\beta) \right\}. \end{aligned}$$

That is,

$$F_{nx} = - \sum_{i=0}^m \left\{ K \cdot \frac{(|\vec{r}_\alpha - \vec{r}_\beta| - L)}{|\vec{r}_\alpha - \vec{r}_\beta|} \cdot (x_\alpha - x_\beta) \cdot \frac{\partial}{\partial x_n} (x_\alpha - x_\beta) \right\}. \quad (3.10)$$

In a similar manner, Eq. (3.8) and Eq. (3.9) can be written as

$$F_{ny} = - \sum_{i=0}^m \left\{ K \cdot \frac{(|\vec{r}_\alpha - \vec{r}_\beta| - L)}{|\vec{r}_\alpha - \vec{r}_\beta|} \cdot (y_\alpha - y_\beta) \cdot \frac{\partial}{\partial y_n} (y_\alpha - y_\beta) \right\}. \quad (3.11)$$

$$F_{nz} = - \sum_{i=0}^m \left\{ K \cdot \frac{(|\vec{r}_\alpha - \vec{r}_\beta| - L)}{|\vec{r}_\alpha - \vec{r}_\beta|} \cdot (z_\alpha - z_\beta) \cdot \frac{\partial}{\partial z_n} (z_\alpha - z_\beta) \right\}. \quad (3.12)$$

Using numerical difference,

$$F_{nr} = - \frac{\partial H}{\partial r_n},$$

$$\frac{\partial H}{\partial r_n} \approx \frac{\Delta H}{\Delta r_n},$$

$$\frac{\Delta H}{\Delta r_n} = \frac{H(r_n + dr_n) - H(r_n)}{dr_n},$$

we have,

$$F_{nr} \approx - \frac{H(r_n + dr_n) - H(r_n)}{dr_n}. \quad (3.13)$$

In the numerical computation, we use Runge Kutta 4th order (RK4) as the numerical integrator [14],[17]. Let the problem be specified as follows,

$$\frac{d^2 f}{dt^2} = a(f), \quad (3.14)$$

$$g = \frac{df}{dt}. \quad (3.15)$$

Then the RK4 method for this problem is given by the following equations:

$$g = g_0 + \frac{1}{6}(dg_1 + 2dg_2 + 2dg_3 + dg_4), \quad (3.16)$$

$$f = f_0 + \frac{1}{6}(df_1 + 2df_2 + 2df_3 + df_4), \quad (3.17)$$

where

$$dg_1 = a(f_0) \cdot dt$$

$$df_1 = g_0 \cdot dt$$

$$dg_2 = a\left(f_0 + \frac{df_1}{2}\right) \cdot dt$$

$$df_2 = \left(g_0 + \frac{dg_1}{2}\right) \cdot dt$$

$$dg_3 = a\left(f_0 + \frac{df_2}{2}\right) \cdot dt$$

$$df_3 = \left(g_0 + \frac{dg_2}{2}\right) \cdot dt$$

$$dg_4 = a(f_0 + df_3) \cdot dt$$

$$df_4 = (g_0 + dg_3) \cdot dt$$

In the spring model, we start at the initial configuration of the system, coordinate of atoms in all axis. Next, a selected atom is relocated from equilibrium. Hamiltonian is then calculated. The force is computed in two ways; one is through Eq. (3.10-3.12) by formula method, and the other is through Eq. (3.13) by numerical method.

We divide the force by mass to obtain the acceleration. After that, we use the RK4 to solve the integral form to take the velocity, and finally the new position. Hamiltonian is calculated and these new positions are stored to the new configuration in the next time step. (Fig. 3.2)

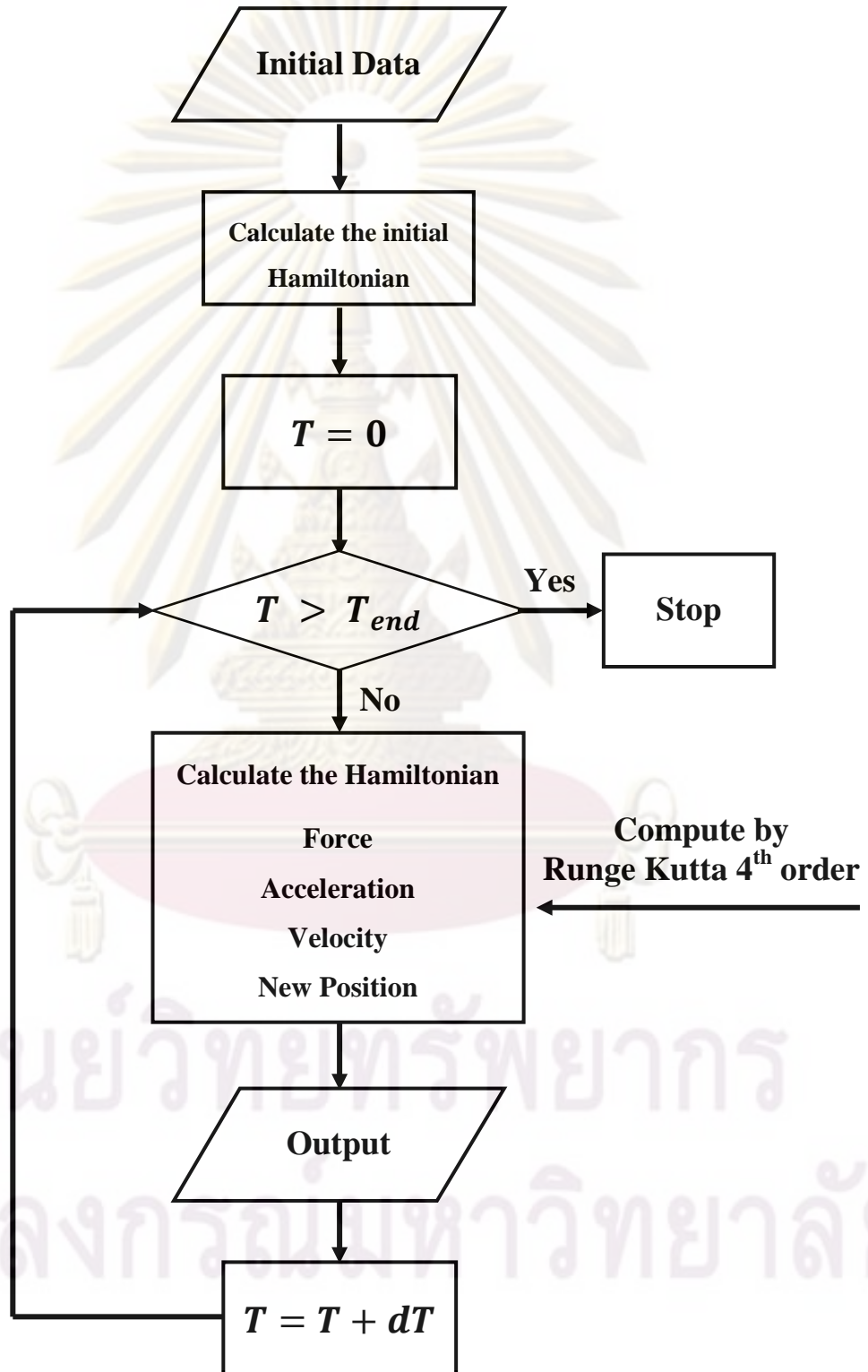


Fig. 3.2 Flow chart of the spring model

3.2 Case Study on Simple Cubic Crystal

The model observes the system of simple cubic 3x3x3. The description of the system is set as shown in the table:

Table 3.1 The description of the system; simple cubic 3x3x3.

Variable	Description of the variable
The number of atom	$j = 0,1,2,\dots,26.$
Mass of atom	$M = 1.0$ unit.
The number of spring	$i = 0,1,2,\dots,80.$
Spring constant	$K = 1.0$ unit.
Equilibrium length of spring	$L = 1.0$ unit.
Differential step	$dr = 1.0E-08.$
Time step	$dt = 0.01.$
Run time	2000.

Initial configuration of the system:

Atom no.13 is displaced from the coordinate (1.0,1.0,1.0) to (1.1,1.1,1.1).

Boundary condition of the system:

Periodic boundary condition.

The result from the simulation are shown as follows:

3.2.1 Relation of Hamiltonian with Time

The model calculated the motion of atoms in the system and the Hamiltonian at each time step. In calculation the forces acting on the atoms based on the obtained Hamiltonian, two methods were attempted. One was done through the analytically derived derivative of Hamiltonian. This was labeled as the H -formula method. The other was to obtain the derivative numerically. This was labeled as the H -

numerical method. The results obtained with these two methods were compared with the initial Hamiltonian (H_0) as shown in Fig. 3.3.

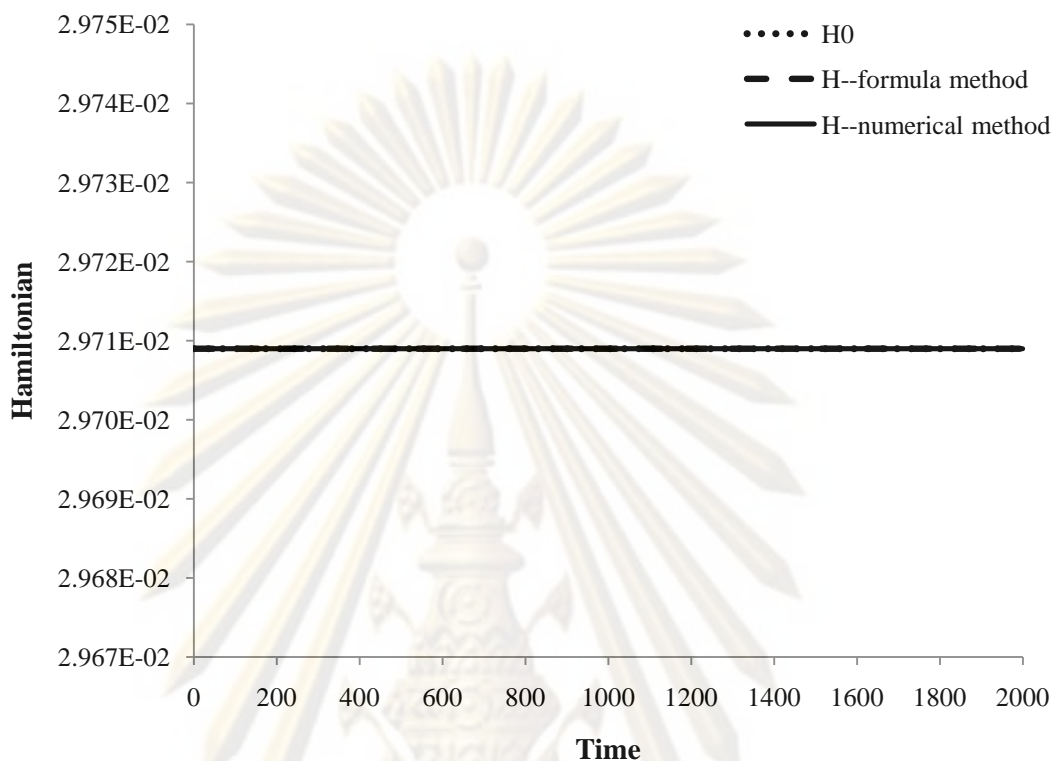


Fig. 3.3 Hamiltonian of the system with time

The Hamiltonian of the system by the formula method is found to be constant with time, meanwhile the value that calculated by the numerical method is slightly increases with time. They both nevertheless are very close to H_0 . The differences are at the 8th decimal point, which is very small. Hence, we could verify the conservation of Hamiltonian of the system and identify that the model can simulate this system in run time 2000.

3.2.2 Vibration of Atom No.13

The model can simulate the position of the displaced atom (no.13) versus time, also the Hamiltonian calculation. The position can be calculated by the formula method and the numerical method. The relation of position in X , Y and Z axis with time are shown respectively in the Fig. 3.4-3.6

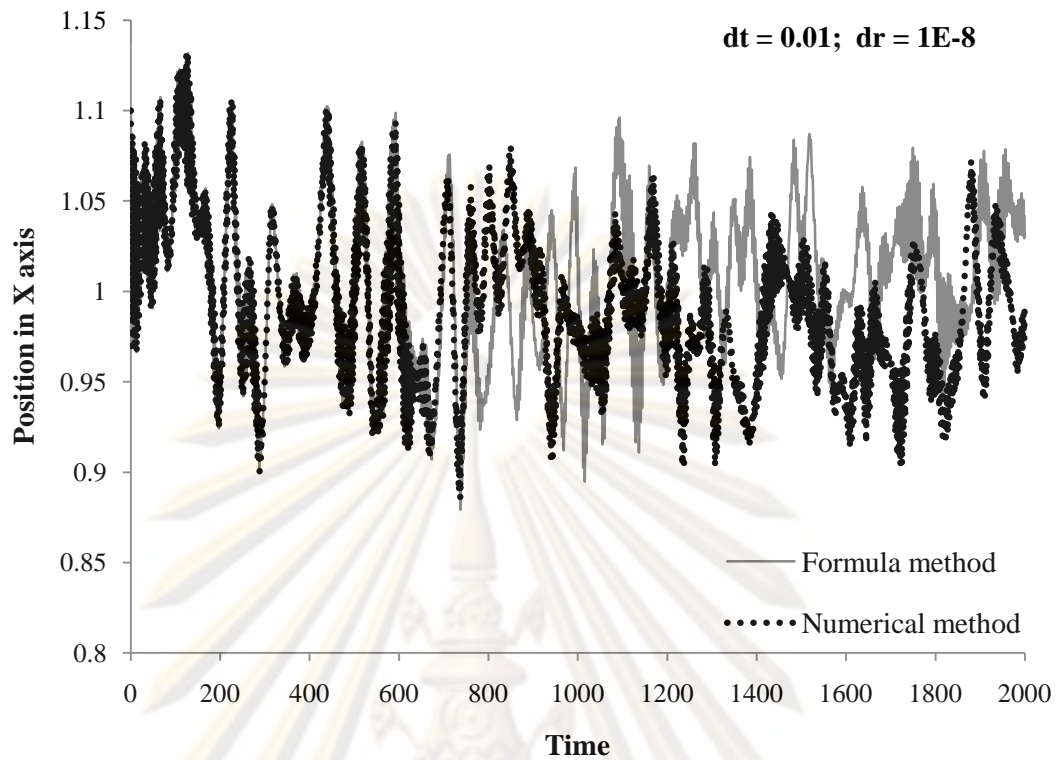
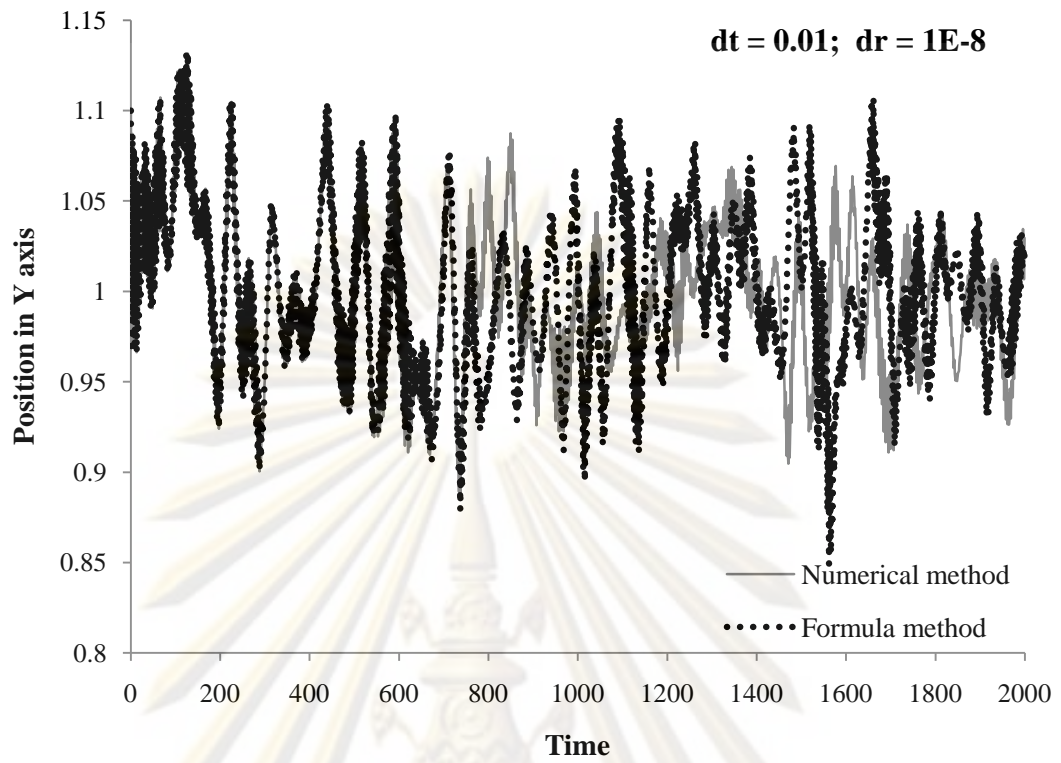
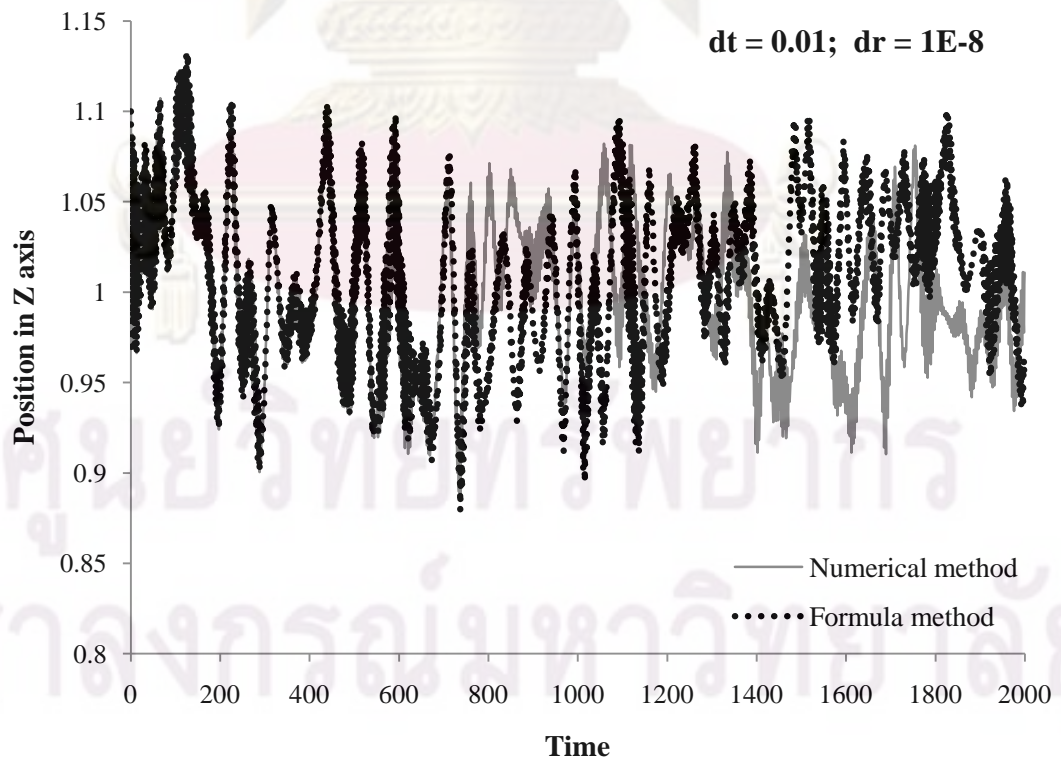


Fig. 3.4 Position of atom no.13 in X axis

The position of the atom no.13 varies around the equilibrium state that shows the vibration of the atom in the system. The value of position of atom no.13 in each axis versus time by both methods is the same value until the run time is about 700. This means that the numerical method in this model can show the exact value in limited run time. Therefore, making an adjustment of the differential step is needed for decreasing different value, but that will make more/longer time in simulation usage.

Fig. 3.5 Position of atom no.13 in Y axisFig. 3.6 Position of atom no.13 in Z axis

CHAPTER IV

SIMULATION ON POINT DEFECT FOR FCC STRUCTURE

4.1 Simulation of Point Defect with ALINE

In CHAPTER III, although it is clear that the model can accurately provide the results from calculation based on the Hamiltonian in order to simulate of the atom vibration in a simple cubic crystal, there still are some limitations in crystal system and particularly in terms of time usage of simulation.

For this study, to reduce the time needed to develop and verify the code, the numerical program ALINE which was written with the same concept was implemented. The ALINE program is strongly effective in simulating point defects in FCC crystal system. Thus, the program has been implied in this research to study the dynamics of point defects as well as the transients of the system.

ALINE (Atomic Laboratory for Interactive Numerical Experiments) is a three-dimensional interactive molecular dynamics program designed to simulate dynamics of the defects in the crystalline structure. This program couples a molecular dynamics code to a Graphical User Interface and runs on a UNIX-X11 Window System platform with the MOTIF library, which contained in many standard Linux releases. ALINE provides an effective and user-friendly framework for numerical experiments, in which the main parameters can be interactively varied and the system visualized in various ways. So, this program is suitable for the problem of interest in this study.

In model system, the core of the program is based on a Molecular Dynamics algorithm for a system of N particles interacting with each other either through a pair-wise Lennard-Jones (LJ) potential or a many-body EAM potential,

Lennard-Jones pair-wise potential

In the case of LJ potential, the total potential energy at time t is expressed as a sum over all the pairs $i - j$, with $i, j = 1, \dots, N$,

$$E_p = \frac{1}{2} \sum_{i \neq j} W_{ij}(r_{ij}), \quad (4.1)$$

where $r_{ij} = |r_i(t) - r_j(t)|$ is the distance between particles i and j located at $r_i(t)$ and $r_j(t)$. To improve the performance of simulations without affecting considerably the dynamics of the system, an interaction cut-off r_c is introduced and the cell method used for computing the interatomic forces. The functional form of the LJ potential used is modified as follows,

$$W_{ij}(r) = \begin{cases} V_{ij}(r) - (r - r_c)V'_{ij}(r_c) - V_{ij}(r_c), & r < r_c, \\ 0, & r > r_c. \end{cases} \quad (4.2)$$

Where $V'_{ij}(r_c) = [\partial V_{ij}(r)/\partial r]_{r=r_c}$, while $V_{ij}(r)$ represents the unperturbed LJ potential, the same as Eq. (2.17),

$$V_{ij}(r) = 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r} \right)^{12} - \left(\frac{\sigma_{ij}}{r} \right)^6 \right], \quad (4.3)$$

where ε_{ij} and σ_{ij} represent the energy and length scales, respectively, of the interaction potential between the i^{th} and the j^{th} particle. The corresponding potential minimum is $V_{ij} = -\varepsilon_{ij}$ at $r = 2^{1/6}\sigma_{ij}$. The parameter ε_{ij} and σ_{ij} may be different depending on the particle types and are assumed to be related to those of particles i and j as $\varepsilon_{ij} = \sqrt{\varepsilon_{ii}\sigma_{jj}}$ and $\sigma_{ij} = \sigma_{ii} + \sigma_{jj}$. The last two terms on the right-hand side of Eq. (4.2) represent correction terms which make the potential and its first derivative continuous at the cut-off $r = r_c$. It is to be noticed that Eq. (4.2) represents a potential with functional form different from that of the original LJ potential, the equilibrium distance and the potential minimum may differ appreciably if r_c is chosen too small. Thus, the potential parameters ε_{ij} and σ_{ij} are properly rescaled at the beginning of the run in order to preserve the original values of the potential minimum and equilibrium distance for the unperturbed LJ potential. All the LJ parameters ε , σ , and r_c are now the basic input parameters, which can be fixed by the user and are by default assigned the rescaled values $\varepsilon = 1$, $\sigma = 1$, and $r_c/\sigma = 2.5$, while the equilibrium lattice constant a is computed by minimizing the interaction energy with the

neighbors. This does not create big problems because it is expected that if σ is of the order of unity, also the lattice constant a will be of the same order of magnitude. For time scale, in ALINE used time step, $dt = 0.01$.

In general, the material of the study was presumed to have FCC type structure. The kind of atoms or ions that formed the basis of the structure was not specified. While this could have large effect on the length and time scale of the transient, the pattern in which the transient proceeded which was of major interest should be less affected.

For the simulations, the lattice constant was presumed to have the value of unity and the periodic condition was used for the boundary condition. Four cases of simulations were attempted on single vacancy. The volumes of these systems were initially containing (a) 2000 atoms, (b) 6750 atoms, (c) 16000 atoms and (d) 31250 atoms. In each case, only one atom at the center of the system was assumed to be missing when the simulation was initiated. Then the multiple vacancies (2-, 3-, 4-, and 5-atoms) were observed in system of 2000 atoms. With the periodic boundary, the different number of atoms contained in the system then in turn identified the defect density in the actual material.

4.2 Box-Counting Model

In obtaining the average atomic volume, the volume averagely occupied by one atom, it was necessary to count the number of atom contained in a given small volume, subsystem, and then to divide it by the volume. It should be noted that if the subsystem was too small, the number of atoms contained could be varied very widely. On the other hand, if the subsystem was too large, then the significant of difference in number of atoms then might be lost. It was therefore necessary to choose the subsystem that suited the system. The idea of dividing the system in subsystems or blocks of linear dimension had been applied to a number of problems related to phase transitions since 1964 [18],[19],[20].

Since ALINE produced the output which described the positions of atoms at the different time step, by specifying the subsystem as described in Fig. 4.1, it was then possible to count the number of atoms contained in each subsystem.

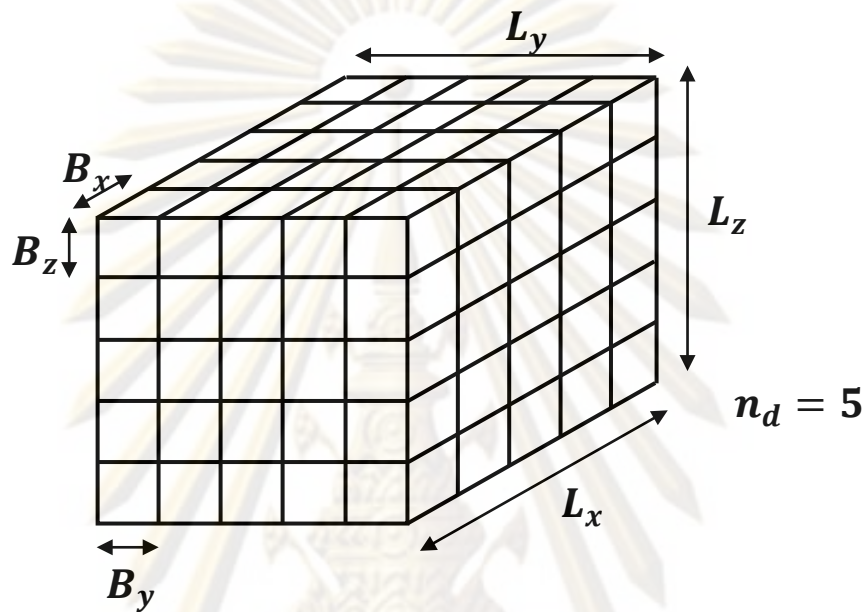


Fig. 4.1 System of length L_x , width L_y and height L_z was divided in to subsystems, each with the length, the width and the height of B_x , B_y and B_z

$$\left(B_x = \frac{L_x}{n_d}, B_y = \frac{L_y}{n_d}, B_z = \frac{L_z}{n_d} \right).$$

4.3 Average Atomic Volume

From the distribution of atoms in the system and box-counting model, the information regarding the number of subsystems containing the same number of atoms was obtained and the frequency distribution was created. With i being the specific case where the number of atom n_i in the subsystem was counted and f_i being the number of such subsystems, the average atomic volume V_m was calculated as

$$V_m = \frac{\sum_i f_i \frac{B_x B_y B_z}{n_i}}{\sum_i f_i} \quad (4.4)$$

It should be noted that $\sum_i f_i$ was the total number of subsystems and $\sum_i f_i n_i$ was the total number of atoms contained in the whole system.

4.4 Variation of Average Atomic Volume with Number of Divisions

From the results obtained from ALINE, transformed to 'dat' files, containing the value of x, y, and z coordinates, the box-counting model was used and the average atomic volume was calculated. As it was necessary to identify the suitable size of subsystem which would have allowed for the calculation of the average atomic volume that was both stable (was not widely varied by the number of subsystems) but still sensitive to the change in number of atoms that were moving in and out of the subsystems. To consider for the suitable size, the counting obtained with different number of subsystems (n_d^3 , n_d was the number of divisions in each axis) were obtained and the related average atomic volume were calculated as shown in Fig. 4.2-4.5

From Fig. 4.2-4.5, the very large value of n_d in every case resulted in the very small average atomic volume. This was because the actual volumes of the subsystems in these cases were so small such that they would either contain only one atom or not at all. This condition highly distorts and under estimated the average atomic volume. On the other hand, with the very low number of n_d , the subsystems became large such that the change in number of atoms it contained could be insignificant. It was of interest that a peak in the average atomic volume curved was observed in every case. It was tempting to assumed that the value of n_d associated with such peak gave the suitable of subsystems to be used. However, it must be reminded that with the FCC type of crystalline structure, the cube shape volume of the system meant that the total initial atoms of (a) 2000, (b) 6750, (c) 16000 and (d) 31250 were respectively associated with the arrangement of 2 structure sets; (a) 10x10x10, (b) 15x15x15, (c) 20x20x20 and

(d) 25x25x25. As a result, in each case, the value of n_d that was a common denominator to such arrangement should be avoided. This was because it gave the subsystems that coincided with the crystalline arrangements and, therefore, might lead to the bias results. This assessment was confirmed as shown by graphs for the systems with no defects in Fig. 4.2-4.5

For the systems with one single vacancy defect, the different values were chosen for n_d in each case. For system with the total atoms of 2000, n_d was 3. The system with the total atoms of 6750, n_d was 7. For the system with the total atoms of 16000, n_d was 11, and the value of 15 was chosen for n_d for the system with the total atoms of 31250

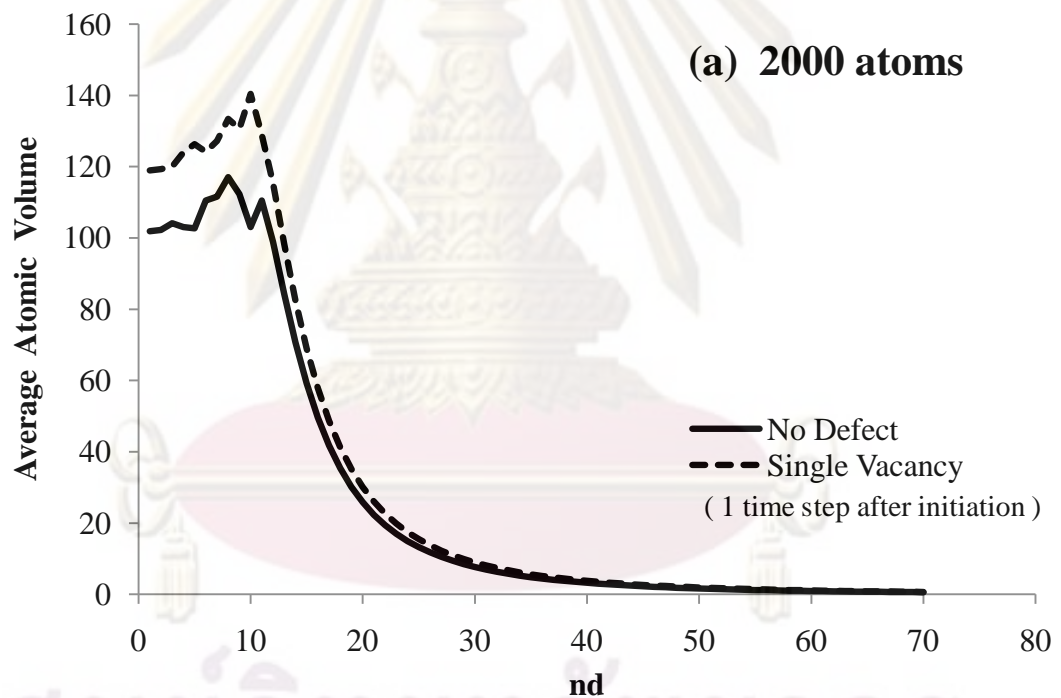


Fig. 4.2 Variation of Average Atomic Volumes with Number of subsystems (n_d);

(a) system with initial 2000 atoms.

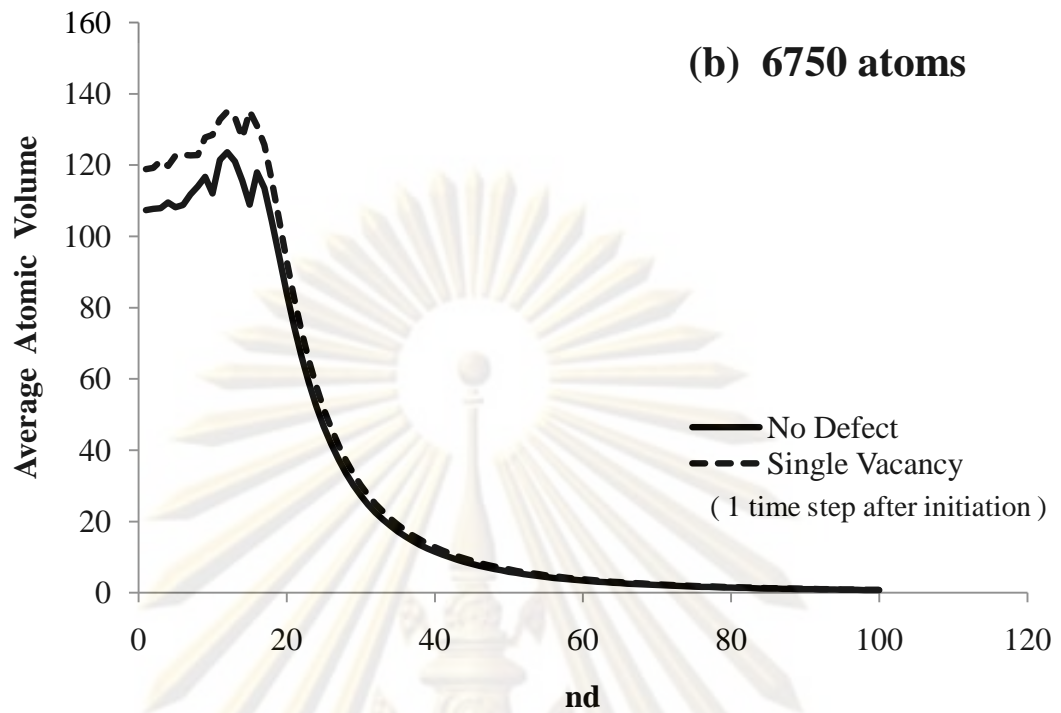


Fig. 4.3 Variation of Average Atomic Volumes with Number of subsystems (n_d);
(b) system with initial 6750 atoms.

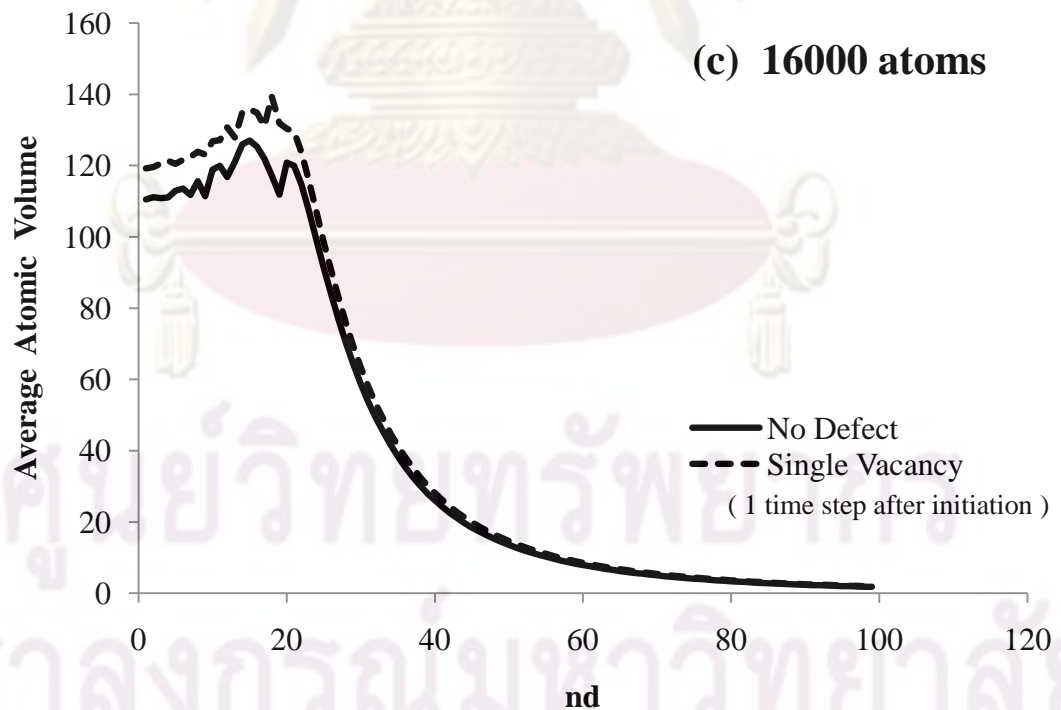


Fig. 4.4 Variation of Average Atomic Volumes with Number of subsystems (n_d);
(c) system with initial 16000 atoms.

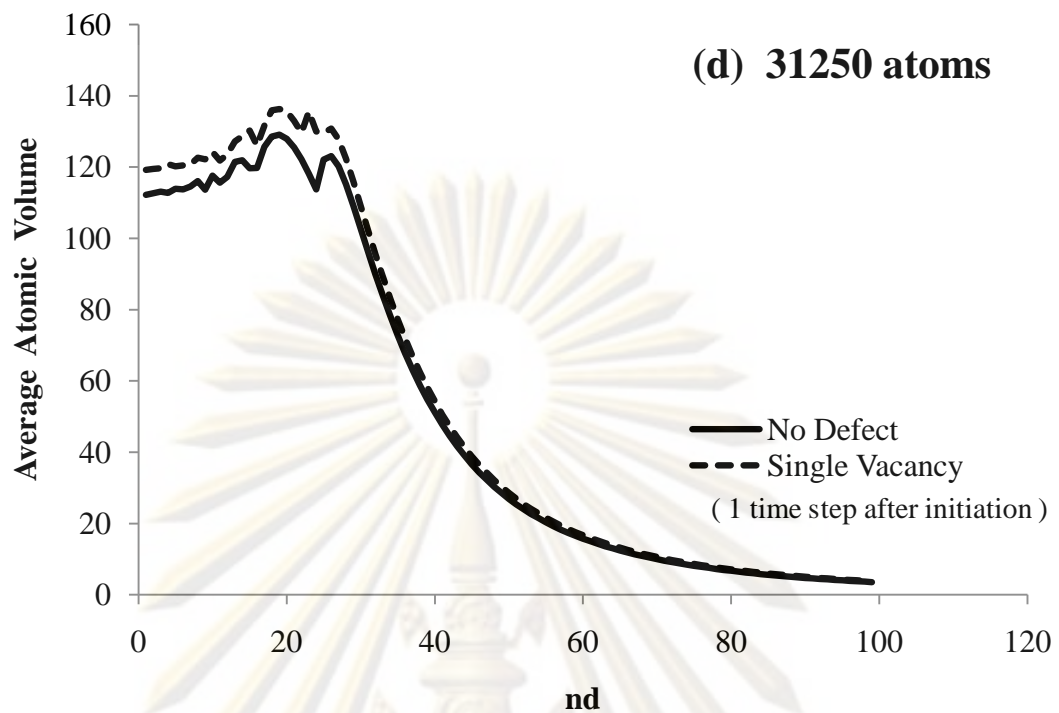


Fig. 4.5 Variation of Average Atomic Volumes with Number of subsystems (n_d);
(d) system with initial 31250 atoms.

4.5 Transients of Single Vacancy Defects

With the values for n_d as previously described, their transients were simulated. Due to the very large number of atoms involved in the simulation and those complex movements, the change in the average atomic volume was chosen to represent the transient. The results as obtained were shown in Fig. 4.6.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

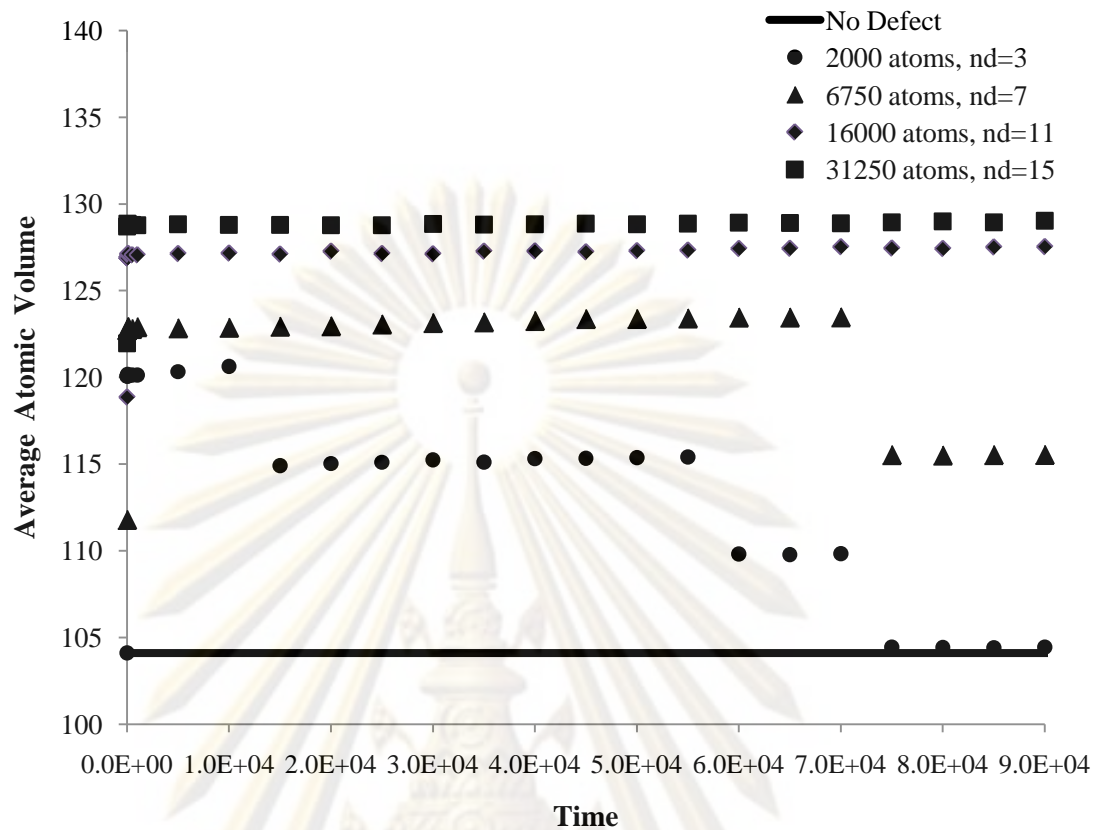


Fig. 4.6 Change in Average Atomic Volume over Times due to single vacancy defect of system with initial 2000 atoms, 6750 atoms, 16000 atoms and 31250 atoms.

It was clearly seen for the cases of 2000 and 6750 initial atoms that the average atomic volumes were suddenly increased. This was due to the space left by the missing atom which allowed the neighbor atoms to expand. The average atomic volume then was gradually decreased as the neighboring atoms were expanding to fill such space. It was observed that the process was slower as the total initial atom was increased. Considering that the expanding was likely caused by the repulsion between atoms, the longer transient period would suggest that there are the shielding effects that retard the expanding process.

4.6 Transients of Multiple Vacancies Defects

For the systems with multiple vacancy defects, the different numbers of vacancies were studied in system of 2000 atoms, which their vacancies atoms were 2-, 3-, 4-, and 5- atoms. The transients of these systems were observed by choosing $n_d =$

3, and the change in the average atomic volume over time of single vacancy showed in Fig. 4.7 to compare with multiple vacancies those were shown in Fig. 4.8-4.11.

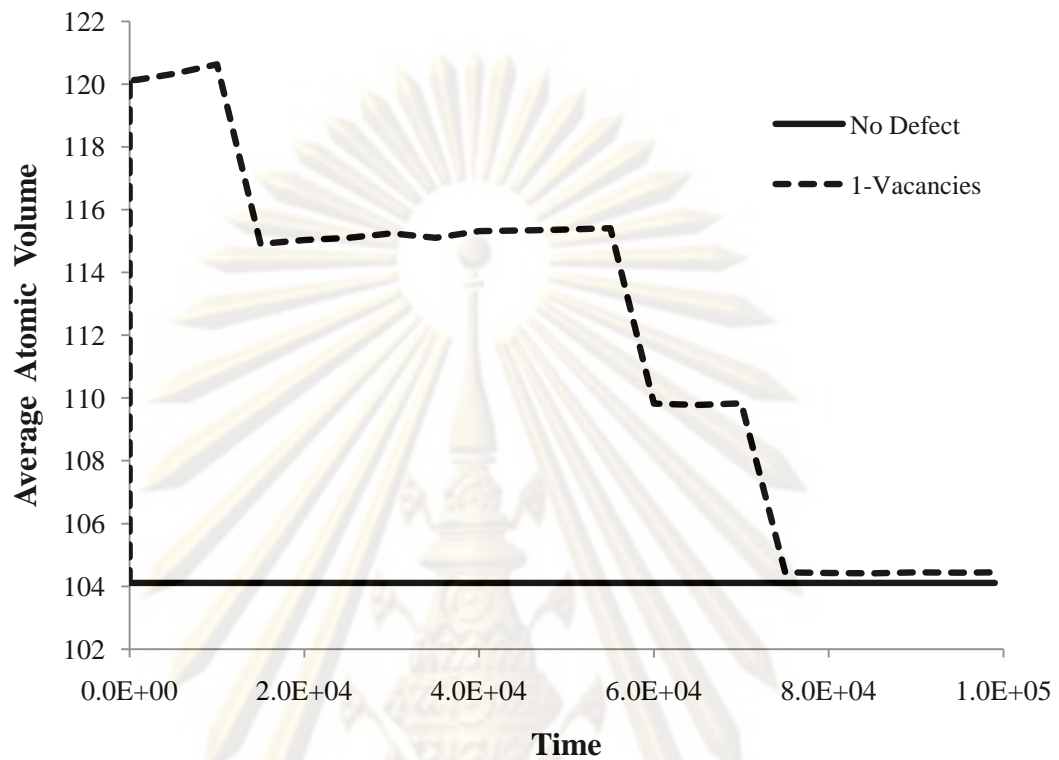


Fig. 4.7 Change in Average Atomic Volume over Times due to single vacancy defect in system with initial 2000 atoms.

It showed that the more vacancies there were, the faster average atomic volume was decreased. The average atomic volume were also found to be varied such that the more vacancies tended to result in the larger final average atomic volume.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

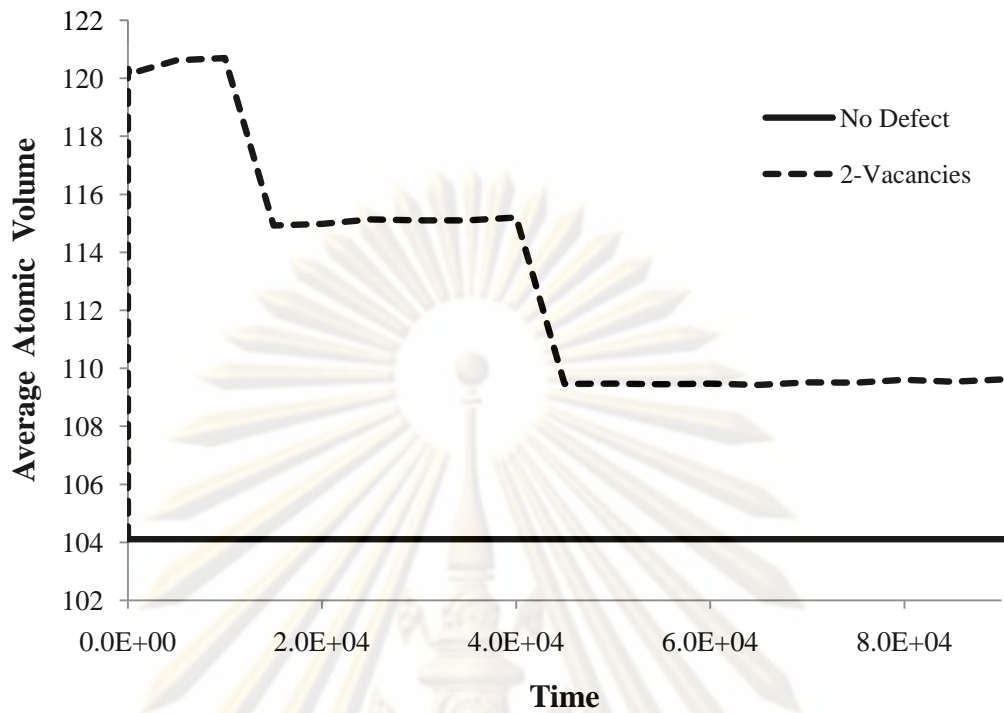


Fig. 4.8 Change in Average Atomic Volume over Times due to 2-vacancies defect in system with initial 2000 atoms.

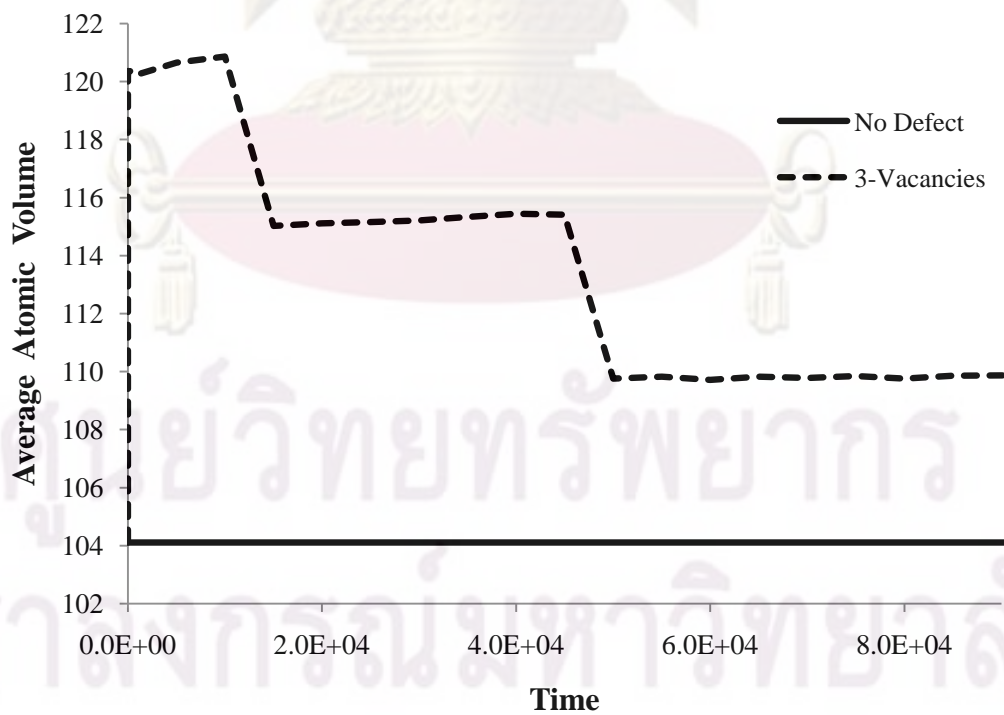


Fig. 4.9 Change in Average Atomic Volume over Times due to 3-vacancies defect in system with initial 2000 atoms.

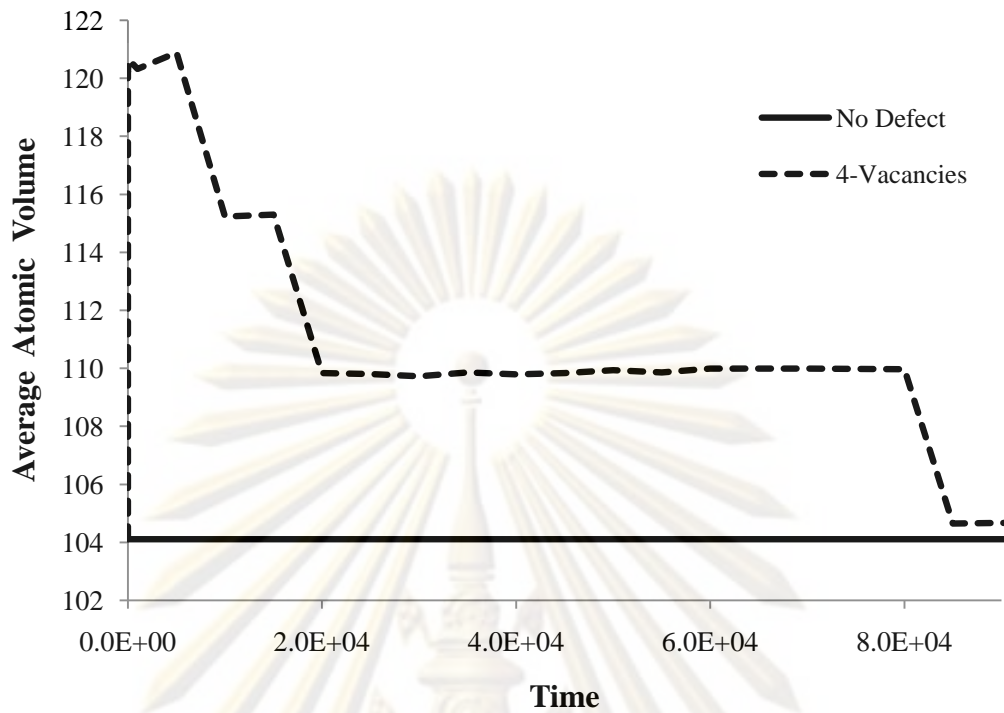


Fig. 4.10 Change in Average Atomic Volume over Times due to 4-vacancies defect in system with initial 2000 atoms.

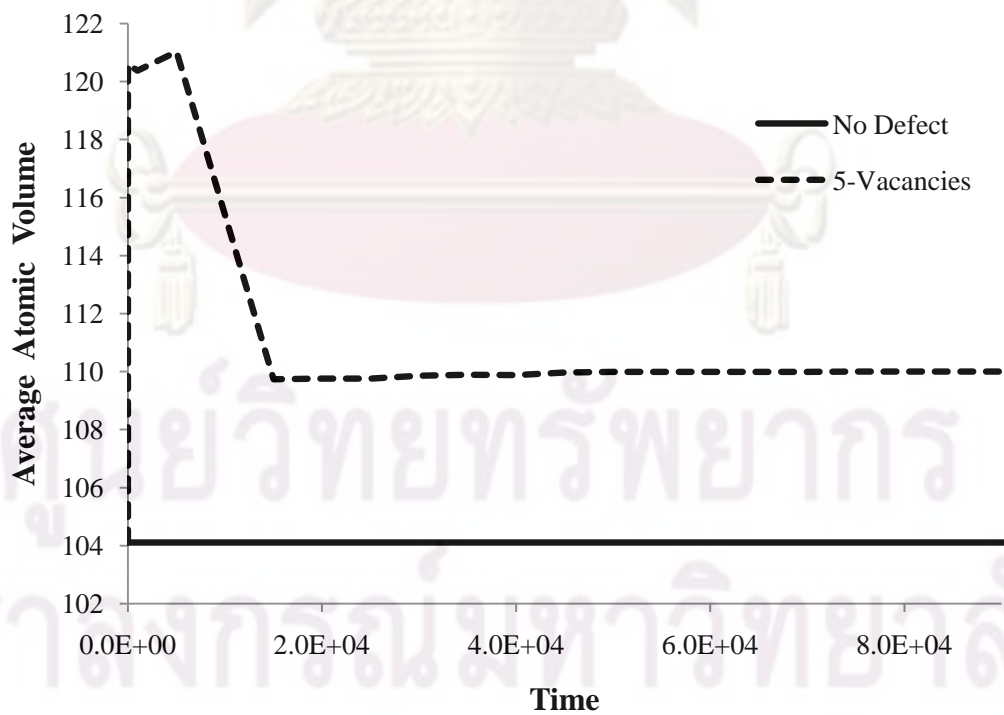


Fig. 4.11 Change in average atomic volume over times due to 5-vacancies defect in system with initial 2000 atoms.

CHAPTER V

CONCLUSION AND SUGGESTION

5.1 Conclusion

The dissertation has studied the phenomena of the point defect transient via two models, the spring-mass model and the ALINE program.

At first, the spring-mass model was created to observe the point defect dynamics in simple cubic crystal by the Hamiltonian calculation in MD simulation method, and then use RK4 is implied as a numerical integrator. This model showed the conservation of Hamiltonian and the vibration of displaced atom over the run time 2000.

In addition, the dissertation has studied the phenomena of the point defect transient via running ALINE program. The study on the transient of the single vacancy defect in the crystalline medium of FCC type was conducted. Due to the range number of atoms and their complex movements, the change in the average atomic volume over time was used to indicate the process of transient.

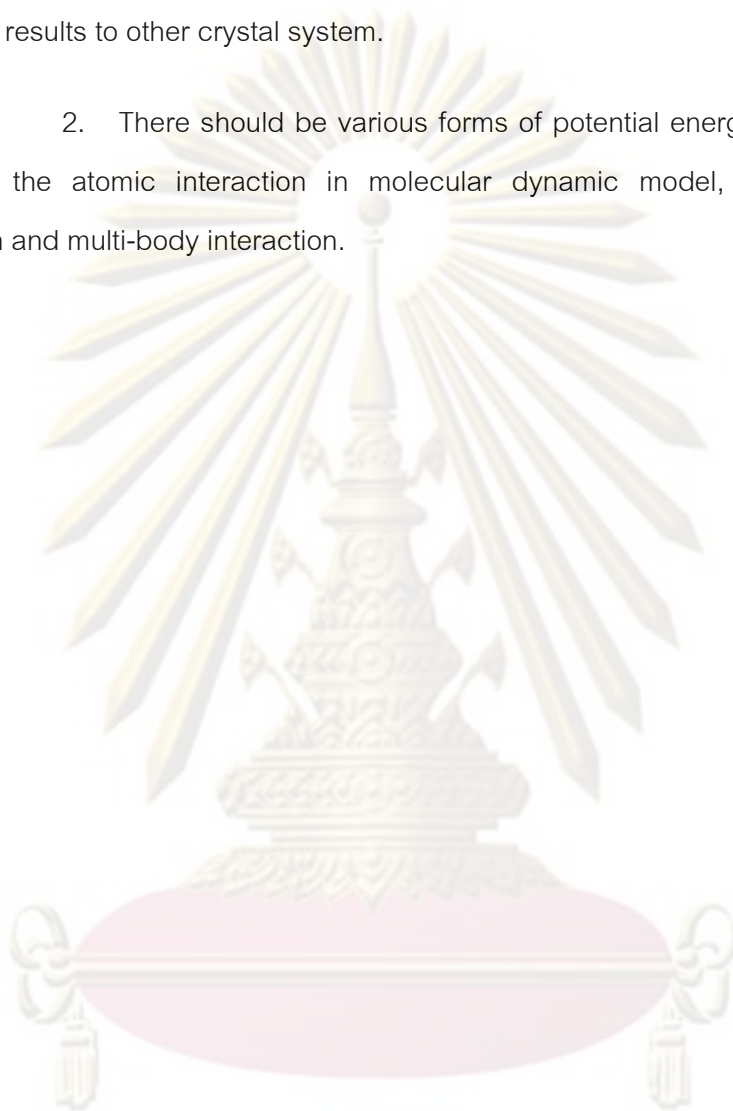
While it was observed in general that the crystalline medium would expand and eventually fill in the space left by the defect, the actual time needed for the process and the actual size of the defect were not indicated and analysed. Even though the similar pattern of transients was to be expected, the effect of defect size and the time needed for transient process were still very crucial. The effect of the defect density and/or the size of the system to be simulated were also of concern and required further investigation.

Even more important was how the type of atoms/ions that formed basis of the crystalline structure affects the transient process. Further study in this regard was necessary. In this case, the actual model for the interaction between atoms/ions must be clearly described and verified.

5.2 Suggestions

1. The study should be conducted in other different phenomena of point defect, such as self-interstitials, or substitutional. Moreover, there should be extending results to other crystal system.

2. There should be various forms of potential energy function that can represent the atomic interaction in molecular dynamic model, both of two-body interaction and multi-body interaction.



ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

REFERENCES

- [1] Kittel, C. Introduction to solid state physics. 8th ed. Hoboken, NJ : Wiley & Sons, 2005.
- [2] Ercolessi, F. A molecular dynamics primer. Spring College in Computational Physics, ICTP, Trieste, June 1997.
- [3] Merimaa, J., Perondi, L.F., and Kaski, K. An interactive simulation program for visualizing complex phenomena in solids. Computer Physics Communications. 124 (2000): 60-75.
- [4] Shilkrot, L.E., Curtin, W.A., and Miller, R.E. A coupled atomistic/continuum model of defects in solids. Journal of the Mechanics and Physics of Solids. 50 (2002): 2085-2106.
- [5] Finnis, M. Interatomic forces in materials. Progress in Materials Science. 49 (2004): 1-18.
- [6] Liu, W.K., Karpov, E.G., Zhang, S., and Park, H.S. An introduction to computational nanomechanics and materials. Computer Methods Applied Mechanics and Engineering. 193 (2004): 1529-1578.
- [7] Lim, T.C. Application of Maclaurin series in relating interatomic potential functions: A review. Journal of Mathematical Chemistry. 36, 2 (2004): 147-160.
- [8] Lim, T.C. Connection among classical interatomic potential functions. Journal of Mathematical Chemistry. 36, 3 (2004): 261-269.
- [9] Lim, T.C. Relationship between the 2-body parameters of the Biswas-Hamann and the Bauer-Maysenholder-Seeger potential functions. Czechoslovak Journal of Physics. 54, 5 (2004): 553-559.
- [10] Patriarca, M., Kuronen, A., Robles, M., and Kaski, K. Computer Physics Communications. 176 (2007): 38-47.
- [11] Forsblom, M. Atomistic simulation of lattice defects. Doctoral dissertation, Department of Physics, Royal Institute of Technology. 2005.
- [12] Young, H.D., and Freedman, R.A. University Physics. 10th ed. San Francisco : Addison-Wesley Publishing, 2000.

- [13] Sutmann, G. Classical Molecular Dynamics. Quantum Simulations of Complex Many-Body Systems : NIC Series, 10 (2002): 211-254.
- [14] Rapaport, D.C. The art of molecular dynamics simulation. 2nd ed. Cambridge : Cambridge University Press, 2004.
- [15] Munetoh, S., Motooka, T., Moriguchi, K., and Shintani, A. Interatomic potential for Si-O systems using Tersoff parameterization. Computational Materials Science. 39 (2007): 2334-339.
- [16] Brenner, D.W., et al. A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons. Journal of Physics: Condensed Matter. 14 (2002) 783-802.
- [17] Raabe, D. Computational material science. Weinheim : Wiley-VCH, 1998.
- [18] van Kampen, N.G. Condensation of a classical gas with long-range attraction. Physical Review. 135 (July 1964): 362-369.
- [19] Kaski K., Binder, K., and Gunton, J.D. Study of cell distribution functions of the three-dimensional Ising model. Physical Review B. 29 (April 1984): 3996-4009.
- [20] Rovere, M., Hermann, D.W., and Binder, K. Block Density Distribution Function Analysis of Two-Dimensional Lennard-Jones Fluids. Europysics Letters. 6 (August 1988): 585-590.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



Appendices

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Appendix A

Program A1: Spring-Mass model; Force calculation by formula method.

```

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>

/*          CONSTANT FOR ALL SYSTEM          */
#define L    1.0          //equilibrium distance
//for Harmonic Potential
#define K    1.0          //force constant
#define MS   1.0          //mass
#define T    0.01        //time step
#define NDIM 30          //mass storage in each axis
#define NDIM3 27000      //NDIM3 = NDIM^3 (mass storage in x,y,z)
#define NDIM33 81000     //NDIM33 = NDIM3*3 (spring storage in x,y,z)
#define D    3           //lattice size (for SC : A*A*A)
#define RT   5000        //run time

/*          INITIAL DATA :: SPRING, POSITION, VELOCITY          */
double fx[NDIM3], fy[NDIM3], fz[NDIM3], ax[NDIM3], ay[NDIM3], az[NDIM3],
       vx[NDIM3], vy[NDIM3], vz[NDIM3], x[NDIM3], y[NDIM3], z[NDIM3];
double x_0[NDIM3], y_0[NDIM3], z_0[NDIM3];
double dvx[4][NDIM3],dvy[4][NDIM3],dvz[4][NDIM3],
       dx[4][NDIM3],dy[4][NDIM3],dz[4][NDIM3];

/*          function :: m2ijkl          */
/*          Convert 1D index for spring to 3D index          */
void m2ijkl(int mm, int ndd, int *mi, int *mj, int *mk, int *ml)
{
    int ii,jj,kk,ll;
    kk=mm/(ndd*ndd*3);

```

```

jj=(mm-kk*ndd*ndd*3)/(3*ndd);
ii=(mm-kk*ndd*ndd*3-jj*ndd*3)/3;
ll=mm-kk*ndd*ndd*3-jj*ndd*3-ii*3;
*mi=ii;
*mj=jj;
*mk=kk;
*ml=ll;
}
/*          function :: ijk2n          */
/*          Convert 3D index for lattice to 1D index          */
int ijk2n(int a, int b, int c, int ndd)
{
int N;
N=a+b*ndd+c*ndd*ndd;
return N;
}
/*          function :: n2ijk          */
/*          Convert 1D index for lattice to 3D index          */
void n2ijk(int nn, int ndd, int *ni, int *nj, int *nk)
{
int iii,jjj,kkk;
kkk=nn/(ndd*ndd);
jjj=(nn-kkk*ndd*ndd)/ndd;
iii=nn-kkk*ndd*ndd-jjj*ndd;
*ni=iii;
*nj=jjj;
*nk=kkk;
}
/*          function :: xleft          */
/*          find 1D index for the lattice connected to the left side of the spring          */
double xleft(int nright,int nleft)

```

```

{
  if(nleft!=nright+(D-1))
    return x[nleft];
  else
    return -(D*L-x[nleft]);
}
/*          function :: yleft          */
double yleft(int nright,int nleft)
{
  if(nleft!=nright+(D*(D-1)))
    return y[nleft];
  else
    return -(D*L-y[nleft]);
}
/*          function :: zleft          */
double zleft(int nright,int nleft)
{
  if(nleft!=nright+(D*D*(D-1)))
    return z[nleft];
  else
    return -(D*L-z[nleft]);
}
/*          function :: xright         */
double xright(int nright)
{
  return x[nright];
}
/*          function :: yright         */
double yright(int nright)
{
  return y[nright];
}

```

```

}
/*          function :: zright          */
double zright(int nright)
{
    return z[nright];
}
/*          Harmonic Potential          */
/*          function :: coeff_xh          */
/*Calculate the magnitude of the force exerted by a spring in the X direction */
double coeff_x(double xa, double xb, double ya, double yb, double za, double zb)
{
    double Fnx;
    Fnx=-K*(1-(L/(sqrt(pow((xa-xb),2)+pow((ya-yb),2)+pow((za-zb),2)))))*(xa-xb);
    return Fnx;
}
/*          function :: coeff_yh          */
double coeff_y(double xa, double xb, double ya, double yb, double za, double zb)
{
    double Fny;
    Fny=-K*(1-(L/(sqrt(pow((xa-xb),2)+pow((ya-yb),2)+pow((za-zb),2))))*(ya-yb);
    return Fny;
}
/*          function :: coeff_zh          */
double coeff_z(double xa, double xb, double ya, double yb, double za, double zb)
{
    double Fnz;
    Fnz=-K*(1-(L/(sqrt(pow((xa-xb),2)+pow((ya-yb),2)+pow((za-zb),2))))*(za-zb);
    return Fnz;
}
/*          function e_k          */
/*          Calculate the kinetic energy of a lattice          */

```

```

double e_k(double vxx, double vyy, double vzz) {
    return 0.5*MS*(vxx*vxx+vyy*vyy+vzz*vzz);
}

/*          function e_p          */
/*    Calculate the potential energy by Harmonic Potential    */
double e_p(double xa, double xb, double ya, double yb, double za, double zb) {
    return 0.5*K*pow(sqrt(pow((xa-xb),2)+
        pow((ya-yb),2)+
        pow((za-zb),2))-L,2);
}

/*          function :: hamiltonian          */
double ham(int nd)
{
    double Hm,Ek,Ep,xl, yl, zl, xr, yr, zr;
    int i, nd3;
    int mii, mjj, mkk, mll;
    int ii, jj, kk;
    int nl, nr;
    nd3=nd*nd*nd;
    Ek=0.0;
    for(i=0; i<nd3; i++)
        Ek=Ek+e_k(vx[i],vy[i],vz[i]);
    nd3=nd3*3;
    Ep=0.0;
    for(i=0; i<nd3; i++) {
        m2ijkl(i,nd,&mii,&mjj,&mkk,&mll);
        nr=ijk2n(mii,mjj,mkk,nd);
        xr=xright(nr);
        yr=yright(nr);
        zr=zright(nr);
        n2ijk(nr,nd,&ii,&jj,&kk);
    }
}

```

```

if(mll==0) {
    ii=ii-1;
    if(ii<0) ii=nd-1;
} else if(mll==1) {
    jj=jj-1;
    if(jj<0) jj=nd-1;
} else {
    kk=kk-1;
    if(kk<0) kk=nd-1;
}
nl=ijk2n(ii,jj,kk,nd);
xl=xleft(nr,nl);
yl=yleft(nr,nl);
zl=zleft(nr,nl);
Ep=Ep+e_p(xl,xr,yl,yr,zl,zr);
}
Hm=Ek+Ep;
return Hm;
}
/*      function :: accelerated calculation      */
void calc_a(double f_x[], double f_y[], double f_z[],
            double *a_xx, double *a_yy, double *a_zz){
int m, n, nd, mii, mjj, mkk, mll, ii, jj, kk, n_r, n_l;
double cx, cy, cz;
nd=D;
for(m=0; m<nd*nd*nd*3; m++){
    m2ijkl(m,nd,&mii,&mjj,&mkk,&mll);
    n_r=ijk2n(mii,mjj,mkk,nd);
    n2ijk(n_r,nd,&ii,&jj,&kk);
    switch(m%3) {
        case 0 :

```

```

if(ii == 0) ii=nd;
n_l=ijk2n(ii-1,jj,kk,nd);
        // x-direction
cx=coeff_x(xright(n_r),xleft(n_r,n_l),
        yright(n_r),yright(n_l),
        zright(n_r),zright(n_l));
f_x[n_r]=f_x[n_r]+cx;
f_x[n_l]=f_x[n_l]-cx;
        // y-direction
cy=coeff_y(xright(n_r),xleft(n_r,n_l),
        yright(n_r),yright(n_l),
        zright(n_r),zright(n_l));
f_y[n_r]=f_y[n_r]+cy;
f_y[n_l]=f_y[n_l]-cy;
        // z-direction
cz=coeff_z(xright(n_r),xleft(n_r,n_l),
        yright(n_r),yright(n_l),
        zright(n_r),zright(n_l));
f_z[n_r]=f_z[n_r]+cz;
f_z[n_l]=f_z[n_l]-cz;
break;
case 1 :
    if(jj == 0) jj=nd;
    n_l=ijk2n(ii,jj-1,kk,nd);
            // x-direction
cx=coeff_x(xright(n_r),xright(n_l),
            yright(n_r),yleft(n_r,n_l),
            zright(n_r),zright(n_l));
f_x[n_r]=f_x[n_r]+cx;
f_x[n_l]=f_x[n_l]-cx;
            // y-direction

```



```

cy=coeff_y(xright(n_r),xright(n_l),
           yright(n_r),yleft(n_r,n_l),
           zright(n_r),zright(n_l));
f_y[n_r]=f_y[n_r]+cy;
f_y[n_l]=f_y[n_l]-cy;
           // z-direction
cz=coeff_z(xright(n_r),xright(n_l),
           yright(n_r),yleft(n_r,n_l),
           zright(n_r),zright(n_l));
f_z[n_r]=f_z[n_r]+cz;
f_z[n_l]=f_z[n_l]-cz;
break;
case 2 :
if(kk == 0) kk=nd;
n_l=ijk2n(ii,jj,kk-1,nd);
           // x-direction
cx=coeff_x(xright(n_r),xright(n_l),
           yright(n_r),yright(n_l),
           zright(n_r),zleft(n_r,n_l));
f_x[n_r]=f_x[n_r]+cx;
f_x[n_l]=f_x[n_l]-cx;
           // y-direction
cy=coeff_y(xright(n_r),xright(n_l),
           yright(n_r),yright(n_l),
           zright(n_r),zleft(n_r,n_l));
f_y[n_r]=f_y[n_r]+cy;
f_y[n_l]=f_y[n_l]-cy;
           // z-direction
cz=coeff_z(xright(n_r),xright(n_l),
           yright(n_r),yright(n_l),
           zright(n_r),zleft(n_r,n_l));

```

```

    f_z[n_r]=f_z[n_r]+cz;
    f_z[n_l]=f_z[n_l]-cz;
    break;
}
}
for(n=0; n<nd*nd*nd; n++){
    *(a_xx+n)=f_x[n]/MS;
    *(a_yy+n)=f_y[n]/MS;
    *(a_zz+n)=f_z[n]/MS;
}
}
/*****
/*          MAIN PROGRAM          */
*****/
main(){

    int i, j, k, l, m, mii, mjj, mkk, mll, n, nd, n_left, n_right, nprint;
    int ii, jj, kk, d_n, d_a, runtime;
    float d_x, d_y, d_z, rd_n, rd_a;
    float t,dt, h0, hn, rkf;
    char filename[20];
    double time;// check running time; in ms. (divide CLOCK_PER_SEC to get in sec.)
    runtime=RT;
    nd=D;
    for(n=0; n<nd*nd*nd; n++) {
        n2ijk(n,nd,&i,&j,&k);
        x[n]=i*1.0;
        y[n]=j*1.0;
        z[n]=k*1.0;
        vx[n]=0.0;
        vy[n]=0.0;

```

```

    vz[n]=0.0;
}
printf("Displaced atom no. : ");
scanf("%d",&d_a);
printf("displaced = ");
scanf("%f",&rd_a);
printf("filename : ");
scanf("%s",filename);

FILE *spring;
spring=fopen(filename,"w");

d_n=d_a;
rd_n=rd_a;
fprintf(spring,"Displaced [%d] from %f %f %f ",d_n,x[d_n],y[d_n],z[d_n]);

x[d_n]=x[d_n]+rd_n;
y[d_n]=y[d_n]+rd_n;
z[d_n]=z[d_n]+rd_n;

d_x=x[d_n];
d_y=y[d_n];
d_z=z[d_n];

fprintf(spring,"to %f %f %f \n\n",x[d_n],y[d_n],z[d_n]);
// Harmonic Potential
fprintf(spring,".: Harmonic Potential ::\n\n");
fprintf(spring,"time step = %f\n\n",T);

h0=ham(nd);
//printf("h0 = %f \n\n",h0);

```

```

//system("pause");

fprintf(spring,"h0= %20.10f\n\n",h0);
dt=T;
t=0;
fprintf(spring,"time h0-nm hn-nm x y z\n");
nprint=0;
do{
  for(n=0; n<nd*nd*nd; n++){ //1st step of rk
    fx[n]=0.0;
    fy[n]=0.0;
    fz[n]=0.0;
    x_0[n]=x[n];
    y_0[n]=y[n];
    z_0[n]=z[n];
  }
  calc_a(fx,fy,fz,ax,ay,az);

  for(n=0; n<nd*nd*nd; n++){
    dvx[0][n]=ax[n]*dt;
    dx[0][n]=vx[n]*dt;
    dvy[0][n]=ay[n]*dt;
    dy[0][n]=vy[n]*dt;
    dvz[0][n]=az[n]*dt;
    dz[0][n]=vz[n]*dt;
  }

  for(n=0; n<nd*nd*nd; n++){ //2nd step of rk
    fx[n]=0.0;
    fy[n]=0.0;
    fz[n]=0.0;
    x[n]=x_0[n]+dx[0][n]/2.0;

```

```

y[n]=y_0[n]+dy[0][n]/2.0;
z[n]=z_0[n]+dz[0][n]/2.0;
}
calc_a(fx,fy,fz,ax,ay,az);
for(n=0; n<nd*nd*nd; n++){
    dvx[1][n]=ax[n]*dt;
    dx[1][n]=(vx[n]+dvx[0][n]/2.0)*dt;
    dvy[1][n]=ay[n]*dt;
    dy[1][n]=(vy[n]+dvy[0][n]/2.0)*dt;
    dvz[1][n]=az[n]*dt;
    dz[1][n]=(vz[n]+dvz[0][n]/2.0)*dt;
}
for(n=0; n<nd*nd*nd; n++){ //3rd step of rk
    fx[n]=0.0;
    fy[n]=0.0;
    fz[n]=0.0;
    x[n]=x_0[n]+dx[1][n]/2.0;
    y[n]=y_0[n]+dy[1][n]/2.0;
    z[n]=z_0[n]+dz[1][n]/2.0;
}
calc_a(fx,fy,fz,ax,ay,az);
for(n=0; n<nd*nd*nd; n++){
    dvx[2][n]=ax[n]*dt;
    dx[2][n]=(vx[n]+dvx[1][n]/2.0)*dt;
    dvy[2][n]=ay[n]*dt;
    dy[2][n]=(vy[n]+dvy[1][n]/2.0)*dt;
    dvz[2][n]=az[n]*dt;
    dz[2][n]=(vz[n]+dvz[1][n]/2.0)*dt;
}
for(n=0; n<nd*nd*nd; n++){ //4th step of rk
    fx[n]=0.0;

```

```

fy[n]=0.0;
fz[n]=0.0;
x[n]=x_0[n]+dx[2][n];
y[n]=y_0[n]+dy[2][n];
z[n]=z_0[n]+dz[2][n];
}
calc_a(fx,fy,fz,ax,ay,az);
for(n=0; n<nd*nd*nd; n++){
    dvx[3][n]=ax[n]*dt;
    dx[3][n]=(vx[n]+dvx[2][n])*dt;
    dvy[3][n]=ay[n]*dt;
    dy[3][n]=(vy[n]+dvy[2][n])*dt;
    dvz[3][n]=az[n]*dt;
    dz[3][n]=(vz[n]+dvz[2][n])*dt;
}
// Update the velocities and the positions for the next time step
for(n=0; n<nd*nd*nd; n++){
    vx[n]=vx[n]+(dvx[0][n]+2.0*dvx[1][n]+2.0*dvx[2][n]+dvx[3][n])/6.0;
    vy[n]=vy[n]+(dvy[0][n]+2.0*dvy[1][n]+2.0*dvy[2][n]+dvy[3][n])/6.0;
    vz[n]=vz[n]+(dvz[0][n]+2.0*dvz[1][n]+2.0*dvz[2][n]+dvz[3][n])/6.0;

    x[n]=x_0[n]+(dx[0][n]+2.0*dx[1][n]+2.0*dx[2][n]+dx[3][n])/6.0;
    y[n]=y_0[n]+(dy[0][n]+2.0*dy[1][n]+2.0*dy[2][n]+dy[3][n])/6.0;
    z[n]=z_0[n]+(dz[0][n]+2.0*dz[1][n]+2.0*dz[2][n]+dz[3][n])/6.0;
}
hn=ham(nd);
if(t==0){
    fprintf(spring,"%5.2f %15.10f %15.10f",t,h0,hn);
    fprintf(spring,"%15.5f %15.5f %15.5f\n",x[d_n],y[d_n],z[d_n]);
}
nprint=nprint+1;

```

```

if(nprint==100){
    fprintf(spring,"%5.2f %15.10f %15.10f",t,h0,hn);
    fprintf(spring,"%15.5f %15.5f %15.5f\n",x[d_n],y[d_n],z[d_n]);
    nprint=0;
}
t=t+dt;           // change to next step;
}while (t<runtime);
    time=clock();
    fprintf(spring,"time usage :: %f ms.\n",time);
system("pause");
}

```

=====

Program A2: Spring-Mass model; Force calculation by numerical method.

```

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
/*          CONSTANT FOR ALL SYSTEM          */
#define L    1.0           //equilibrium distance
//for Harmonic Potential
#define K    1.0           //force constant
#define MS   1.0           //mass
#define T    0.01         //time step
#define dr   0.00000001   //differential step
#define NDIM 30           //mass storage in each axis
#define NDIM3 27000       //NDIM3 = NDIM^3 (mass storage in x,y,z)
#define NDIM33 81000      //NDIM33 = NDIM3*3 (spring storage in x,y,z)
#define D    3            //lattice size (for SC : A*A*A)
#define RT   2000         //run time

```

```

/*          INITIAL DATA :: SPRING, POSITION, VELOCITY          */
double fx[NDIM3], fy[NDIM3], fz[NDIM3], ax[NDIM3], ay[NDIM3], az[NDIM3],
       vx[NDIM3], vy[NDIM3], vz[NDIM3], x[NDIM3], y[NDIM3], z[NDIM3],
       rx[NDIM3], ry[NDIM3], rz[NDIM3];
double x_0[NDIM3], y_0[NDIM3], z_0[NDIM3];
double dvx[4][NDIM3],dvy[4][NDIM3],dvz[4][NDIM3],
       dx[4][NDIM3],dy[4][NDIM3],dz[4][NDIM3];

/*          function :: m2ijkl          */
/*          Convert 1D index for spring to 3D index          */
void m2ijkl(int mm, int ndd, int *mi, int *mj, int *mk, int *ml)
{
  int ii,jj,kk,ll;
  kk=mm/(ndd*ndd*3);
  jj=(mm-kk*ndd*ndd*3)/(3*ndd);
  ii=(mm-kk*ndd*ndd*3-jj*ndd*3)/3;
  ll=mm-kk*ndd*ndd*3-jj*ndd*3-ii*3;
  *mi=ii;
  *mj=jj;
  *mk=kk;
  *ml=ll;
}

/*          function :: ijk2n          */
/*          Convert 3D index for lattice to 1D index          */
int ijk2n(int a, int b, int c, int ndd)
{
  int N;
  N=a+b*ndd+c*ndd*ndd;
  return N;
}

/*          function :: n2ijk          */
/*          Convert 1D index for lattice to 3D index          */

```



```

void n2ijk(int nn, int ndd, int *ni, int *nj, int *nk)
{
    int iii,jjj,kkk;
    kkk=nn/(ndd*ndd);
    jjj=(nn-kkk*ndd*ndd)/ndd;
    iii=nn-kkk*ndd*ndd-jjj*ndd;
    *ni=iii;
    *nj=jjj;
    *nk=kkk;
}

/*          function :: xleft          */
/* find 1D index for the lattice connected to the left side of the spring */
double xleft(int nright,int nleft)
{
    if(nleft!=nright+(D-1))
        return x[nleft];
    else
        return -(D*L-x[nleft]);
}

/*          function :: yleft          */
double yleft(int nright,int nleft)
{
    if(nleft!=nright+(D*(D-1)))
        return y[nleft];
    else
        return -(D*L-y[nleft]);
}

/*          function :: zleft          */
double zleft(int nright,int nleft)
{
    if(nleft!=nright+(D*D*(D-1)))

```

```

    return z[nleft];
else
    return -(D*L-z[nleft]);
}
/*          function :: xright          */
double xright(int nright)
{
    return x[nright];
}
/*          function :: yright          */
double yright(int nright)
{
    return y[nright];
}
/*          function :: zright          */
double zright(int nright)
{
    return z[nright];
}
/*          function e_k          */
/*          Calculate the kinetic energy of a lattice          */
double e_k(double vxx, double vyy, double vzz) {
    return 0.5*MS*(vxx*vxx+vyy*vyy+vzz*vzz);
}
/*          function e_pm          */
/*          Calculate the potential energy by Harmonic Potential          */
double e_p(double xa, double xb, double ya, double yb, double za, double zb) {
    return 0.5*K*pow(sqrt(pow((xa-xb),2)+
        pow((ya-yb),2)+
        pow((za-zb),2))-L,2);
}

```

```

/*          function :: hamiltonian          */
double ham(int nd)
{
    double Hm,Ek,Ep,xl, yl, zl, xr, yr, zr;
    int i, nd3;
    int mii, mjj, mkk, mll;
    int ii, jj, kk;
    int nl, nr;

    nd3=nd*nd*nd;
    Ek=0.0;
    for(i=0; i<nd3; i++)
        Ek=Ek+e_k(vx[i],vy[i],vz[i]);

    nd3=nd3*3;
    Ep=0.0;
    for(i=0; i<nd3; i++) {
        m2ijkl(i,nd,&mii,&mjj,&mkk,&mll);
        nr=ijk2n(mii,mjj,mkk,nd);
        xr=xright(nr);
        yr=yright(nr);
        zr=zright(nr);
        n2ijk(nr,nd,&ii,&jj,&kk);
        if(mll==0) {
            ii=ii-1;
            if(ii<0) ii=nd-1;
        } else if(mll==1) {
            jj=jj-1;
            if(jj<0) jj=nd-1;
        } else {
            kk=kk-1;

```

```

    if(kk<0) kk=nd-1;
}
nl=ijk2n(ii,jj,kk,nd);
xl=xleft(nr,nl);
yl=yleft(nr,nl);
zl=zleft(nr,nl);
Ep=Ep+e_p(xl,xr,yl,yr,zl,zr);
}
Hm=Ek+Ep;
return Hm;
}

```

```

void coeff(double fnx[], double fny[], double fnz[])
{
    double hh, hnewx, hnewy, hnewz;
    int n;
    for(n=0;n<D*D*D;n++){
        rx[n]=x[n];
        ry[n]=y[n];
        rz[n]=z[n];
    }
    hh=ham(D);
    for(n=0;n<D*D*D;n++){
        x[n]=x[n]+dr;
        hnewx=ham(D);
        fnx[n]=-(hnewx-hh)/dr;
        x[n]=rx[n];
        y[n]=y[n]+dr;
        hnewy=ham(D);
        fny[n]=-(hnewy-hh)/dr;
        y[n]=ry[n];
    }
}

```

```

z[n]=z[n]+dr;
hnewz=ham(D);
fnz[n]=-(hnewz-hh)/dr;
z[n]=rz[n];
}
}
/*          function :: accelerated calculation          */
void calc_a(double f_x[], double f_y[], double f_z[],
            double *a_xx, double *a_yy, double *a_zz){
int m, n, nd, mii, mjj, mkk, mll, ii, jj, kk, n_r, n_l;
double cx, cy, cz, cf; // cf = force from function Coeff
nd=D;
for(n=0; n<nd*nd*nd; n++){
*(a_xx+n)=f_x[n]/MS;
*(a_yy+n)=f_y[n]/MS;
*(a_zz+n)=f_z[n]/MS;
}
}
/*****/
/*          MAIN PROGRAM          */
/*****/
main(){
int i, j, k, l, m, mii, mjj, mkk, mll, n, nd, n_left, n_right, nprint;
int ii, jj, kk, d_n, d_a, runtime;
float d_x, d_y, d_z, rd_n, rd_a;
float t,dt, h0, hn, rkf;
char filename[20];

double time; // check running time; in ms. (divide CLOCK_PER_SEC to get in sec.)

```

```

runtime=RT;

nd=D;
for(n=0; n<nd*nd*nd; n++) {
    n2ijk(n,nd,&i,&j,&k);
    x[n]=i*1.0;
    y[n]=j*1.0;
    z[n]=k*1.0;
    vx[n]=0.0;
    vy[n]=0.0;
    vz[n]=0.0;
}
printf("Displaced atom no. : ");
    scanf("%d",&d_a);
printf("displaced = ");
    scanf("%f",&rd_a);
printf("filename : ");
    scanf("%s",filename);

FILE *spring;
spring=fopen(filename,"w");

d_n=d_a;
rd_n=rd_a;
fprintf(spring,"Displaced [%d] from %f %f %f ",d_n,x[d_n],y[d_n],z[d_n]);

x[d_n]=x[d_n]+rd_n;
y[d_n]=y[d_n]+rd_n;
z[d_n]=z[d_n]+rd_n;
d_x=x[d_n];
d_y=y[d_n];

```

```
d_z=z[d_n];
fprintf(spring,"to %f %f %f\n\n",x[d_n],y[d_n],z[d_n]);
```

```
// Harmonic Potential
```

```
fprintf(spring,":: Harmonic Potential ::\n\n");
```

```
fprintf(spring,"time step = %f\n\n",T);
```

```
h0=ham(nd);
```

```
fprintf(spring,"h0= %20.10f\n\n",h0);
```

```
dt=T;
```

```
t=0;
```

```
fprintf(spring,"time h0-nm hn-nm x y z\n");
```

```
nprint=0;
```

```
do{
```

```
for(n=0; n<nd*nd*nd; n++){ //1st step of rk
```

```
fx[n]=0.0;
```

```
fy[n]=0.0;
```

```
fz[n]=0.0;
```

```
x_0[n]=x[n];
```

```
y_0[n]=y[n];
```

```
z_0[n]=z[n];
```

```
}
```

```
coeff(fx,fy,fz);
```

```
calc_a(fx,fy,fz,ax,ay,az);
```

```
for(n=0; n<nd*nd*nd; n++){
```

```
dvx[0][n]=ax[n]*dt;
```

```
dx[0][n]=vx[n]*dt;
```

```
dvy[0][n]=ay[n]*dt;
```

```
dy[0][n]=vy[n]*dt;
```

```
dvz[0][n]=az[n]*dt;
```

```
dz[0][n]=vz[n]*dt;
```

```

}
for(n=0; n<nd*nd*nd; n++){ //2nd step of rk
    fx[n]=0.0;
    fy[n]=0.0;
    fz[n]=0.0;
    x[n]=x_0[n]+dx[0][n]/2.0;
    y[n]=y_0[n]+dy[0][n]/2.0;
    z[n]=z_0[n]+dz[0][n]/2.0;
}
coeff(fx,fy,fz);
calc_a(fx,fy,fz,ax,ay,az);
for(n=0; n<nd*nd*nd; n++){
    dvx[1][n]=ax[n]*dt;
    dx[1][n]=(vx[n]+dvx[0][n]/2.0)*dt;
    dvy[1][n]=ay[n]*dt;
    dy[1][n]=(vy[n]+dvy[0][n]/2.0)*dt;
    dvz[1][n]=az[n]*dt;
    dz[1][n]=(vz[n]+dvz[0][n]/2.0)*dt;
}
for(n=0; n<nd*nd*nd; n++){ //3rd step of rk
    fx[n]=0.0;
    fy[n]=0.0;
    fz[n]=0.0;
    x[n]=x_0[n]+dx[1][n]/2.0;
    y[n]=y_0[n]+dy[1][n]/2.0;
    z[n]=z_0[n]+dz[1][n]/2.0;
}
coeff(fx,fy,fz);
calc_a(fx,fy,fz,ax,ay,az);
for(n=0; n<nd*nd*nd; n++){
    dvx[2][n]=ax[n]*dt;

```



```

dx[2][n]=(vx[n]+dvx[1][n]/2.0)*dt;
dvy[2][n]=ay[n]*dt;
dy[2][n]=(vy[n]+dvy[1][n]/2.0)*dt;
dvz[2][n]=az[n]*dt;
dz[2][n]=(vz[n]+dvz[1][n]/2.0)*dt;
}
for(n=0; n<nd*nd*nd; n++){ //4th step of rk
  fx[n]=0.0;
  fy[n]=0.0;
  fz[n]=0.0;
  x[n]=x_0[n]+dx[2][n];
  y[n]=y_0[n]+dy[2][n];
  z[n]=z_0[n]+dz[2][n];
}
coeff(fx,fy,fz);
calc_a(fx,fy,fz,ax,ay,az);
for(n=0; n<nd*nd*nd; n++){
  dvx[3][n]=ax[n]*dt;
  dx[3][n]=(vx[n]+dvx[2][n])*dt;
  dvy[3][n]=ay[n]*dt;
  dy[3][n]=(vy[n]+dvy[2][n])*dt;
  dvz[3][n]=az[n]*dt;
  dz[3][n]=(vz[n]+dvz[2][n])*dt;
}
// Update the velocities and the positions for the next time step
for(n=0; n<nd*nd*nd; n++){
  vx[n]=vx[n]+(dvx[0][n]+2.0*dvx[1][n]+2.0*dvx[2][n]+dvx[3][n])/6.0;
  vy[n]=vy[n]+(dvy[0][n]+2.0*dvy[1][n]+2.0*dvy[2][n]+dvy[3][n])/6.0;
  vz[n]=vz[n]+(dvz[0][n]+2.0*dvz[1][n]+2.0*dvz[2][n]+dvz[3][n])/6.0;

  x[n]=x_0[n]+(dx[0][n]+2.0*dx[1][n]+2.0*dx[2][n]+dx[3][n])/6.0;

```

```

y[n]=y_0[n]+(dy[0][n]+2.0*dy[1][n]+2.0*dy[2][n]+dy[3][n])/6.0;
z[n]=z_0[n]+(dz[0][n]+2.0*dz[1][n]+2.0*dz[2][n]+dz[3][n])/6.0;
}
hn=ham(nd);
if(t==0){
    fprintf(spring,"%5.2f %10.10f %10.10f ",t,h0,hn);
    fprintf(spring,"%10.5f %10.5f %10.5f\n",x[d_n],y[d_n],z[d_n]);
}
nprint=nprint+1;
if(nprint==1000){
    fprintf(spring,"%5.2f %10.10f %10.10f ",t,h0,hn);
    fprintf(spring,"%10.5f %10.5f %10.5f\n",x[d_n],y[d_n],z[d_n]);
    nprint=0;
}
t=t+dt;           // change to next step;
}while (t<runtime);
    time=clock();
    fprintf(spring,"time usage :: %f ms.\n",time);
system("pause");
}

```

=====

ศูนย์วิทยทรัพยากร
 Program A3: Box-Counting model.

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

จุฬาลงกรณ์มหาวิทยาลัย

```

#define MXI 500
#define MXJ 500
#define MXK 500
#define MXC 10000
#define MXD 71

int box[MXI][MXJ][MXK], count[MXC];
main() {
    char filename[100],filenamer[100],filenameo[100];
    int  maxdata, n, ndiv, i, j, k;
    double xmx, ymx, zmx, xmn, ymn, zmn;
    double x, y, z, dx, dy, dz;
    double vspec_avg, total_count, sumsd, sd;
    FILE *ifile;
    FILE *rfile;
    FILE *ofile;
    maxdata=0;
    xmx=0.0;
    xmn=0.0;
    ymx=0.0;
    ymn=0.0;
    zmx=0.0;
    zmn=0.0;

    printf("Input File : ");
    scanf("%s",filename);
    printf("run no. of divisions in file : ");
    scanf("%s",filenamer);
    printf("Vspect and S.D. in file : ");
    scanf("%s",filenameo);

    ifile=fopen(filename,"r");
    rfile=fopen(filenamer,"a+");

```

```

ofile=fopen(filenameo,"a+");

do {
fscanf(ifile,"%d%lf%lf%lf%lf\n",&n,&x,&y,&z);
maxdata=maxdata+1;
if(x>xmx) xmx=x;
if(x<xmn) xmn=x;
if(y>ymx) ymx=y;
if(y<ymin) ymn=y;
if(z>zmx) zmx=z;
if(z<zmin) zmn=z;
} while(!feof(ifile));
fclose(ifile);
for(ndiv=1;ndiv<MXD;ndiv++){
fprintf(rfile,"no. of divisions = %d\n",ndiv);
dx=(xmx-xmn)/ndiv;
dy=(ymx-ymin)/ndiv;
dz=(zmx-zmin)/ndiv;
xmx=xmx+0.001*dx;
xmn=xmn-0.001*dx;
ymx=ymx+0.001*dy;
ymn=ymn-0.001*dy;
zmx=zmx+0.001*dz;
zmn=zmn-0.001*dz;
fprintf(rfile,"maxdata xmx xmn ymx ymn zmx zmn Volume\n");
fprintf(rfile,"%10d %10.4lf %10.4lf %10.4lf %10.4lf %10.4lf %10.4lf
%10.4lf\n",
maxdata,xmx,xmn,ymx,ymn,zmx,zmn,
(xmx-xmn)*(ymx-ymin)*(zmx-zmin));
dx=(xmx-xmn)/ndiv;
dy=(ymx-ymin)/ndiv;

```

```

dz=(zmx-zmn)/ndiv;
fprintf(rfile,"dx dy dz\n");
fprintf(rfile,"%10.4f %10.4f %10.4f
%10.4f\n",dx,dy,dz,dx*dy*dz*ndiv*ndiv*ndiv);

for(i=0; i<ndiv; i++)
for(j=0; j<ndiv; j++)
for(k=0; k<ndiv; k++)
box[i][j][k]=0;

ifile=fopen(filename,"r");
do {
fscanf(ifile,"%d%lf%lf%lf\n",&n,&x,&y,&z);
i=floor((x-xmn)/dx);
j=floor((y-ymn)/dy);
k=floor((z-zmn)/dz);
box[i][j][k]=box[i][j][k]+1;
} while(!feof(ifile));
fclose(ifile);
for(i=0; i<MXC; i++)
count[i]=0;
for(i=0; i<ndiv; i++)
for(j=0; j<ndiv; j++)
for(k=0; k<ndiv; k++)
count[box[i][j][k]]=count[box[i][j][k]]+1;
vspec_avg=0.0;
total_count=0.0;
for(i=0; i<MXC; i++)
if((count[i]>0) && (i>0)) {
fprintf(rfile,"%5d %10d %10.4f\n",i,count[i],dx*dy*dz/i);

```

```
total_count=total_count+count[i];
vspec_avg=vspec_avg+dx*dy*dz/i*count[i];
}
vspec_avg=vspec_avg/total_count;
fprintf(rfile,"%10lf %10.4lf",total_count,vspec_avg);

sumsd=0.0;
for(i=0; i<MXC; i++)
if((count[i]>0) && (i>0)) {
sumsd=sumsd+(count[i]*pow(dx*dy*dz/i-vspec_avg,2));
}
sd=pow(sumsd/total_count,0.5);
fprintf(rfile," %10.4lf\n\n",sd);
fprintf(ofile,"%d %10.4lf %10.4lf\n",ndiv,vspec_avg,sd);
}
fclose(rfile);
fclose(ofile);
}
```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Appendix B

Procedure to run ALINE program

1. Download program from <http://cpc.cs.qub.ac.uk/>
2. Expand the package by the commands:


```
gunzip aline.tar.gz
tar -xvf aline.tar
```
3. Compile the program by the commands:


```
xmkmf
Make Makefiles
make
```
4. Run program with command line:


```
./aline
```
5. Program could be started with the main GUI (Graphical User Interface) as shown in Fig. B1.
6. At pull down menu, click 'Program' and select 'Make a new configuration', the window as shown in Fig. B2.
7. Input data for system of interest:
 - CRYSTAL TYPE;
 - SIZE;
 - BOUNDARY CONDITIONS;
 - (Border Width)/(Lattice Constant) Ratio;
 - $T(t=0)$;
 Then click 'APPLY' and 'CONFIRM'.
8. Create the defect by pull down menu 'Specimen' and select 'CRACK' as shown in Fig. B3, to input data of
 - CRACK CENTER;
 - CRACK SIZE;
 - CRACK ANGLE;
 Then click 'ON/OFF' and 'CONFIRM'.

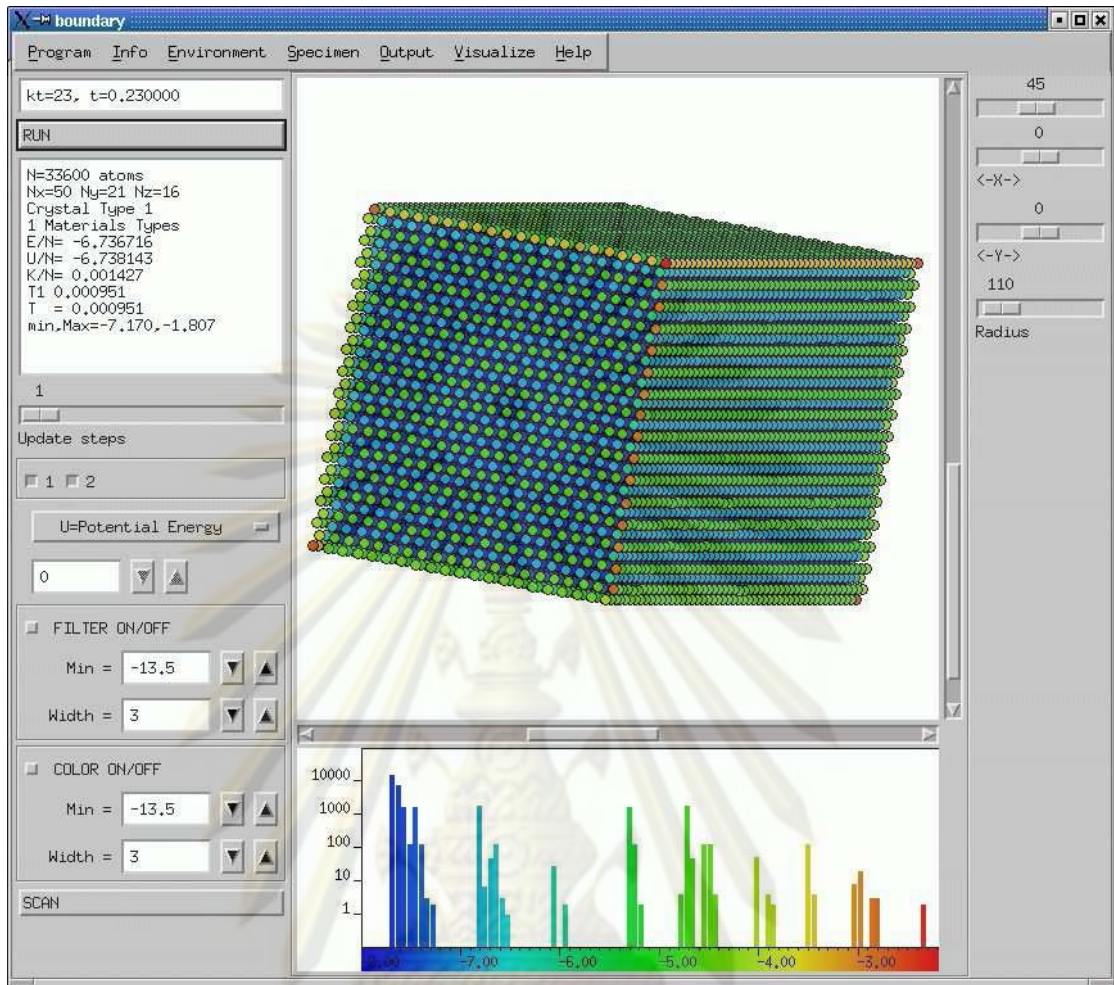


Fig. B1 The main GUI window of ALINE.



Fig. B2 Initial condition builder window appearing at the starting of the program.

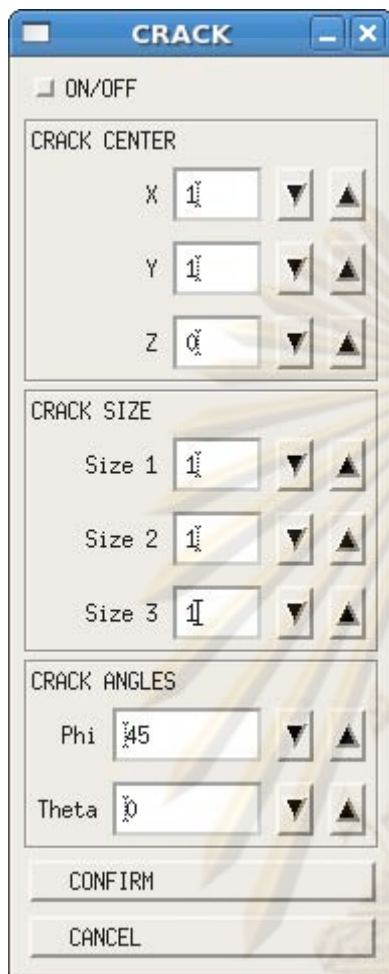


Fig. B3 Window to create defects.



Fig. B4 Window to set the time for

recoding the output files.

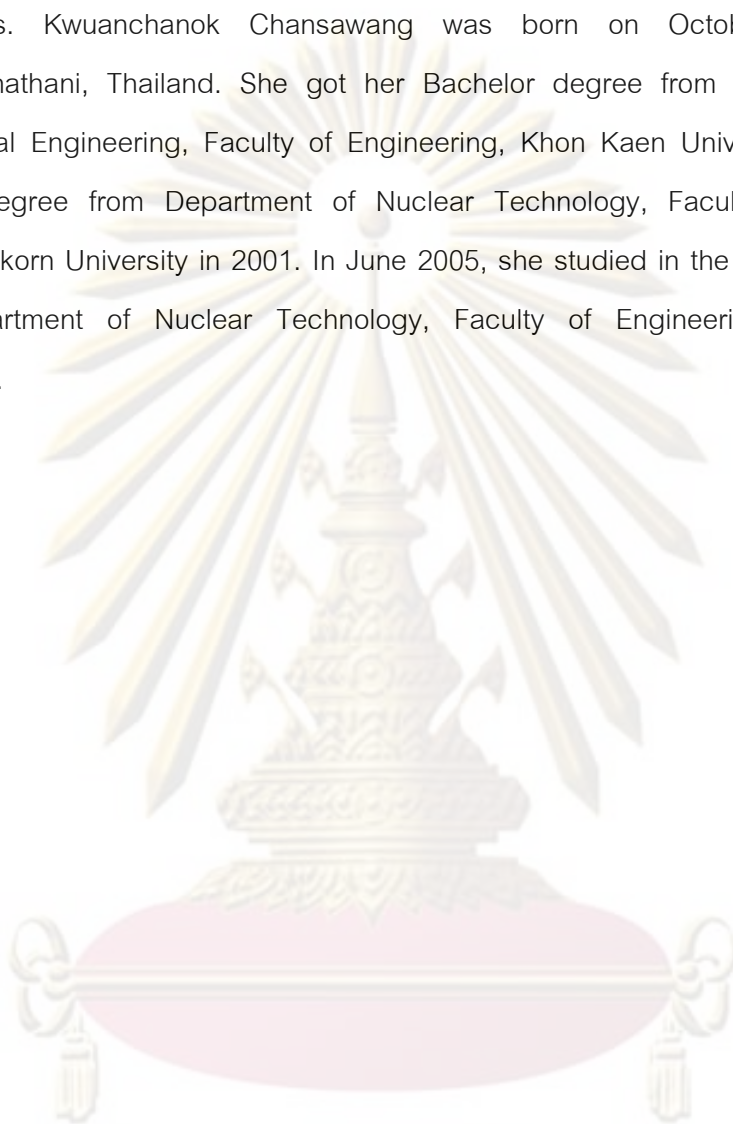
9. Setting time to record the coordinate of each atom at the pull down menu 'Output' and select 'ANIMATION' as shown in Fig. B4.
10. At the main GUI, click 'RUN'.
11. For more information, please see reference no. [10].



ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Biography

Ms. Kwuanchanok Chansawang was born on October 29, 1975 in Ubonratchathani, Thailand. She got her Bachelor degree from the Department of Agricultural Engineering, Faculty of Engineering, Khon Kaen University in 1998 and Master degree from Department of Nuclear Technology, Faculty of Engineering, Chulalongkorn University in 2001. In June 2005, she studied in the Doctoral degree at the Department of Nuclear Technology, Faculty of Engineering, Chulalongkorn University.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย