

A Highly Scalable Simulation Model for Atomistic Calculation of Thermal Properties of Silicon

Lin Sun
Mechanical Engineering
Purdue University
West Lafayette, IN, U.S.A.

Chinh Le
Rosen Center for Advanced Computing
ITaP, Purdue University
West Lafayette, IN, U.S.A.

Faisal Saied
Rosen Center for Advanced Computing
ITaP, Purdue University
West Lafayette, IN, U.S.A.

Jayathi Y. Murthy
Mechanical Engineering
Purdue University
West Lafayette, IN, U.S.A.

Abstract: A portable parallel program developed for molecular dynamics simulations shows excellent scalability on a variety of architectures including IBM's BlueGene, IBM's Power4 P655+, and Linux clusters. A task that would have taken 30 days on a single processor can now be completed in 2.5,5 hours on 1024 Power4+ processors and 1728 BlueGene's PowerPC processors respectively, using our parallel molecular dynamics algorithm. This program computes the thermal conductivity of bulk silicon. In particular, we investigate how computational domain size (number of atoms) affects the predicted accuracy. We also explore the thermal conductivity of thin film silicon at room temperature. Results show that, as the thickness increases up to 300 nm, the thermal conductivity approaches the bulk thermal conductivity. This work permits us to confidently apply molecular dynamics to the simulation of thermal conductivity of both bulk materials and thin films.

Keywords: scalability, massively parallel, molecular dynamics, thermal conductivity, MPI.

Introduction

Recent advances in modeling, algorithms and computing technology have made it feasible to determine fundamental thermal properties of materials from atomistic calculations using molecular dynamics (MD). The advantage of using MD is that once an empirical potential is chosen, no fitting parameters are necessary and anharmonic effects are included in an intrinsic way. However, very large computational resources are required to perform MD simulations on a large space and time scales with complex interatomic potentials. Thermal conductivity computations employing Green-Kubo formalism¹ are particularly onerous because statistics must be collected over long time periods. Although computer power has been growing rapidly in the last decade, typical thermal conductivity computations on a desktop PC may require 50 CPU hours for systems as small as 10^3 atoms. This

leads to a formidable task even for a system with a characteristic length of 100 nm, and greatly limits the applicability of MD simulations for real systems. Therefore, the concept of spreading the work over many processors to perform the task at reasonable time cost is a natural one.

Bulk silicon and thin silicon films are widely used in the semiconductor industry. It is essential to precisely predict their thermal conductivities in order to analyze relevant thermal problems in silicon-based devices. Typical simulations are done with only a few thousands atoms with periodic boundary conditions to save on computational cost. The effects of finite simulation domain have been discussed in several papers and the results are not consistent with each other^{1,2,3}. In Volz's work, for example, a maximum atom number reaching 64000 was used, but computed values of bulk thermal conductivity were an order of magnitude smaller than measurements, necessitating a correction to

computed values for long-wavelength phonons. Domains of 1728 atoms were used in Schelling's work, and values of bulk thermal conductivity within 30% of experimentally measured values were predicted. Nearly all domain sizes investigated are smaller than the mean free path of silicon at room temperature, of the order of 300 nm. On the other hand, domains with a few thousand atoms would be sufficient to capture the dominant wavelength at room temperature or higher.

Similar issues arise with thin films. Currently, the silicon thin film deposited in the transistors only has a thickness of few nanometers. It has been found that the thermal conductivity of thin films can be reduced by an order of magnitude over bulk because of the phonon confinement and phonon-boundary scattering⁴. Since it is very expensive and difficult to perform experimental measures at nanometer scale, MD simulation provides an easier way to predict these properties. However, it is unclear how well MD simulations can capture thin film properties and what domain sizes are sufficient.

In this paper, we present a parallel program for MD simulations and implement it to predict the silicon thermal conductivity both for bulk and thin films. We describe first the general methodology used in MD simulation and then the parallel implementation. Two test problems, the first computing the bulk thermal conductivity of silicon using Green-Kubo method, and the second investigating the normal thermal conductivity of thin-film silicon using non-equilibrium MD are described. Performance measures for IBM's BlueGene, IBM's Power4 P655+, and Linux clusters are presented.

Theoretical background

Molecular Dynamics simulation

Classical molecular dynamics computes the evolution of a N-atom system with time. The atoms are regarded as particles with mass m_i , and obey Newton's second law:

$$m_i \frac{d^2 \vec{r}_i}{dt^2} = \sum_{j=1, j \neq i}^N \vec{F}_{ij} \quad (1)$$

where m_i is the mass of atom i , \vec{r} is the position vector. \vec{F}_{ij} is the force vector exerted by atom j on atom i , and can be derived from the interatomic potential. In order to precisely describe the interaction between atoms, a suitable potential must

be chosen. A variety of three-body potentials, such as the Stillinger-Weber⁵ and Tersoff⁶ potentials have been used to simulate silicon. In our simulations, a newly-developed empirical potential for silicon called Environment Dependent Interatomic Potential (EDIP)^{7,8,9} is used. This potential permits each atom only interact with its four nearest neighbors. Thus, the time cost on force calculation is greatly reduced.

The procedure for a typical MD simulation is (1) Set up the initial conditions, for instance, the domain size, system temperature, and initial position and velocity; (2) Calculate the interaction forces between atoms based on the potential function; (3) Calculate the acceleration from forces and integral it with time to get the velocity and position at the next step; (4) Analyze the data and calculate the properties of interest; (5) Write out the data; (6) Use the new position and velocity data as the starting point and repeat step (2-5) until the desired total time has elapsed.

In our simulations, the initial configuration of atoms employs the lattice structure of silicon in which two interpenetrating FCC structures are displaced along the body diagonal by one-fourth the diagonal. The initial velocity of each atom is randomly distributed corresponding to the initial temperature. A micro-canonical ensemble (NVE) is employed for the simulations, so that the total number of atoms, volume and energy of system are conserved. The leap-frog technique¹⁰ is used for time-integration of Newton's equation of motion. We assume that the boundaries in all three directions are periodic so that the atoms leaving a bounding surface re-enter the domain through the corresponding periodic face on the opposite side. A time step $\Delta t = 10^{-15}$ sec (i.e. 1 fs) is used in all simulations presented here. The total running time is in the range of 1-3 ns (10^6 MD steps); the first 100 ps of simulation are used to relax the system to equilibrium. Beyond this point, data are gathered for statistical analysis.

Thermal conductivity prediction

Two methods are commonly used in thermal conductivity prediction, equilibrium MD (EMD) and non-equilibrium MD (NEMD). EMD is also referred as the Green-Kubo method, while NEMD is referred to direction simulation method.

In this work, bulk thermal conductivity is predicted using the Green-Kubo method. As shown in eqn. (2), the thermal conductivity is defined as

integral of the autocorrelation of heat current vectors.

$$\lambda_{\alpha\beta} = \frac{1}{Vk_b T^2} \int_0^{\infty} \langle J_{\alpha}(\tau) \cdot J_{\beta}(0) \rangle dt \quad (2)$$

where V is the system volume, k_b is the Boltzmann constant, T is the system temperature, and the subscripts α and β represent the three Cartesian coordinates. $\vec{J}(\tau)$ is the heat current vector as a function of time, and is given by:

$$\vec{J}(\tau) = \frac{d}{d\tau} \left[\sum_i r_i (m_i v_i^2 / 2 + U_i) \right] \quad (3)$$

where m is atomic mass, r and v are respectively the atomic position and velocity. U represents potential energy. The subscript i is atom index. The Green-Kubo method allows the determination of the complete thermal conductivity tensor. For silicon, the thermal conductivity along the high-symmetry directions [100], [010] and [001] are expected to be the same. The bulk thermal conductivity shown in this paper is the mean thermal conductivity found by averaging the values in each of these directions, unless otherwise noted.

In contrast, the key idea of NEMD is to set up a temperature gradient or impose a heat flux across the simulation domain. Then the thermal conductivity can be calculated by Fourier's Law:

$$q = \lambda A \frac{\Delta T}{\Delta l} \quad (4)$$

In order to achieve this goal, the system is divided into three regions, heating, cooling and conduction. At each time step, a certain amount of energy is added to the atoms in heating region and the same amount is subtracted from cooling region. Over a sufficient time, a steady heat flux can be obtained across the domain. This approach is used to predict the thin film thermal conductivity and the velocity-rescaling algorithm for heating and cooling is taken from reference¹¹.

Parallel MD program

Consider the MD simulation of a system with one million atoms. This number corresponds to a silicon cube with 27 nm length. Run serially, a typical MD job would require 10^6 steps and takes more than 30 days to complete on a Power4 P655+ processor. The force calculation is found to take approximately 90% of the total time. However, the algorithm for force calculation is rigid and no adjustment is allowed. Therefore, the only feasible

partition scheme to parallelize the MD program is to decompose the entire domain into small sub-regions. Each sub-region is assigned to a particular processor. All the atoms in a sub-region reside on a given processor and all related variables are also localized on that processor. Thus, no global arrays are necessary and the need for large memory is avoided. Based on these considerations, the flow chart of the parallel program is shown in Fig. 1.

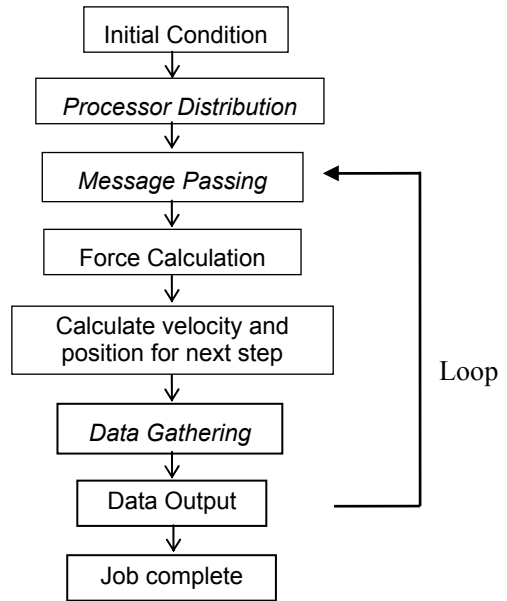


Figure 1. Flow chart of parallel MD program

Compared with a typical serial MD program, only three extra steps are added shown by italic characters in Fig. 1. In the *Processor Distribution* stage, all the processors involved are arranged in a Cartesian coordinate system as shown in Fig. 2. By labeling of coordinates (x,y,z), each processor knows its location and neighbors. The entire simulation domain fits in these processor arrays. Simultaneously all the atoms are evenly distributed ensuring that the load is distributed evenly over all processors. Processors are initially synchronized and code optimization is performed.

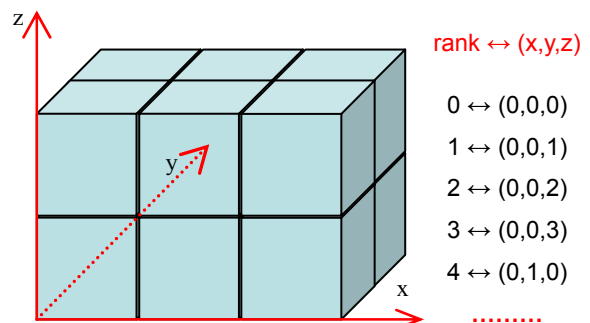


Figure 2. Illustration of processor distribution.

Table 1. System configuration of the three computer platforms tested.

Name	Macbeth (MB)	DataStar (DS)	BlueGene (BG)
Node Type	AMD Opteron 145	IBM Power4 P655+/P690+	PowerPC
Number of nodes	98	272/7	1024
CPUs per Node	2	8/32	2
Memory per Node	4 GB	16/256 GB	512MB
CPU Speed	1.8 GHz	1.5/1.7 GHz	700MHz
Interconnection	Gigabit Ethernet	IBM switch	IBM switch

Since atoms interact with each other, processors need to communicate with each other. Fortunately, the interatomic forces for silicon atoms are short-range, which means that most of the interactions happen inside of the subregion and relatively few atoms at the boundary need to interact with those from adjacent processors. To accommodate the latter, the atoms near the boundary must be identified, and the information regarding boundary atoms on processor i is sent to its neighbors. Simultaneously the information regarding boundary atoms from each neighbor is sent to processor i . This involves only a small fraction of the total number of atoms. In order to store the message transferred from neighbors, a ghost zone is created around the local zone and local arrays need to be expanded by a certain length which is dependent on the message size. Besides this interaction, atoms are allowed to move from one processor to another, and the associated variables are transferred explicitly. This movement does not occur frequently for a condensed solid phase, since all the atoms just fluctuate with a small magnitude around their equilibrium positions. All these message exchanges are done in the *Message Passing* stage in each loop. Furthermore, at the end of every computation loop, relevant data is collected from each processor to make statistical measurements over all the atoms. This is done in the *Data Gathering* stage. For the parallel program to be efficient, the computation time must be much larger than the communication time. Computation load depends on the simulation algorithm, which is generally not adjustable. So communication time must be minimized.

As discussed above, each processor needs to communicate with its surrounding processors. This involves nine processors for two-dimensional and twenty- seven processors for three-dimensional applications. To reduce the communication time, the message exchange is performed in such a way that each processor only needs to talk with the processors which share the common faces but not

the ones at the corners. Figure 3 shows this communication in 2-D problem. In this way, each processor only communicates with four face neighbors and the messages from the four corners are automatically included. Applying this to the 3-D case, each processor only needs to communicate with six instead of twenty-six processors.

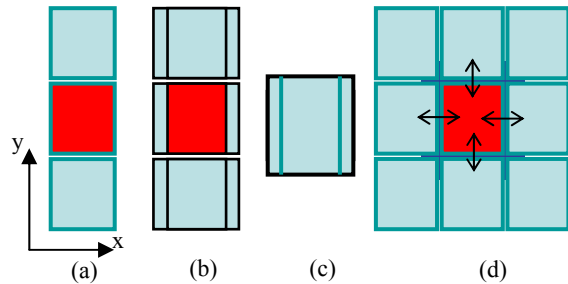


Figure 3: 2-D illustration of message exchange between processors. (a) Consider three processors; b) Each processor sends and receives messages along the x direction from adjacent neighbors and stores them in ghost zone; (c) Each processor sends and receives messages along the y direction, which include the data from local and ghost zone; (d) By communicating with four face neighbors, the messages from the four corners are automatically included.

To reduce communication times, all the associated data for boundary atoms are packed into one buffer and sent out. After the message is received, the data is unpacked and reverts to its original format. Therefore, in each simulation loop, each processor only sends out six messages and receives six messages in 3D. Several combinations of blocking and unblocking MPI commands¹² were tested; it was found that unblocking sends and receives work best for our simulation. Specifically, six “Irecv” were posted first, then “Isend” along three directions were posted in turn, finally “Wait” calls were made to complete the message passing. This set of routines was found to perform best amongst all the combinations tested.

Performance

A general parallel program developed to simulate three dimensional applications with a

variable number of atoms and on a variable number of processors was tested on three parallel platforms (Table 1). This program is highly portable requiring no code modification. The Makefile needs adjustment only with regards to compilers and their options.

A simulation task of 10^6 atoms requiring 10^6 MD steps was implemented. The total time to complete the task is presented as a function of the number of processors in Fig 4. The total times on 1024 DS's Power4+ processors and 1728 BG's PowerPC processors are about 2.5 and 5 hours respectively. In contrast, the total serial time on a single Power4+ processor is approximately 30 days. The scalability of the parallel program and the hardware is presented in Fig. 5 which plots the inverse of the total time (proportional to the speedup) versus the number of processors. In Fig. 5, the dashed line represents the ideal speedup, which is a 45° line. The performance of the MD parallel program is most efficient on BG. The BG PowerPC processor is much slower than the AMD Opteron and IBM Power4 processor, while its network is fast (approximately about 10 times faster than Giga Ethernet on the linux cluster). Therefore, the computation time / communication time ratio for each BG processor is large even when the number of processor is large.

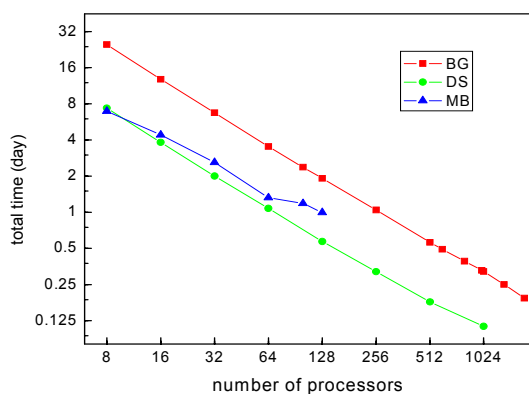


Figure 4: The total time for an MD simulation of 10^6 atoms and 10^6 steps on BlueGene, DataStar and Macbeth clusters.

The efficiency of the parallel MD program can also be examined by comparing communication and computation time. The smaller the communication time is relative to the computation time, the more parallel the program, and closer to the ideal speedup. Therefore, the goal is to minimize the

communication time. The communication and computation times on BG, DS and MB are respectively presented in Fig. 6. Communication in BG and DS is about ten times faster than in MB; thus the communication time in BG and DS is much smaller than the computation time. For MB, communication and computation times become comparable as the number of processors reaches 64; the parallel MD program's efficiency is only about 50% and decreases rapidly. A high degree of efficiency and scalability on DS and BG gives us a robust frame for further realistic applications.

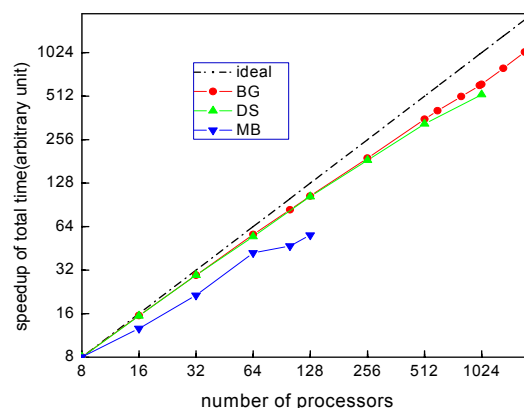


Figure 5: Parallel MD program scalability on BlueGene, DataStar and Macbeth clusters.

Thermal conductivity predictions

The Green-Kubo method is used to quantify the domain size effect on the prediction of thermal conductivity in EDIP silicon. We simulate a series of cubic domains with atoms number ranging from 64 to 216,000. A convergence in the thermal conductivity value is found to occur when the system size is above 512 atoms at 300K as shown in Fig. 7. For each simulation case, five independent runs are performed and three high-symmetry directions are considered. Thus, the error bar on each data point is obtained by averaging fifteen values. It can be seen that simulation results match experimental data to about 20%, a reasonable accuracy for MD simulations. This is despite the fact that our computational domain is much smaller than the mean free path (at room temperature, the mean free path of silicon is approximately 260 nm^{13}). The dominant wavelength at 300K is $1\sim 2 \text{ nm}$; this length scale is captured by our domain sufficiently well that thermal conductivity predictions are obtained relatively accurately. As the

domain size increases, the predicted value remains fairly constant while the statistical error is improved. This suggests that a relatively small domain can be used to predict bulk thermal conductivities if periodic boundary conditions are applied. A number of different runs are necessary to obtain a reasonable average value. A large domain has a smaller statistical error, but it requires more computational resources and time.

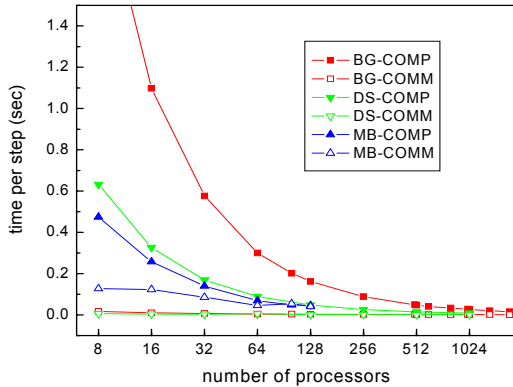


Figure 6. Comparison of computation time and communication time at each time step.

In a second demonstration of thermal conductivity calculation, we apply non-equilibrium MD to calculate the normal thermal conductivity of thin film silicon. The cross section is 5×5 nm with periodic boundaries, and thickness ranges from 10 nm to 350 nm which covers the phonon ballistic and diffusive limit. The maximum atom number reaches one million. The results are shown in Fig. 8. As the thickness becomes smaller than the phonon MFP, phonon transport is ballistic and constrained by the boundaries. In this regime, the thermal conductivity is nearly proportional to the thickness. When the thickness is greater than MFP, transport is diffusive and the thermal conductivity approaches the bulk value. These results are in good agreement with previous theoretical predictions^{14,15}.

Conclusion

A parallel molecular dynamics program was developed and used to predict the thermal conductivity of solid phase materials. Inter-processor communication in the program is optimized and a high degree of efficiency is reached. The program is portable and can run on different platforms without code modification. Three different parallel architectures including IBM's

BlueGene, IBM's Power4 P655+, and AMD Opteron/GigE Linux clusters were used and a high degree of scalability was observed for massively parallel applications (60% and 53% for 1024 P655+ processors and 1728 BG PowerPC processors respectively). This program was used to investigate the effects of computational domain size on the prediction of bulk silicon thermal conductivity. The thermal conductivity for silicon thin film structures at room temperature as a function of thickness was also explored. Results show that, as the thickness increases to 300 nm, the computed thermal conductivity approaches the bulk value gradually. This work established the viability of using molecular dynamics to simulate thermal properties in both bulk and nanoscale domains.

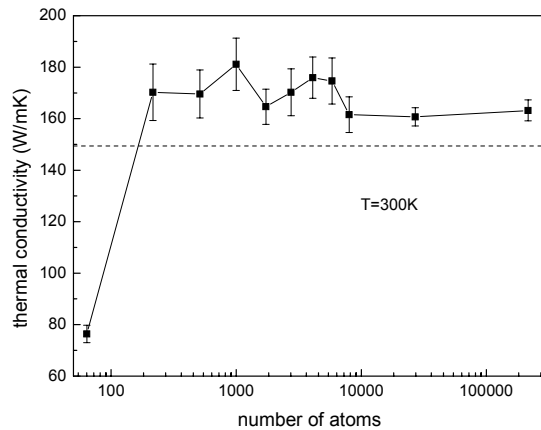


Figure 7: Predicted thermal conductivity of bulk silicon using Green-Kubo method as a function of number of atoms. The experimental value is 148 W/mK.

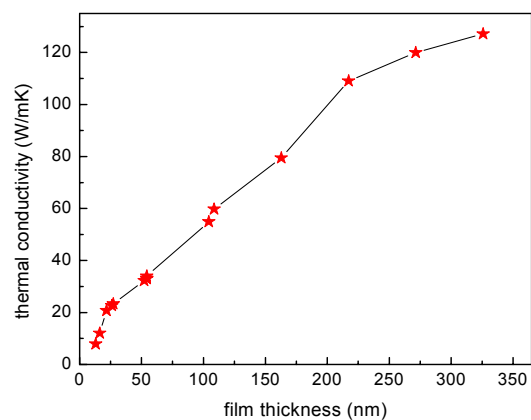


Figure 8: Thermal conductivity in normal direction of silicon thin film as a function of thickness.

ACKNOWLEDGMENTS

Support of L. Sun and J.Y. Murthy under NSF grants EEC-0228390, CTS-0312420 and CTS-0219098 is gratefully acknowledged. The parallel MD program was run on San Diego Supercomputing Center's IBM BlueGene and IBM Power4+ DataStar, and on the Macbeth Linux cluster of the Rosen Center for Advanced Computing at Purdue University.

-
- [1] Schelling, P.K., Phillpot S.R., and Keblinski, P., "Comparison of Atomic-level Simulation Methods of Computing Thermal Conductivity", *Phys. Rev. B*, Vol. 65, 144306, pp 1-12. 2002.
- [2] Che, J.W., Cagin, T., Deng, W. and Goddard, W.A., "Thermal Conductivity of Diamond and Related Materials from Molecular Dynamics Simulations", *Journal of Chemical Physics*, Vol. 113, No.16, pp6888-6900. 2000.
- [3] Volz, S. and Chen, G., "Molecular-Dynamics Simulation of Thermal Conductivity of Silicon Crystals", *Phys. Rev. B*, Vol. 61, pp. 2651-2656. 2000.
- [4] Ju Y. S., "Phonon Heat Transport in Silicon Nanostructures", *Applied Physics Letters*, Vol. 87, 153106, 2005.
- [5] Stillinger, F. and Weber, T. A., "Computer Simulation of Local Order in Condensed Phases of Silicon", *Phys. Rev. B* 31, pp5262-5271. 1985.
- [6] Tersoff J., "New Empirical Approach for the Structure and Energy of Covalent Systems", *Phys. Rev. B* 37, pp 6991-7000. 1988.
- [7] Bazant, M.Z., and Kaxiras, E., "Modeling of Covalent Bonding in Solids by Inversion of Cohesive Energy Curves", *Physical Review Letters*, Vol.77, No21, pp 4370-4373. 1996.
- [8] Bazant, M. Z. and Kaxiras, E., and Justo, J. F., "Environment Dependent Interatomic Potential for Bulk Silicon", *Phys. Rev. B* 56, pp8542,1997.
- [9] Justo, J. F., Bazant, M. Z., E. etc., "Interatomics Potential for Silicon Defects and Disordered Phases", *Phys. Rev. B* 58, pp 2539. 1998.
- [10] Rapaport, D.C., "The Art of Molecular Dynamics Simulation", Cambridge University Press. 1995.
- [11] Ikeshoji T. and Hafskjold B., "Non-equilibrium Molecular Dynamics Calculation of Heat Conduction in Liquid and through Liquid Gas Interface", *Molecular Physics*, Vol 81. No.2, pp251-261. 1994.
- [12] William Gropp, Ewing Lusk, Anthony Skjellum, "Using MPI, portable parallel programming with the message passing interface", second edition, the MIT press. 1999.
- [13] Chen, G., "Thermal Conductivity and Ballistic-phonon Transport in the Cross-plane direction of Superlattices", *Phys. Rev. B*, Vol. 57, No.23, pp 14958- 14973. 1998.
- [14] Qiheng Tang, "A Molecular Dynamics Simulation: the Effect of Finite Size on the Thermal Conductivity

in a Single Crystal Silicon", *Molecular Physics*, Vol. 102, pp1959. 2004.

- [15] Gomes C. J., Madrid M., Goicochea J. V. and Amon C. H., "Silicon Thin Film Thermal Conductivity in Ballistic and Diffusive Regimes Predicted by Molecular Dynamics", *Proceedings of ASME Summer Heat Transfer Conference*, July San Francisco, California. 2005.