LA-UR-95- 438

Conf-9408218--2

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

TITLE: Applications of Langevin and Molecular Dynamics Methods

AUTHOR(S): Peter S. Lomdahl

SUBMITTED TO: Proceedings of "Nonlinearity and Disorder" Summer School, University of Madrid, Spain 1994

aug. 15-19, 1994

MASTER

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO. 836 R4
ST. NO. 2629 5/81

DISCLAIMER

Applications of Langevin and Molecular Dynamics Methods

P. S. Lomdahl
pxl@viking.lanl.gov
Theoretical Division and Advanced Computing Laboratory
Los Alamos National Laboratory, Los Alamos, New Mexico 87545

January 24, 1995

ABSTRACT

The basic principles behind Langevin stochastic simulation and Molecular Dynamics (MD) methods are outlined and illustrated with examples from realistic applications to a number of nonlinear problems in condensed matter physics and materials science.

1. INTRODUCTION

Computer simulation of complex nonlinear and disordered phenomena from materials science is rapidly becoming an active and new area serving as guide for experiments and for testing of theoretical concepts. This is especially true when novel massively parallel computer systems and techniques are used on these problems. In particular the Langevin dynamics simulation technique has proven useful in situations where the time evolution of a system in contact with a heat bath is to be studied. The traditional way to study systems in contact with a heat bath has been via the Monte Carlo method. While this method has indeed been used successfully in many applications, it has difficulty addressing true dynamical questions. Large systems of coupled stochastic ODEs (or Langevin equations) are commonly the end result of a theoretical description of higher dimensional nonlinear systems in contact with a heat bath. The coupling is often local in nature, because it reflects local interactions formulated on a lattice, the lattice for example represents the underlying discreteness of a substrate of atoms or discrete k-values in Fourier space. The fundamental unit of parallelism thus has a direct analog in the physical system we are interested in.

In these lecture notes we will illustrate the use of Langevin stochastic simulation techniques on a number of nonlinear problems from materials science and condensed matter physics that have attracted attention in recent years. First, we will review the idea behind the fluctuation-dissipation theorem which forms that basis for the numerical Langevin stochastic simulation scheme. We then show applications of the technique to various problems from condensed matter and materials science.

2. SIMPLE STOCHASTIC THEORY AND NUMERICAL SCHEME

2.1 Fluctuation-Dissipation Theory

In this section we consider the basic fluctuation-dissipation theorem[1] which forms the basis for equilibrium statistical mechanics and the numerical schemes we use. Let us consider a simple stochastic differential equation:

$$M\frac{d^2x}{dt^2} + M\Gamma\frac{dx}{dt} + M\omega_0 x = \xi(t). \tag{1}$$

Which has the general solution:

$$\begin{aligned} x(t) &= e^{-\Gamma t/2}[A\cos\omega t + B\sin\omega t] \\ &\quad + \int_{t_0}^{t} d\tau \frac{1}{M\omega} e^{-\Gamma(t-\tau)/2}\sin[\omega(t-\tau)]\xi(\tau) \end{aligned} \tag{2}$$

where $\omega = \sqrt{\omega_0 + (\frac{\Gamma}{2})^2}$. It is also convenient to introduce the Green's function as:

$$G(t) = \frac{1}{M\omega} e^{-\Gamma t/2}\sin[\omega t] \tag{3}$$

The random noise force has the following characteristics $\xi(t)$:

$$\langle \xi(t) \rangle = 0 \quad , \quad \langle \xi(t)\xi(t_0) \rangle = C\delta(t-t_0), \tag{4}$$

and the probability distribution:

$$P(\xi) = \frac{e^{-\xi^2/2\sigma}}{\sqrt{2\pi\sigma}}. \tag{5}$$

We will determine C so that the system is in thermal equilibrium. Using the general solution (2), we can write its time derivative as

$$\begin{aligned} \dot{x}(t) &= e^{-\Gamma t/2}\{[-\frac{\Gamma}{2}A + \omega B]\cos\omega t + [-\frac{\Gamma}{2}B - \omega A]\sin\omega t\} \\ &\quad + \int_{t_0}^{t} d\tau \dot{G}(t-\tau)\xi(\tau). \end{aligned} \tag{6}$$

where we have introduced the time derivative of the Green's function (3):

$$\dot{G}(t) = \frac{1}{M\omega} e^{-\Gamma t/2}\{-\frac{\Gamma}{2}\sin\omega t + \omega\cos\omega t\}. \tag{7}$$

We can now write the time derivative of the general solution (6), $\dot{x}(t)$, as containing a deterministic and a stochastic part:

$$\dot{x}(t) = \dot{x}_d(t) + \dot{x}_\xi(t). \tag{8}$$

or

$$\dot{x}^2(t) = \dot{x}_d{}^2(t) + 2\dot{x}_d(t)\dot{x}_\xi(t) + \dot{x}_\xi{}^2(t). \tag{9}$$

Taking the ensemble average we get

$$\langle \dot{x}^2 \rangle = \dot{x_d}^2(t) + 2\dot{x_d}(t)\langle \dot{x_\xi}(t) \rangle + \langle \dot{x_\xi}^2(t) \rangle. \tag{10}$$

By using (4) we get in the limit $t \to \infty$:

$$\langle \dot{x}^2 \rangle = \langle \dot{x_\xi}^2(t) \rangle. \tag{11}$$

By inserting (6) we further get:

$$\begin{aligned} \langle \dot{x}^2 \rangle &= \langle \int_{t_0}^{t} d\tau_1 \dot{G}(t-\tau_1)\xi(\tau_1) \int_{t_0}^{t} d\tau_2 \dot{G}(t-\tau_2)\xi(\tau_2) \rangle \\ &= \int_{t_0}^{t} d\tau_1 \int_{t_0}^{t} d\tau_2 \dot{G}(t-\tau_1)\dot{G}(t-\tau_2)\langle \xi(\tau_1)\xi(\tau_2) \rangle. \\ &= C \int_{t_0}^{t} d\tau_1 [\dot{G}(t-\tau_1)]^2 \\ &= C \int_{0}^{t-t_0} ds [\dot{G}(s)]^2 \end{aligned} \tag{12}$$

where we have used $\dot{G}(s)$ as given by (7). It is easy to verify that in the limit $t \to \infty$ (12) reduces to:

$$\int_0^\infty ds [\dot{G}(s)]^2 = \frac{1}{2M^2\Gamma}. \tag{13}$$

or

$$\lim_{t\to\infty} \langle \dot{x}(t)^2 \rangle = \frac{C}{2M^2\Gamma}, \tag{14}$$

which can also be written

$$\lim_{t\to\infty} \langle \frac{M}{2} \dot{x}(t)^2 \rangle = \frac{C}{4M\Gamma}. \tag{15}$$

This is supposed to be $\frac{1}{2}k_BT$ by equilibrium statistical mechanics, i.e.:

$$\frac{C}{4M\Gamma} = \frac{1}{2}k_BT \;\Rightarrow\; C = 2k_BTM\Gamma. \tag{16}$$

So when the random force correlation function (4) is given by

$$\langle \xi(t)\xi(t_0) \rangle = 2k_BTM\Gamma\delta(t-t_0), \tag{17}$$

the kinetic energy is approaching its equilibrium value.

2.2 Numerical Schemes for SDEs

The numerical treatment of stochastic differential eqautions (SDE) like (1) has not kept up with the significant advances that have taken place in the treatment of ODEs in the past couple of decades. However, Greenside and Helfand[2] provides several numerical schemes to deal systematically with SDEs. Their work is based on extensions to the conventional Runge-Kutta scheme for ODEs.

In the standard Runge-Kutta method for $\frac{dx}{dt} = f(x)$, $f(x)$ is evaluated at definite intermediate points. These are used to extrapolate to $\hat{x}(t+h)$ which is accurate to a given order in the time step $O(h^k)$. For example, the simple Euler method is given by:

$$x_{n+1} = x_n + hf(t_n, x_n) \tag{18}$$

However, this scheme is not very good, the error $O(h^2)$ and the scheme is generally not stable. But applying the scheme a second time and taking the average, yields a 2nd order Rung-Kutta scheme:

$$x_{n+1} = x_n + \frac{h}{2}[f(t_n, x_n) + f(t_n + h, x_n + hf(t_n, x_n))]. \tag{19}$$

which has an error $O(h^3)$.

For SDEs $f(x)$ is evaluated at stochastically selected points. The algorithm is such that all moments of $\hat{x}(t+h) - x(t)$ is correct to $O(h^k)$.

$$\frac{dx}{dt} = f(x) + \xi(t), \tag{20}$$

where

$$\langle \xi(t) \rangle = 0 \quad , \quad \langle \xi(t)\xi(t') \rangle = \alpha\delta(t - t'). \tag{21}$$

The scheme is given by:

$$\begin{aligned} g_1 &= f(t_n, x_n), \\ g_2 &= f(t_n + h, x_n + hg_1 + \sqrt{h\alpha}Z), \\ x_{n+1} &= x_n + \frac{h}{2}(g_1 + g_2) + \sqrt{h\alpha}Z. \end{aligned} \tag{22}$$

Where Z is Gaussian random variable. The proof that scheme is $O(h^2)$ can be found in [2] which also presents higher order schemes.

3. SURFACE GROWTH

The dynamics of growth processes of thin films and other surfaces is of considerable technological interest and has been the subject of intense experimental studies since the advent of novel STM and AFM techniques. In this section we will briefly outline how computational modeling of surface growth is particularly well suited for the Langevin stochastic simulation scheme outline in the previous section. We have adopted a modified dynamic "solid-on-solid" model which captures the basic physics of surface growth [3, 4]. We present the essential features of our model and refer the reader to ref. [5] for a more detailed account of our results.

We consider the following Hamiltonian:

$$\mathcal{H} = \frac{1}{2}\sum_{i,j}(\phi_i - \phi_j)^2 - \sum_i \cos(\phi_i) + I\sum_i \phi_i, \tag{23}$$

where ϕ_i is a continuous variable measuring the height of a 2D surface at lattice site i. The first term in the Hamiltonian favors growth close to sites that already *have* grown, i. e. the energy is minimized when ϕ_i is equal to the value at its neighboring sites. The sum is over nearest neighbors only. The second term in the Hamiltonian favors values of ϕ_i close to $2\pi n$ corresponding to layers of atoms. The last term is a uniform driving force corresponding to the chemical potential difference between surface and ambient vapor. This Hamiltonian is nothing but the discrete version of the sine-Gordon equation in two space dimension. The time evolution of ϕ_i is given by the following equations of motion which we have also augmented with Langevin noise and dissipation:

$$\frac{d^2\phi_i}{dt^2} = \sum_j(\phi_{i+j} - \phi_i) + \sin(\phi_i) + I + \epsilon\frac{d\phi_i}{dt} \div \lambda_i(t), \tag{24}$$

where the noise is taken as Gaussian:

$$\langle\lambda_i(t)\rangle \quad , \quad \langle\lambda_i(t)\lambda_j(t')\rangle = 2\epsilon T\delta_{ij}\delta(t-t'). \tag{25}$$

We are primarily interested in dynamics of growth in the presence of defects like screw dislocations. The screw dislocation can be analytically approximated by the following continuum expression:

$$\phi(x,y) = \tan^{-1}\left(\frac{x}{y}\right) \equiv \phi_0. \tag{26}$$

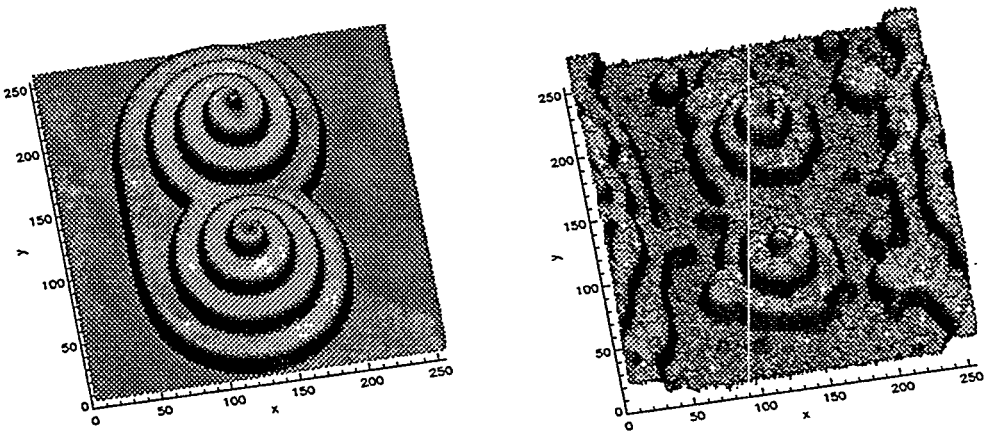

Fig. 1. Snapshot from the time evolution of a simulation of (24) $I = 0.8$, and $\epsilon = 1.0$. The Frank-Read source acts as a seed for spiral surface growth. In (a) $T = 0$ and in (b) $T = 1.0$.

However, this expression for ϕ_0 does not satisfy the continuum static equation of motion. In order for the screw dislocation to be a static solution to the equations of motion we consider growth *relative* to ϕ_0, i. e. the argument to the sine function in (1) is modified to include ϕ_0. The static continuum equation is $\nabla^2\phi = \sin(\phi - \phi_0)$.

We note here that ϕ_0 can represent an ensemble of screw dislocations. In many simulations we have taken ϕ_0 to represent a pair of fixed screw dislocations with opposite Burger's vectors, i. e. a Frank-Read source for crystal growth.

Direct simulations of (24) show clear evidence for spiral surface growth due to the Frank-Read source. In Fig. 1a we show a 3D perspective shaded surface plot of a snapshot well into the time evolution of the growth process. This figure was obtained for essentially zero temperature and $I = 0.8$ and $\epsilon = 1.0$. The spacing between the screw dislocations in the Frank-Read source is 100 lattice spacings. In Fig. 1b we show the corresponding picture for a simulation with $T = 1.0$. It is clear from this figure, that as well as spiral growth additional growth is occurring via homogeneous nucleation. In this situation we are in a regime where the two growth mechanisms are coexisting and competing. We can summarize the basic physics of our model in a temperature-pressure phase diagram which is shown in Fig. 2. The presence of defects like the screw dislocations of the the Frank-Read source dramatically change the dynamics of the growth process at low temperatures and driving forces (pressure). The critical force for growth to occur is strongly decreased in the presence of defects. The dynamics of the spiral growth is characterized by $\langle\langle \frac{d\phi_i}{dt} \rangle\rangle \approx I^2$, where the double brackets indicate time and space averages. We also note that our system exhibits a Kosterlitz-Thouless transition at high fields and temperatures where the presence of screw dislocation defects is unimportant. We again refer the interested reader to ref. [5]. for more details.

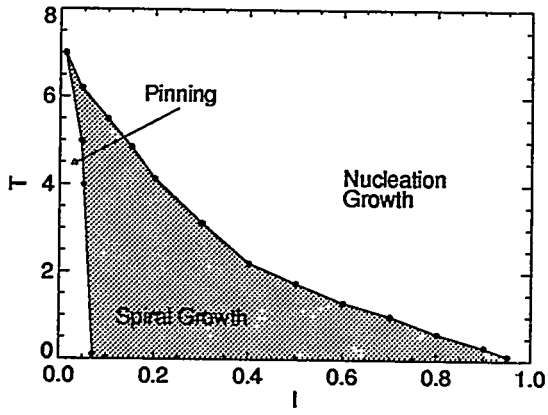

Fig. 2. Schematic phase diagram showing regimes of spiral growth, lattice discreteness pinning, and growth via homogeneous nucleation. From ref. [5].

We have illustrated how Langevin stochastic simulation methods running on massively parallel multicomputers makes modeling of 2D nonlinear dynamic processes like surface growth easy and interactive. We want to stress that our model of surface growth is indeed simple, but it nevertheless captures the essential physics of the underlying process. It is straight forward to modify the model to include more realistic (materials specific) potentials, effects of disorder, diffusing cores (dynamic Frank-Read sources), and coupling to reaction-diffusion systems (hereby adding chemistry).

All of these changes can readily be accommodated within the Langevin dynamics formulation.

4. LARGE-SCALE MOLECULAR DYNAMICS

How does a piece of metal break? How do cracks propagate? What are the effects of impurities and grain boundaries? These are only a few of many interesting questions in materials science research. To tackle many of these questions, computer simulation is playing a greater role than ever due to the rapidly increasing capabilities of high performance supercomputers[6, 7, 8]. For several decades, the method of molecular dynamics (MD) has been used to study properties of materials at an atomistic level [9]. The idea behind an MD simulation is very simple; one sets up a large collection of atoms (in a crystal lattice for example) and directly solves Newton's equations of motion $F = ma$. While conceptually simple, this task presents a formidable computing problem. If the atoms interact according to a pair potential (e.g. gravity, coulomb, Van der Waals, etc.) the direct solution of this general N body problem will require calculating nearly $N(N-1)/2$ forces. To complicate matters, the atoms in many materials simulations may interact via more complicated embedded atom or many-body potentials. Since direct methods quickly overwhelm the computing capabilities of even the fastest supercomputers, many schemes have been developed to solve both long-range and short-range problems.

Despite the development of clever algorithms for reducing the complexity of MD simulations, most MD simulations have been severely limited by two main factors. The first of these is the fact that realistic simulations of materials may require tens of millions to billions of atoms. Even if a billion atom MD simulation could be performed, it would still be a "small" simulation considering the fact that a speck of dust can contain more than a billion atoms. The second problem is the time-scales involved. In most MD simulations, a single timestep may be on the order of a femtosecond, yet in order to perform a realistic experiment, it may be desirable to follow a simulation for several microseconds. As a result of these two major limitations, most MD simulations have, until recently, been limited to only a few hundred thousand atoms and a relatively small number of timesteps.

With the development of massively parallel supercomputers, there has been been considerable interest in developing fast parallel MD algorithms [10, 11]. As a result, simulation sizes have jumped to more than 100 million atoms and the time required to perform a simulation significantly reduced [12]. We will describe our efforts at Los Alamos National Laboratory to develop a fast code for performing large scale MD simulations with more than 100 million atoms on the Connection Machine 5 (CM-5) and Cray T3D massively parallel supercomputers.

In many materials simulations, it is possible to assume that the atoms only interact with other atoms that are nearby (this is due to screening effects that mask out the long range forces). A cutoff distance r_c is specified and any two atoms that are further away from each other than this distance do not interact. The short-range MD problem involves two critical aspects. First, one must develop a scheme for determining which

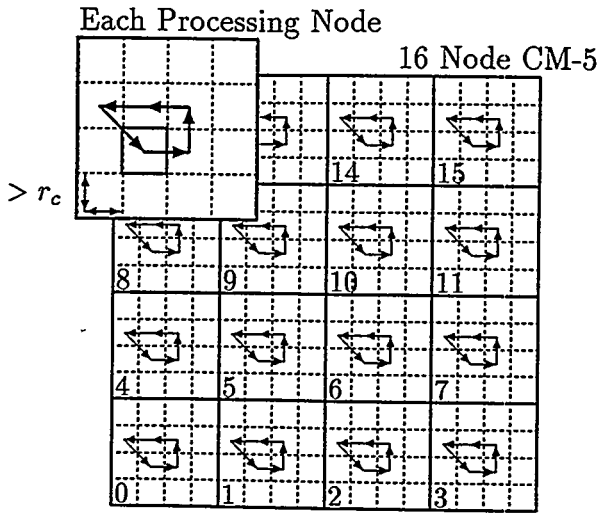

Fig. 3. Processor layout and force calculation.

atoms interact with each other. Secondly, an efficient method for calculating the forces between those atoms must be implemented.

Our algorithm is based on a cell method and has been described in detail in [10]. Here we highlight its main features in 2D. The algorithm extends naturally to 3D. The algorithm begins by dividing space into large regions that are assigned to the different processing nodes available. Each node then further subdivides its region into small cells, each with dimensions slightly larger than the cutoff distance r_c as shown in Figure 3. The atoms are placed into the proper node and subcell according to the atom's coordinates.

This structure organizes the atoms in a way that allows us to easily calculate forces. Since each cell is slightly larger than the cutoff distance r_c, only atoms in the same cell or nearest neighboring cells will contribute to the total force on each atom. To calculate the forces on the atoms in each cell, we introduce the idea of an "interaction path." The path in 2D is shown in Figure 3. The path serves two purposes. First, it specifies the order in which neighboring cells are processed in the force calculation. At each step of the path, forces between the atoms in the starting cell and the neighboring cell are calculated. By applying Newton's third law, forces are accumulated by both cells. This cuts the number of force calculations in half and allows us to consider only half of the neighboring cells (forces from a cell's lower neighbors will be calculated when those cells follow the path). When all of a cell's neighbors reside on the same processor, the path simply specifies the order in which forces are calculated between those cells. However, when neighboring cells reside on different processing nodes, the path serves to coordinate the message passing between nodes. When the path crosses a processor boundary, atom coordinates and accumulated forces are sent to a neighboring processor. This processor then calculates the forces between the received particles and its own cells. Afterwards, the processor

passes the atoms and accumulated forces to the next processor on the path. Eventually, the atoms will be sent back to the original processor along with all of the forces that were calculated. It is important to note that both positions and forces are carried along the path. For corner cells, the path may pass through as many as six different processing nodes that all require atom positions and which contribute to the total force on the atoms before they are returned to the original processor. This process of calculating forces proceeds serially on each node. The nodes run asynchronously except for boundary cells where all of the nodes participate in synchronous "send and receive" type message passing operations.

Once all of the forces have been calculated, the equations of motion are integrated using the Langevin Dynamics scheme outlined in section 2 or by using a deterministic 2nd order scheme if no coupling to external noise is desired. The atoms are then moved to new positions and the data structures updated to reflect the new positions of the atoms. This is done by checking all of the atom coordinates and moving atoms to new cells if necessary. When atoms need to change processing nodes, asynchronous message passing is used to send atoms through the network to their new destination. Afterwards, we are ready to calculate forces again and the process is repeated.

The algorithm has been implemented in a code SPaSM (Scalable Parallel Short-range Molecular dynamics). SPaSM is written almost entirely in ANSI C and uses explicit message passing for communications. To reduce portability problems, SPaSM uses a custom message passing library that we have developed. This library is then implemented in whatever native message passing environment is available. On the CM-5, we used CMMD and on the T3D we used a combination of PVM and Cray shared memory functions (a special version of the library is available that allows SPaSM to run on single processor workstations as well). In addition, on the CM-5, we have written the force calculation in CDPEAC, the assembler language for programming the vector units.

The code allows a variety of short-range potentials to be used. A typical short-range potential is given by the truncated Lennard Jones 6-12 potential

$$V(r) = \begin{cases} 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right) & 0 < r \leq r_c \\ 0 & r_c < r \end{cases} \tag{27}$$

Since the potential quickly drops to zero, we truncate it at a distance r_c. No atoms will interact beyond this point. While the LJ potential is one of the most common short-range potentials for many MD studies, our code allows any short-range pair potential to be used through a table lookup and linear interpolation scheme. In addition, more complicated potentials can be used such as embedded atom potentials which are useful for simulating metals. We refer the reader to [10, 11, 12, 13] for more details on timing, computer implementation etc.

We conclude this section by showing a snapshot from a recent MD simulation of fracture in a thin plate with 38 million atoms Figure 4. We hope that the reader is now at least somewhat convinced that Langevin and MD simulation techniques are extremely useful tool for studying nonlinear disordered problems and encourage him or her to seek more information in the literature cited.

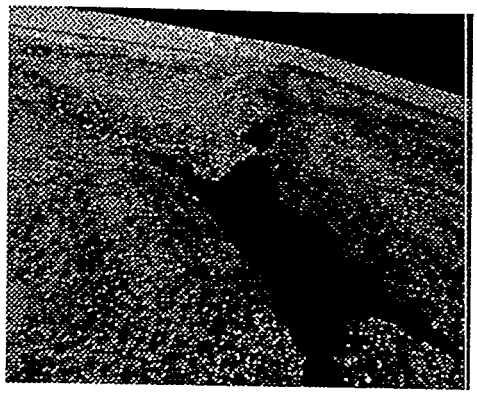

Fig.4: Fracture experiment with 38 million atoms.

ACKNOWLEDGMENTS

I would like to thank my collaborators, David Beazley, Alan Bishop, Niels Grønbech Jensen and Bill Kerr for their contributions to the work described here. I thank the Los Alamos Advanced Computing Laboratory for generous support and for making their facilities available to us. This work was performed under the auspices of the US Department of Energy.

References

[1] R. KUBO, *Statistical Mechanics*, North-Holland (1990).

[2] H. S. GREENSIDE AND E. HELFAND, *Bell Syst. Tech. J.* **60**, *1927 (1981).*

[3] F. C. FRANK AND W. T. READ, *Phys. Rev.* **79**, *722 (1950)*; W. K. BURTON, N. CABRERA, AND F. C. FRANK, *Philos. Trans. R. Soc. London* **243A**, *299 (1951)*; N. CABRERA AND M. M. LEVINE, *Philos. Mag.* **1**, *450 (1956).*

[4] J. D. WEEKS AND G. H. GILMER, *Adv. Chem. Phys.* **40**, *157 (1979).*

[5] F. FALO, A. R. BISHOP, P. S. LOMDAHL, AND B. HOROVITZ, *Phys. Rev. B***43**, *8081 (1991).*

[6] W. D. HILLIS, "The Connection Machine", *Scientific American*, (June 1987) 108.

[7] C. E. LEISERSON ET AL. in *Proc. of Symposium on Parallel and Distributed Algorithms '92*, June 1992, San Diego.

[8] P. J. HATCHER AND M. J. QUINN, *Data-Parallel Programming of MIMD Computers*, The MIT Press (1991).

[9] M. P. ALLEN AND D. J. TILDESLEY, *Computer Simulations of Liquids*, Clarendon Press, Oxford (1987).

[10] D. M. BEAZLEY AND P. S. LOMDAHL, *Parall. Comp. 20 (1994) 173-195.*

[11] P. TAMAYO, J.P. MESIROV AND B.M. BOGHOSIAN, *Proc. of Supercomputing 91*, IEEE Computer Society, 1991, pp: 462.

[12] P.S. LOMDAHL, P. TAMAYO, N. GRØNBECH-JENSEN AND D.M. BEAZLEY, *Proc. of Supercomputing 93*, IEEE Computer Society (1993), pp: 520–527.

[13] D. M. BEAZLEY, P. S. LOMDAHL, P. TAMAYO AND N. GRØNBECH-JENSEN, Proceedings of the 8th International Parallel Processing Symposium (IPPS'94), IEEE Computer Society, 1994, pp: 800–809.