ELSEVIER

Computer Physics Communications 83 (1994) 197–214

Computer Physics Communications

Multiresolution molecular dynamics algorithm for realistic materials modeling on parallel computers

Aiichiro Nakano, Rajiv K. Kalia, Priya Vashishta

Concurrent Computing Laboratory for Materials Simulations, Department of Computer Science, Department of Physics&Astronomy, Louisiana State University, Baton Rouge, LA 70803-4001, USA

Received 21 April 1994; revised 1 August 1994

Abstract

For realistic modeling of materials, a molecular-dynamics (MD) algorithm is developed based on multiresolutions in both space and time. Materials of interest are characterized by the long-range Coulomb, steric and charge-dipole interactions as well as three-body covalent potentials. The long-range Coulomb interaction is computed with the fast multipole method. For bulk systems with periodic boundary conditions, infinite summation over repeated image charges is carried out with the reduced cell multipole method. Short- and medium-range non-Coulombic interactions are computed with the multiple time-step approach. A separable tensor decomposition scheme is used to compute three-body potentials. For a 4.2 million-particle SiO_2 system, one MD step takes only 4.8 seconds on the 512-node Intel Touchstone Delta machine and 10.3 seconds on 64 nodes of an IBM SP1 system. The constant-grain parallel efficiency of the program is $\eta' = 0.92$ and the communication overhead is 8% on the Delta machine. On the SP1 system, $\eta' = 0.91$ and communication overhead is 7%.

1. Introduction

Molecular dynamics (MD) simulation is rapidly becoming an integral part of computer-aided materials design. Design of materials like nanophase metals and ceramics requires large-scale MD simulations. These materials are synthesized from atom clusters of nanometer (10^{-9} m) size [1]. Remarkable improvements in strength and flexibility have been reported for nanophase metals and ceramics. MD simulations can provide a microscopic understanding of the structure-property relationship in nanophase materials.

Since cluster-assembled nanophase materials are composed of large-scale structural units, each consisting of 10^3 to 10^5 particles, the simulation system must comprise at least 10^6 to 10^7 particles. To prepare a well-thermalized MD configuration and calculate the mechanical and thermal properties, it is necessary to carry out multimillion-particle simulations for 10^5 to 10^7 time steps. In order for simulations to be completed within a reasonable time, it is therefore crucial to achieve a speed of a few seconds per step.

For systems interacting via simple, short-ranged binary potentials, it is possible to simulate *multimil-*

SSDI 0010-4655(94)00108-1

lion particles at a speed of a *few seconds / step* using recent advances in numerical techniques and parallel computing technology [2]. However, a vast majority of materials require high-quality interatomic potentials for reliable atomistic simulations. For example, interatomic potentials for many ceramics must include the effects of long-range Coulomb interaction, charge-dipole interaction, steric hindrance, and bending and stretching of covalent bonds [3]. Each of these potentials requires enormous computing and thus a special algorithm.

The most prohibitive computational problem is associated with the Coulomb potential. Because of its long range, each atom interacts with all the other atoms in the system. Therefore the evaluation of the Coulomb potential for an N-particle system requires $O(N^2)$ operations, which makes large-scale MD simulations difficult. Recent hierarchical algorithms [4–6] have revolutionized the computation of the Coulomb potential. The fast multipole method (FMM) uses the truncated multipole expansion and local Taylor expansion for the Coulomb potential field [6]. By computing both expansions recursively on a hierarchy of cells, the Coulomb potential is computed with O(N) operations.

In many materials simulations, periodic boundary conditions are used to minimize surface effects. The summation over infinitely repeated image charges must be carried out to compute the Coulomb potential. The conditionally convergent interaction is usually calculated by the Ewald method [7]. Though the Ewald algorithm scales practically as $O(N)$ for moderate system size [7], the computation becomes prohibitive for larger systems. Ding, Karasawa, and Goddard have developed the reduced cell multipole method (RCMM) which makes the computation of the Coulomb potential feasible for multimillion-particle systems with periodic boundary conditions [8]. In RCMM, distant images of multimillion particles are replaced by a small number of fictitious particles with the same leading multipoles as the original system. With little computational effort, the Ewald summation is applied to these reduced images.

Another computational difficulty is caused by the non-Coulombic part of the interatomic potential. In most materials, the potential in the medium range (up to the cut-off length, $r_c = 5 - 10$ Å) is a complicated function of the interatomic distance, r, so that the multipole expansion is not applicable. The finest cell in the FMM must be larger than r_c, and within the nearest-neighbor cells the interaction must be calculated directly without multipoles. With $r_c = 5 - 10$ Å, the direct force calculation for each atom involves hundreds of other atoms. The $O(10^2 N)$ pair computation dominates the computation time, considering the complicated form of the potential at $r < r_c$. We employ the multiple time-step (MTS) approach [9] to reduce the computation in this spatial regime. The MTS method is based on the fact that the farther the distance between particles the slower is the time variation of forces. Therefore different time steps are used to compute forces for different interparticle separations. The MTS algorithm typically achieves a 5- to 7-fold speedup [9]. On parallel computers, the communication overhead is significantly reduced by the MTS method [10,11].

The covalent bond-bending and stretching forces are described in the three-body potentials [3]. However, triple sums in three-body potentials make the computation inefficient. The computation can be made efficient by decomposing three-body interactions into separable tensor components [11,12]. The resulting expression involves only double sums, which can be evaluated efficiently with a speedup of a factor of two [11].

We have developed a highly efficient MD algorithm based on multiresolutions in both space and time. The long-range Coulomb potential for periodic systems are computed with the RCMM and FMM, while the medium-range non-Coulombic potentials are computed by the MTS method. The three-body interactions are calculated with the separable tensor decomposition. The performance of the multiresolution molecular dynamics (MRMD) algorithm is tested on the 512-node Intel Touchstone Delta machine at Caltech and the 128-node IBM SP1 system at Argonne National Laboratory.

This paper is organized as follows. In Section 2, we describe the key components of the MRMD algorithm. The implementation of the MRMD algorithm on parallel computers is discussed in Section 3. The results of numerical tests are given in Section 4, and Section 5 contains the conclusion.

2. Multiresolution molecular dynamics algorithm

2.1. Variable-shape molecular dynamics

In MD simulations, a physical system consisting of N atoms is represented by a set of atomic coordinates, $\{\boldsymbol{r}_i \mid i = 1 \ldots N\}$. With periodic boundary condition, there are identical particles at $\boldsymbol{r}_i + \nu$, where $\nu = l\boldsymbol{a} + m\boldsymbol{b} + n\boldsymbol{c}$ is a translation vector (l, m, and n are integers). $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ represent the MD box, see Fig. 1. We are concerned with systems for which the potential energy is a sum of two- and three-body potentials,

$$V = \tfrac{1}{2} \sum_{\nu} \sum_{i=1}^{N} {\sum_{j=1}^{N}}' v_{ij}^{(2)}(r_{ij}) + \sum_{i,j<k} v_{jik}^{(3)}(\boldsymbol{r}_{ij}, \boldsymbol{r}_{ik}), \tag{1}$$

where $r_{ij} = |\boldsymbol{r}_{ij}|$, and $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$. In Eq. (1) the prime on the summation over the two-body potential, $v_{ij}^{(2)}(r_{ij})$, indicates the omission of the $i = j$ term when $\nu = 0$. The three-body potential, $v_{jik}^{(3)}(\boldsymbol{r}_{ij}, \boldsymbol{r}_{ik})$, is short-ranged and the minimum-image convention is used to evaluate this term [13].

In this paper, we use the constant-pressure MD scheme by Parrinello and Rahman [14]. In this scheme, the shape and volume of the simulation box (i.e., the vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$) change dynamically. Introducing dimensionless atomic coordinates, $\boldsymbol{s}_i$, through the relation, $\boldsymbol{r}_i = \boldsymbol{h}\boldsymbol{s}_i$, where $\boldsymbol{h} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is the 3×3 translation matrix, the equation of motion for $\boldsymbol{s}_i$ is [14],

$$\ddot{\boldsymbol{s}}_i = -\boldsymbol{G}^{-1}\dot{\boldsymbol{G}}\dot{\boldsymbol{s}}_i + m_i^{-1}\boldsymbol{h}^{-1}\boldsymbol{F}_i, \tag{2}$$

where m_{i} is the mass of the ith particle; $\boldsymbol{F}_i = -\partial V/\partial \boldsymbol{r}_i$ is the force acting on the ith particle; and $\boldsymbol{G} = \boldsymbol{h}^T\boldsymbol{h}$ is the metric tensor with $\boldsymbol{h}^T$ being the transpose of matrix $\boldsymbol{h}$. In Eq. (2), time derivatives are denoted by dots above the variables.

The equation of motion for $\boldsymbol{h}$ is derived through a fictitious Lagrangian formalism [14]:

$$W\ddot{\boldsymbol{h}} = (\boldsymbol{\pi} - P\boldsymbol{I})\Omega(\boldsymbol{h}^T)^{-1}, \tag{3}$$

where W is a fictitious mass associated with the simulation box, P is the hydrostatic pressure, and $\boldsymbol{I}$ is the unit matrix. In Eq. (3) the volume of the system, $\Omega = \det(\boldsymbol{h})$, and the microscopic stress tensor, $\boldsymbol{\pi}$, is calculated from

$$\Omega\boldsymbol{\pi} = \sum_{i=1}^{N} m_i \dot{\boldsymbol{r}}_i'(\dot{\boldsymbol{r}}_i')^T - \tfrac{1}{2} \sum_{\upsilon} \sum_{i=1}^{N} {\sum_{j=1}^{N}}' \boldsymbol{r}_{ij} \left(r_{ij}^{-1} \frac{\mathrm{d}v_{ij}^{(2)}}{\mathrm{d}r_{ij}} \right) \boldsymbol{r}_{ij}^T - \sum_{i,j<k} \left(\boldsymbol{r}_{ij} \frac{\partial v_{jik}^{(3)}}{\partial \boldsymbol{r}_{ij}} + \boldsymbol{r}_{ik} \frac{\partial v_{jik}^{(3)}}{\partial \boldsymbol{r}_{ik}} \right), \tag{4}$$

where $\dot{\boldsymbol{r}}_i' = \boldsymbol{h}\dot{\boldsymbol{s}}_i$.

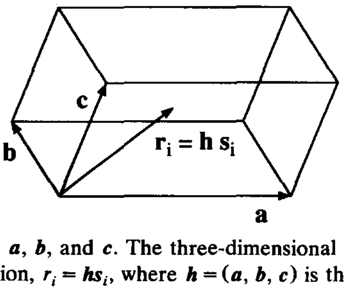

Fig. 1. An MD box is formed by three vectors, $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$. The three-dimensional coordinate, $\boldsymbol{r}_i$, of the ith atom is related to its dimensionless coordinate, $\boldsymbol{s}_i$, through the relation, $\boldsymbol{r}_i = \boldsymbol{h}\boldsymbol{s}_i$, where $\boldsymbol{h} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is the translation matrix.

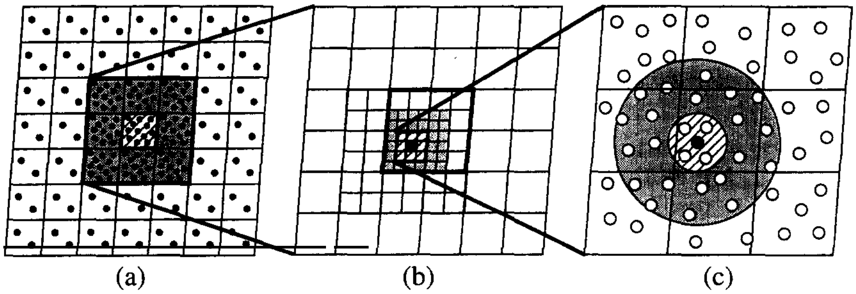

Fig. 2. Schematic representation of spatial multiresolution for a two-dimensional system. (a) Periodically repeated images of the original MD box (hatched). Replacing each well-separated image by a small number of particles with the same leading multipole expansions reduces the computation enormously while maintaining the necessary accuracy. The nearest-neighbor images (shaded) cannot be replaced by the reduced image. (b) A hierarchy of cells in the FMM. The nearest-neighbor cells of the filled cell consist of itself and the hatched cells. The interactive cells are shaded. (c) The direct forces on a particle (solid circle) are due to the primary (open circles within the hatched area), secondary (open circles within the shaded area), and tertiary (the other open circles) neighbor atoms.

The equations of motion, (2) and (3), are integrated with Gear's fifth-order predictor-corrector method [15] using a time increment Δt in Eq. (2) and $n_1 \Delta t (n_1$ is typically 15) in Eq. (3). The most time-consuming part of MD simulations is the calculation of the potential energy, V, interatomic forces, $\boldsymbol{F}_i$, and the microscopic stress tensor, $\boldsymbol{\pi}$. The MRMD algorithms have been designed to efficiently compute the contributions to these quantities from various spatial regimes. These algorithms are described in the following subsections.

2.2. Fast multipole method and reduced cell multipole method

In the MRMD algorithm, the Coulomb interaction is calculated with the fast multipole method (FMM, see Ref. [6]) and the reduced cell multipole method (RCMM, see Ref. [8]). Both methods use the multipole expansion of the Coulomb potential. In the FMM, the Coulomb potential is computed using a hierarchy of cells. The hierarchy is defined by recursively decomposing the MD box according to a tree structure in which a node corresponds to a cell. The root of the tree is at level 0 and it corresponds to the MD box shown in Fig. 1. With periodic boundary conditions, the root cell is regarded as a unit cell whose images are repeated infinitely as shown in Fig. 2(a). At level l, a parent cell is decomposed into $2 \times 2 \times 2$ children cells of equal volume at level $l+1$, see Fig. 2b. The number of cells at level l is thus 8^l per unit cell. The recursive decomposition stops at the leaf level, $l = L$, where further decomposition would cause children cells to have parallel faces with distance less than $\mathrm{r_c}$, see Fig. 2c. We determine r_c such that the interatomic potential is purely Coulombic for $r > r_c$. Therefore any pair of particles interact via the Coulomb potential when they reside in two different leaf cells which do not share any corners.

For the cth cell at level l, the nearest-neighbor cells are defined as the cell c itself and any other cells at the same level with which it shares a corner. At every level l, there are 27 nearest-neighbor cells for each cell. All the other cells at the same level are said to be well-separated from the cth cell. The interactive cells are defined to be well-separated from the cth cell and to be children of the nearest-neighbor cells of c's parent, see Fig. 3.

The FMM computes the multipole expansions, $\Phi_{l,c}$, and the local Taylor expansions, $\Psi_{l,c}$, of the Coulomb potential field recursively on the hierarchy of cells. For charges Z_i distributed at positions $\boldsymbol{r}_i$ in

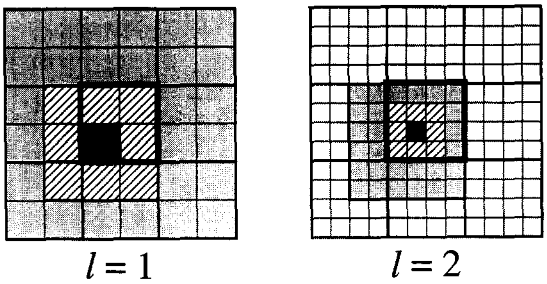

Fig. 3. Cells at tree levels 1 and 2 in a two-dimensional system. The nearest-neighbor cells of the filled cell consist of itself and the hatched cells. All the other cells are well-separated. The shaded cells represent the interactive cells. The unit cell is bounded by thick lines.

the cth cell at level l, the multipole expansion, $\Phi_{l,c} = (Z, \mu_\alpha, Q_{\alpha\beta}, \dots)$, centered at the center of the cell is expressed as,

$$\phi(\boldsymbol{r}) = \frac{Z}{r} + \frac{\mu_\alpha r_\alpha}{r^3} + \frac{Q_{\alpha\beta} r_\alpha r_\beta}{r^5} + \cdots, \tag{5}$$

where $\phi(\boldsymbol{r})$ is the Coulomb potential at position $\boldsymbol{r}$ relative to the center of the cell, $\alpha = x, y, z$, and there is implicit summation over repeated indices. The lower order moments are the charge – $Z = \Sigma_i Z_i$; dipoles – $\mu_\alpha = \Sigma_i Z_i r_{i\alpha}$; and quadrupoles – $Q_{\alpha\beta} = \Sigma_i Z_i (3 r_{i\alpha} r_{i\beta} - \delta_{\alpha\beta} r_i^2)/2$.

We define a partial contribution, $\psi_{l,c}(\boldsymbol{r})$, to the Coulomb potential field in the cth cell at level l due to all the particles in the well-separated cells,

$$\psi_{l,c}(\boldsymbol{r}) = \sum_{c' \notin \{nn(c)\}} \sum_{j \in c'} \frac{Z_j}{|\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{R}_{c'c}|}, \tag{6}$$

where a set of the nearest-neighbor cells of the cth cell is denoted by $\{nn(c)\}$, and $\boldsymbol{R}_{c'c}$ is a vector from the center of the cth cell to that of the c'th cell. The variation of $\psi_{l,c}(\boldsymbol{r})$ is small since it is produced by well-separated charges. Therefore we can expand the field in Taylor series about the center of the cell. Denoting by $\Psi_{l,c} = (\Psi^{(0)}_{l,c}, \Psi^{(1)}_{l,c;\alpha}, \Psi^{(2)}_{l,c;\alpha\beta}, \dots)$, we have

$$\psi_{l,c}(\boldsymbol{r}) = \Psi^{(0)}_{l,c} + \Psi^{(1)}_{l,c;\alpha} r_\alpha + \Psi^{(2)}_{l,c;\alpha\beta} r_\alpha r_\beta + \cdots, \tag{7}$$

where $\boldsymbol{r}$ is the position relative to the center of the cell.

Using these definitions, the total potential energy, Eq. (1), is rewritten as,

$$V = \tfrac{1}{2} \sum_{i=1}^{N} Z_i \psi_{L,c(i)}(\boldsymbol{r}_i) + \tfrac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j \in \{nn(c(i))\} \\ j \neq i}} v^{(2)}_{ij}(r_{ij}) + \sum_{i,j<k} v^{(3)}_{jik}(\boldsymbol{r}_{ij}, \boldsymbol{r}_{ik}), \tag{8}$$

where $c(i)$ denotes the finest cell at level L to which the ith atom belong. The first term of Eq. (8) represents the long-range Coulomb potential between well-separated leaf cells, and the second term is the two-body potential within the nearest-neighbor cells. The first term is called the far-field contribution to the potential energy, while the second and third terms constitute the near-field contribution.

Similarly the microscopic stress tensor, Eq. (4), is rewritten as

$$\Omega\boldsymbol{\pi} = \sum_{i=1}^{N} m_i \dot{\boldsymbol{r}}_i'(\dot{\boldsymbol{r}}_i')^T + \tfrac{1}{2}\sum_{i=1}^{N} Z_i \boldsymbol{\pi}_{L,c(i)}(\boldsymbol{r}_i) - \tfrac{1}{2}\sum_{i=1}^{N} \sum_{\substack{j\in\{nn(c(i))\} \\ j\neq i}} \boldsymbol{r}_{ij}\left(r_{ij}^{-1}\frac{\mathrm{d}v_{ij}^{(2)}}{\mathrm{d}r_{ij}} \right)\boldsymbol{r}_{ij}^T$$

$$- \sum_{i,j<k} \left(\boldsymbol{r}_{ij}\frac{\partial v_{jik}^{(3)}}{\partial \boldsymbol{r}_{ij}} + \boldsymbol{r}_{ik}\frac{\partial v_{jik}^{(3)}}{\partial \boldsymbol{r}_{ik}} \right), \tag{9}$$

where $\boldsymbol{\pi}_{l,c}(\boldsymbol{r})$ is a partial contribution to the stress-tensor field at position $\boldsymbol{r}$ relative to the center of the cth cell at level l due to all the particles in well-separated cells:

$$\boldsymbol{\pi}_{l,c}(\boldsymbol{r}) = \sum_{c'\notin\{nn(c)\}} \sum_{j\in c'} Z_j \frac{(\boldsymbol{r}_j\boldsymbol{r}_j - \boldsymbol{R}_{c'c})(\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{R}_{c'c})^T}{|\boldsymbol{r} - \boldsymbol{r}_j\boldsymbol{R}_{c'c}|^3}, \tag{10}$$

The local Taylor expansion, $\Pi_{l,c} = (\Pi_{l,c;\alpha\beta}^{(0)}, \Pi_{l,c;\alpha\beta\xi}^{(1)}, \Pi_{l,c;\alpha\beta\xi\eta}^{(2)}, \ldots)$, of the partial stress-tensor field, $\boldsymbol{\pi}_{l,c}(\boldsymbol{r})$, is also defined through the relation,

$$\boldsymbol{\pi}_{l,c;\alpha\beta}(\boldsymbol{r}) = \Pi_{l,c;\alpha\beta}^{(0)} + \Pi_{l,c;\alpha\beta\xi}^{(1)} r_\xi + \Pi_{l,c;\alpha\beta\xi\eta}^{(2)} r_\xi r_\eta + \cdots. \tag{11}$$

The local expansions, $\Psi_{L,c}$ and $\Pi_{L,c}$, include the contributions from all the particles in well-separated cells at the leaf level. The straightforward computation of these terms requires $O(N^2)$ operations. The FMM reduces the complexity to $O(N)$ by computing multipoles and local expansions recursively on the hierarchy of cells. This is implemented most conveniently by using five translation operators, $T_{1,\boldsymbol{R}}$, $T_{2,\boldsymbol{R}}$, $T_{3,\boldsymbol{R}}$; $S_{2,\boldsymbol{R}}$, and $S_{3,\boldsymbol{R}}$, defined below [16].

The first operator, $T_{1,\boldsymbol{R}}$, shifts the origin of a multipole expansion. It operates on a multipole expansion centered at $\boldsymbol{R}$, and gives the multipole expansion centered at the origin due to the same charge distribution. By denoting $\Phi' = (Z', \mu'_\alpha, Q'_{\alpha\beta}, \ldots) = T_{1,\boldsymbol{R}}\ (\Phi)$ where $\Phi = (Z, \mu_\alpha, Q_{\alpha\beta}, \ldots)$, the first few relations involved in the translation are written as [17] $Z' = Z$ and $\mu'_\alpha = \mu_\alpha + ZR_\alpha$.

The second class of operators, $T_{2,\boldsymbol{R}}$ and $S_{2,\boldsymbol{R}}$, act on a multipole expansion, Φ, centered at $\boldsymbol{R}$, and give the local Taylor expansions of the fields at the origin produced by the multipoles; $T_{2,\boldsymbol{R}}$ and $S_{2,\boldsymbol{R}}$ give the local expansions for the Coulomb potential, Ψ, and the microscopic stress tensor, Π, respectively. By denoting $\Psi = (\Psi^{(0)}, \Psi_\alpha^{(1)}, \Psi_{\alpha\beta}^{(2)}, \cdots) = T_{2,\boldsymbol{R}},(\Phi)$, the first relation due to the translation is written as [17]

$$\Psi^{(0)} = \frac{Z}{R} + \frac{\mu_\xi R_\xi}{R^3} + \frac{Q_{\xi\eta} R_\xi R_\eta}{R^5} + \cdots. \tag{12}$$

Similarly $\Pi = (\Pi_{\alpha\beta}^{(0)}, \Pi_{\alpha\beta\xi}^{(1)}, \Pi_{\alpha\beta\xi\eta}^{(2)}, \cdots) = S_{2,\boldsymbol{R}}(\Phi)$, where the first relation is

$$\Pi_{\alpha\beta}^{(0)} = R_\alpha R_\beta\left(\frac{Z}{R^3} - \frac{3P_{\eta\eta}}{2R^5} \right) + \frac{P_{\alpha\beta}}{R^3} + \frac{\mu_\alpha R_\beta + \mu_\beta R_\alpha}{R^3} - 3\mu_\eta R_\eta \frac{R_\alpha R_\beta}{R^5 -} - \frac{3}{R^5}(P_{\alpha\eta}R_\eta R_\beta + P_{\beta\eta}R_\eta R_\alpha)$$

$$+ \tfrac{15}{2} P_{\xi\eta} R_\xi R_\eta \frac{R_\alpha R_\beta}{R^7} \cdots. \tag{13}$$

In Eq. (13) $P_{\alpha\beta} = \Sigma_i Z_i r_{i\alpha} r_{i\beta}$ is related to the quadrupole, $Q_{\alpha\beta} = (3P_{\alpha\beta} - P_{\gamma\gamma}\delta_{\alpha\beta})/2$.

In the FMM algorithm, the tree of cells is traversed twice. First, starting from the leaf level, $l = L$, the tree is traversed upward to compute the multipoles, $\Phi_{l,c}$, for all the cells at all the levels. The local Taylor expansions, $\Psi_{l,c}$, and $\Pi_{l,c}$, are then computed from $\Phi_{l,c}$ for all the cells at all the levels, starting at the root level $l = 0$ and traversing the tree downward.

The computation of far-image interactions in the multiresolution molecular dynamics (MRMD) algorithm starts by computing multipoles, $\Phi_{L,c}$, for all the finest cells at the leaf level. Next the

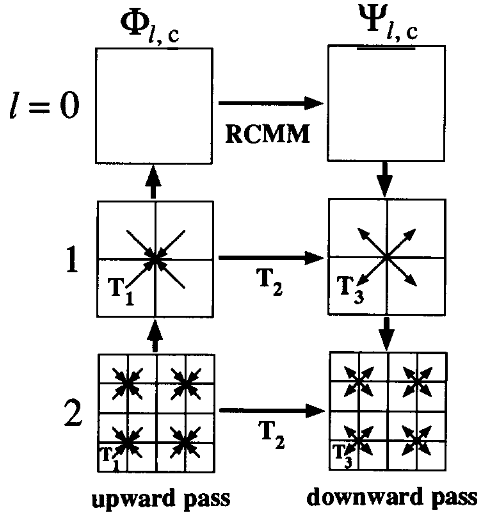

Fig. 4. Schematic of the far-field computation in a two dimensional system. (Left column) Starting from the leaf level, the tree of cells is traversed upward to compute the multipoles for all the cells at all the levels. The multipole-to-multipole operator, T_1, is used to shift the multipoles of children cells. The multipole at the root level is used to compute the local Taylor expansion for the Coulomb potential at level 0 based on the RCMM. (Right column) The tree is traversed downward to compute the local expansions for all the cells at all the levels. The multipole-to-local translation operator, T_2, is used to compute the partial contribution to the local expansion due to the interactive cells. The local-to-local translation operator, T_3, is used to shift the local expansion of the parent cell.

multipoles at coarser levels are calculated by combining the multipoles at deeper levels, see Fig. 4. At level l, $\Phi_{l,c}$ is calculated by shifting and adding up the multipoles of its 8 children cells, $\Phi_{l+1,c'}$, by using the multipole-to-multipole translation operator, $T_{1,\boldsymbol{R}}$:

$$\Phi_{l,c} = \sum_{c' \in \{\text{child}(c)\}} T_{1,\boldsymbol{R}_{c'c}}(\Phi_{l+1,c'}), \tag{14}$$

where {child(c)} is the set of the 8 children cells of the cth cell. Eq. (14) is used to traverse the tree upward, $l = L, L-1, \dots, 1, 0$, and compute the multipoles, $\Phi_{l,c}$, for all the cells at all the levels.

At the next stage of the algorithm, the multipoles at the root level are used to compute the local expansions at level 0. A difficulty arises in calculating $\Psi_{0,0}$ and $\Pi_{0,0}$ which involve the contributions from infinitely repeated image charges that are well-separated from the original MD box. We use the RCMM to compute these contributions to the local expansions [8]. The main idea of the RCMM is that even though the unit cell contains multimillion particles, the force from a well-separated image is described accurately by the first few terms of the multipole expansion. Then each well-separated images can be replaced by a small number of particles, see Fig. 2a. To reproduce the multipole moments up to order K, only $M = (K+1)(K+2)(K+3)/6$ particles are needed [8].

Given the lowest K multipoles of the original N-particle system, $\Phi_{0,0}$, each of the $\infty - 27$ well-separated images is replaced by randomly distributed M charges whose lowest K multipoles equal the corresponding multipoles of the original system. This requires solving a set of M linear equations [8].

The conditionally-convergent infinite summation over these reduced images is calculated by the conventional Ewald method [7]. The Coulomb potential field from the well-separated images is thus separated into rapidly convergent sums in Fourier and real spaces as [8]

$$\psi_{0,0}(\boldsymbol{r}) = \frac{1}{\Omega} \sum_{\boldsymbol{k}}{}' \left(\frac{4\pi}{k^2} \mathrm{e}^{-\gamma^2 k^2} \rho(\boldsymbol{k}) \right) \mathrm{e}^{i\boldsymbol{k}\cdot\boldsymbol{r}} - \sum_{\boldsymbol{\nu} \in \{nn\}} \sum_{j=1}^{M} \frac{Z_j}{|\boldsymbol{r} - \boldsymbol{r}_j - \nu|} \operatorname{erf}\left(\frac{|\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{\nu}|}{2\gamma} \right), \tag{15}$$

where $\boldsymbol{k} = (2\pi/\Omega)(l\boldsymbol{b} \times \boldsymbol{c} + m\boldsymbol{c} \times \boldsymbol{a} + n\boldsymbol{a} \times \boldsymbol{b})$ is a wave vector (l, m, and n are integers); Σ' means that $\boldsymbol{k} = 0$ is excluded from the summation; and $\{nn\}$ denotes a set of translation vectors which connect the center of the original MD box to those of the nearest-neighbor images. In Eq. (15) the charge density, $\rho(\boldsymbol{k})$, is calculated as

$$\rho(\boldsymbol{k}) = \sum_{j=1}^{M} Z_j \, \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}_j}, \tag{16}$$

and erf(z) is the error function. The convergence factor, γ, is chosen to be $\Omega^{1/3}/10$. We also restrict the $\boldsymbol{k}$ sum in Eq. (15) to $|\boldsymbol{k}| < 10\pi/\Omega^{1/3}$. To derive Eq. (15), it must be noted that $\psi_{0,0}(\boldsymbol{r})$ only includes the contribution from well-separated images but not from the nearest-neighbor images. The ordinary Ewald procedure using the Fourier transform must first be applied to the Coulomb potential which also includes the nearest-neighbor contribution. The nearest-neighbor contribution is then subtracted in the real-space sum [8].

The same procedure is applied to the calculation of the stress-tensor field, $\boldsymbol{\pi}_{0,0}(\boldsymbol{r})$,

$$\begin{aligned}\boldsymbol{\pi}_{0,0}(\boldsymbol{r}) = {} & \frac{1}{\Omega} \sum_{j=1}^{M} \sum_{\boldsymbol{k}}{}' \left[\boldsymbol{I} - \frac{2\boldsymbol{k}\boldsymbol{k}^T}{k^2}(1 + \gamma^2 k^2) \right] \frac{4\pi Z_j}{k^2} \exp\left[-\gamma^2 k^2 + \mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{r} - \boldsymbol{r}_j) \right] \\ & - \sum_{j=1}^{M} \sum_{\boldsymbol{\nu}\in\{nn\}} Z_j (\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{\nu})(\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{\nu})^T \\ & \times \left(\frac{1}{|\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{\nu}|^3} \mathrm{erfc}\left(\frac{|\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{\nu}|}{2\gamma} \right) - \frac{1}{\sqrt{\pi}\gamma |\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{\nu}|^2} \exp\left[-|\boldsymbol{r} - \boldsymbol{r}_j - \boldsymbol{\nu}|^2/4\gamma^2 \right] \right),\end{aligned} \tag{17}$$

The local expansion coefficients, $\Psi_{0,0}$ and $\Pi_{0,0}$, are obtained by comparing the Taylor expansion of Eqs. (15) and (17) with respect to $\boldsymbol{r}$ with Eqs. (7) and (11). In this approach, the local expansion coefficients need to be evaluated only once for the entire system, instead of evaluating the complicated expressions, Eqs. (15) and (17), at each particle's position, $\boldsymbol{r}_i$, millions of times [8]. Also the Ewald summation for a M (a few tens) particle systems requires little computational effort (see the timing results in Section 4).

We have thus obtained the local expansions at the root level. At the next stage of the algorithm, starting from $\Psi_{0,0}$ and $\Pi_{0,0}$ at level 0, the local expansions at deeper levels are computed recursively by traversing the tree downward, $l = 0, 1, \ldots, L-1, L$, see Fig. 4. Suppose that we have calculated the local expansions at level $l-1$. The contributions for the local expansion, $\Psi_{l,c}$, are classified into two parts: (i) One due to all the particles in the parent's well-separated cells, and (ii) the other from particles within the interactive set (Fig. 3 shows the interactive sets for cells at level 1 and 2 in two dimensions). The set of particles which contributes to (i) is the same as the one which contributes to the parent's local expansion, $\Psi_{l-1,\mathrm{parent}(c)}$. (Parent($c$) denotes the parent cell of the cth cell.) Therefore, the contribution (i) is obtained by simply shifting the origin of $\Psi_{l-1,\mathrm{parent}(c)}$ by the local-to-local transformation operator, $T_{3,\boldsymbol{R}}$. On the other hand, the contributions from the interactive set must be calculated from the multipoles, $\Phi_{l,c'}$, of the interactive cells by using the multipole-to-local transformation operator, $T_{2,\boldsymbol{R}}$. In summary,

$$\Psi_{l,c} = \sum_{c'\in\{\mathrm{interactive}(c)\}} T_{2,\boldsymbol{R}_{c'c}}(\Phi_{l,c'}) + T_{3,\boldsymbol{R}}(\Psi_{l-1,\mathrm{parent}(c)}), \tag{18}$$

where {interactive(c)} denotes the set of interactive cells of the cth cell, and $\boldsymbol{R}$ is the vector from the center of the cth cell to the center of the parent cell.

Similarly the local expansions of the stress-tensor field is calculated as

$$\Pi_{l,c} = \sum_{c' \in \{\text{interactive}(c)\}} S_{2,\boldsymbol{R}_{c'c}}(\Phi_{l,c'}) + S_{3,\boldsymbol{R}}(\Pi_{l-1,\text{parent}(c)}). \tag{19}$$

After calculating the local expansions for all the cells at the leaf-level, L, $\Psi_{L,c}$ and $\Pi_{L,c}$ represent the Coulomb potential field and the stress-tensor field due to all the particles (including the infinite sums from repeated images) which are well-separated from the cth cell. The far-field interactions – the first term in Eq. (8) and the second term in Eq. (9) – are computed by evaluating these local expansions at the particle positions. The corresponding contribution to the force on the ith particle is obtained by differentiating the first term of Eq. (8) with respect to $\boldsymbol{r}_i$.

2.3. Multiple time-step method

The algorithm in the last subsection completes the computation of the far-field contributions to the potential energy, interatomic forces, and microscopic stress tensor. We now compute the near-field interactions among particles in the nearest-neighbor leaf cells. These interactions are not Coulombic, so they must be computed directly without using multipoles. We use the MTS method to compute the interactions in this regime [9–11].

The MTS method is based on the fact that the longer the distance the slower is the time variation of interparticle interaction. Accordingly, forces from different spatial regimes are decomposed into different classes. In our implementation, the force, $\boldsymbol{F}_i$, acting on the ith particle is divided into three parts: The primary force, $\boldsymbol{F}_{i,\text{p}}$, arises from the interaction with other particles in a sphere of radius, r_a, around the ith particle. The primary cut-off length, r_a, is chosen so that only the bonding atoms contribute to $\boldsymbol{F}_{i,\text{p}}$. Also, we choose $r_\text{a} > r_0$ so that three-body forces are entirely primary forces [11]. The secondary force, $\boldsymbol{F}_{i,\text{s}}$, consists of pairwise interactions in the range $r_\text{a} < r_{ij} < r_\text{c}$. The secondary cut-off length, r_c, is chosen such that beyond this length, the interatomic potential is purely Coulombic. The tertiary force, $\boldsymbol{F}_{i,t}$, arises from particles beyond these distances $r_{ij} > r_\text{c}$ but within the nearest-neighbor cells. Fig. 2c shows atoms which are responsible for primary, secondary, and tertiary forces on a given atom in a two-dimensional system. In our MD simulations of SiO_2 systems, we use $r_\text{a} = 2.6$ Å and $r_\text{c} = 5.5$ Å. Then for the normal-density SiO_2 glass (2.2 g/cm^3), the primary, secondary and tertiary forces on each atom are typically due to a few atoms, 50 atoms, and 300 atoms, respectively.

The secondary force varies more slowly than the primary force, and the tertiary force varies even more slowly than the secondary force. We have analyzed the power spectrum of force fluctuations resolved into different distances, and have observed a superlinear increase of the characteristic time scale with distance [11].

In the MTS method, $\boldsymbol{F}_{i,\text{p}}$ is calculated at every MD step. On the other hand, $\boldsymbol{F}_{i,s}$ and $\boldsymbol{F}_{i,t}$ are calculated at intervals of $n_1\Delta t$ and $n_2\Delta t$, respectively. Typical values we choose for n_1 and n_2 are 15 and 120. In between $\boldsymbol{F}_{i,s}$ and $\boldsymbol{F}_{i,t}$ are calculated from the Taylor series,

$$\boldsymbol{F}_{i,\kappa}(t + J\Delta t) = \sum_{\lambda=0}^{\lambda_{\max}} \frac{(J\Delta t)^\lambda}{\lambda!} \boldsymbol{F}_{i,\kappa}^{(\lambda)}(t), \tag{20}$$

where $\kappa = s$ or t, and $\boldsymbol{F}_{i,\kappa}^{(\lambda)}$ is the λth time derivative of $\boldsymbol{F}_{i,\kappa}$. With the MTS, we typically achieve an order of magnitude speed up of the direct-force calculation [9–11]. Note that this method is adaptive within the length scale r_c because it is based on concentric spheres attached to each atom or a neighbor list for each atom.

The MTS algorithm proceeds as follows. At every n_1 steps (*steps* mod $n_1 = 1$, where *steps* is the time step), two-body forces and the time derivatives of the secondary forces are first calculated using the

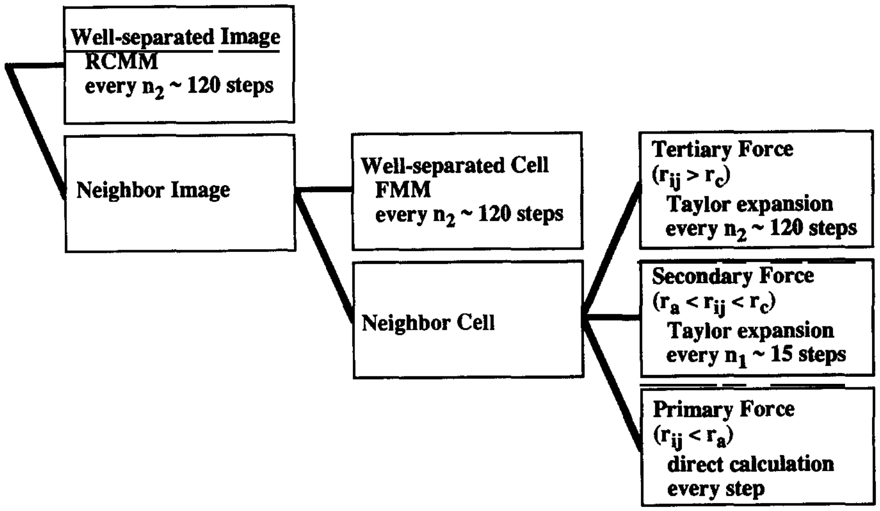

Fig. 5. Schematic of the MTS method used in the MRMD algorithm. The Coulomb potential from the well-separated images is computed with the RCMM at every n_2 steps. The far-field potential within the nearest-neighbor images is computed with the FMM at every n_2 steps. The near-field forces are classified into the primary, secondary and tertiary forces according to the interatomic distances. They are updated at every one, n_1, and n_2 steps, respectively.

linked-cell-list method [7,10,11,13]. A list of particles residing in each finest cell is constructed with the linked-list method. Direct forces on particles are then calculated using the linked list. The time derivatives of the secondary forces are stored in an array, $F_s^{(\lambda)}$, to compute these forces at subsequent steps using Eq. (20). If in addition *steps* mod $n_2 = 1$, the time derivatives of the tertiary forces stored in an array, $F_t^{(\lambda)}$, are also updated. At the same time, a list of primary pair particles is constructed for each atom.

At subsequent n_1 steps, instead of the above procedure, primary forces are calculated using the neighbor list. On the other hand, the secondary- and tertiary-forces are calculated using the stored secondary- and tertiary-force derivatives, $F_s^{(\lambda)}$ and $F_t^{(\lambda)}$, according to the Taylor expansion, Eq. (20). The three-body forces are always calculated using the pair list.

In addition to the direct-force calculation, the different time scales are also applied to the calculation of the far-field interactions. The far-field forces vary even more slowly than the tertiary force within the nearest-neighbor cells. Accordingly, the local expansion coefficients of the Coulomb potential and microscopic stress tensor, $\Psi_{L,c}$ and $\Pi_{L,c}$, are updated at every n_2 time steps, and no Taylor extrapolation is made in between. The first term of Eq. (8) and the second term of Eq. (9) are evaluated at every time step at the updated atom positions, r_i, but without updating the Taylor-expansion coefficients for n_2 steps. The far-field contributions are then added to the direct contributions computed by the MTS method described above. Fig. 5 summarizes the different time steps used in the MRMD algorithm.

2.4. Separable three-body force calculation

The near-field potential also contains a three-body component representing covalent bonding. Triple sums in the three-body potential makes the computation inefficient, and thus the three-body force calculation is one of the most time consuming parts of the MRMD algorithm [11].

We use a tensor decomposition scheme to rewrite the potential into a separable form [11,12]. In this scheme the three-body potential in Eq. (1) is rewritten by decomposing it into tensor components. The resulting expression is separable into two factors, one of which contains only $\boldsymbol{r}_{ij}$ and the other contains only $\boldsymbol{r}_{ik}$:

$$v_{jik}^{(3)}(\boldsymbol{r}_{ij}, \boldsymbol{r}_{ik}) = \sum_{\alpha} w_{ij}^{(\alpha)}(\boldsymbol{r}_{ij}) w_{ik}^{(\alpha)}(\boldsymbol{r}_{ik}). \tag{21}$$

The detailed expression for the decomposed potential is given in Ref. [11]. The resulting expressions for the potential energy, interatomic forces, and microscopic stress tensor involve only double sums. As pointed out by Frenkel [12], all the functions are then evaluated in a similar way as pairwise interactions. Speedup techniques such as the use of Newton's third law reduce the computation time and overall we achieve a speedup by a factor of two in the three-body force calculation [11].

3. Implementation of the multiresolution molecular dynamics on parallel computers

3.1. Domain decomposition for the direct-force calculation

We use the divide-and-conquer strategy based on domain decomposition to implement the MRMD algorithm on parallel computers [2,7,10,11]. The total volume of the system is divided into p subsystems of equal volume, and each subsystem is assigned to a node in an array of p processors. The data associated with particles of a subsystem are assigned to the corresponding node. In the fifth-order predictor-corrector method [15], each node stores an array containing the coordinates and their time derivatives up to the fifth order, $\boldsymbol{r}_i^{(\lambda)}(i = 1,\ldots, N;\ \lambda = 0,\ldots, 5)$, where N is the number of particles in the subsystem (except for the message passing explained below, each node program runs autonomously, so we omit the processor index here). In addition, another array $\sigma_i(i = 1,\ldots, N)$ specifies the species (Si or O) of the particles.

To calculate the direct force on a particle in a subsystem, the coordinates of the particles in the boundaries of 26 neighbor subsystems must be copied from the corresponding nodes. Here, we distinguish the primary- and secondary-boundary particles. In a subsystem, primary boundary particles to the nth neighbor ($n = 1,\ldots, 26$) subsystem are located within a distance $r_a + \delta$ from the boundary with the nth neighbor, where δ is a small positive number. Similarly, secondary boundary particles are those which are within a distance $r_c + \delta'(\delta'$ is another small distance) from the boundary but are not the primary particles.

Whenever *steps* mod $n_1 = 1$, lists of the primary- and secondary-boundary particles are constructed. The data σ_i and $\boldsymbol{r}_i^{(\lambda)}$ $(\lambda = 0,\ldots, \lambda_{\max})$ for primary- and secondary-boundary particles are then transmitted to the neighbor nodes using message passing routines. The arrays, σ_i and $\boldsymbol{r}_i^{(\lambda)}$, are augmented including the copied particles. When *steps* mod $n_1 \neq 1$, only the positions of primary-boundary particles which have been defined at the latest update of the primary-boundary-particle list are transmitted. In the actual code, the message passing to the 26 neighbor nodes is completed in six steps by sending the boundary-particle information to east, west, north, south, up and down neighbor nodes sequentially. The corner and edge boundary particles are copied to proper neighbor nodes by forwarding some of the received boundary particles to other neighbor nodes [11].

After every n_1 MD steps (*steps* mod $n_1 = 0$), a resident particle may move out of its subsystem. Therefore, we check the coordinates of the N resident particles before the next n_1 steps start. Particles which have moved out of a subsystem are sent to the proper neighbor nodes. Newly arrived data are first appended to σ_i and $\boldsymbol{r}_i^{(\lambda)}$ arrays, and then the arrays are compressed by removing the elements which have moved out. At this stage of the algorithm, we update N to represent the number of new resident particles.

In the parallel MTS algorithm, message passing associated with the secondary boundary particles is skipped for n_1 steps, and the only message passing performed at every time step is associated with the primary boundary particles whose number is very small. Consequently, the MTS method dramatically reduces the size of messages [11].

The above message passing is implemented by using asynchronous receive and synchronous copy routines [18,19]. A synchronous send routine blocks the execution of the subsequent lines in the program until the message has been sent out. On the other hand, an asynchronous receive routine does not block the subsequent lines. In the program, an asynchronous receive routine is first called, followed by a corresponding synchronous send routine, to ensure that the information sent by a neighbor node is received as soon as the message arrives.

3.2. Domain decomposition for long-range force calculation

The implementation of hierarchical codes on parallel computers have been reported before [16,20,21]. Suppose the processors are logically organized as a cube of size $p_x \times p_y \times p_z$. For a deeper tree level, $l \geq \log_2[\max(p_x, p_y, p_z)]$, each cell is uniquely assigned to a processor (We consider only those cases where $\max(p_x, p_y, p_z) \leq 2 \min(p_x, p_y, p_z)$). The calculation of the multipoles according to Eq. (14) is local to a processor. To compute the local expansions in Eqs. (18) and (19), we must distinguish the inherited terms from the parent and contributions from the interactive set. The computation of inherited terms is local to a processor. For the computation of the interactive-cell contribution, the multipoles of two skin-layer cells must be copied from the nearest-neighbor nodes, see Fig. 3. In our MRMD algorithm, the multipoles of $8^l/p$ cells per node are augmented with the copied multipoles to form an array consisting of multipoles of $(2^l/p_x + 2)(2^l/p_y + 2)(2^l/p_z + 2)$ cells at each layer. Synchronous copy and asynchronous receive routines are again used to carry out necessary message passing.

However, for lower levels, the number of cells becomes smaller than the number of processors. Here more than one processors is assigned to a physical cell. Consequently many processors become idle or alternatively they duplicate the same computation. In our parallel implementation of the MRMD algorithm, the multipoles for all the cells at a lower level, $l < \log_2[\max(p_x, p_y, p_z)]$, are set to be global variables, see Fig. 6. In the upward pass on the tree, each node computes locally the multipoles of the cells to which it is assigned. The local multipoles are then combined to obtain an array containing all the multipoles for 8^l cells. Global concatenation or gather routines are used to collect local contributions and form a global array [18,19]. In the downward pass, each node computes the local expansions of a uniquely

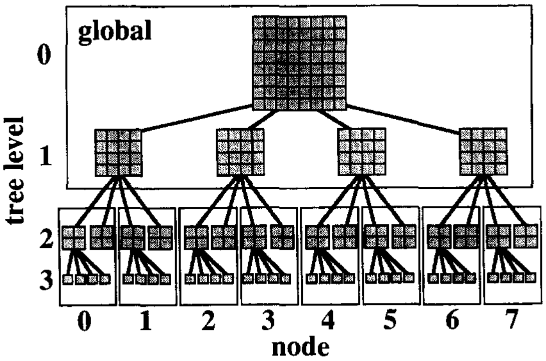

Fig. 6. Assignment of the hierarchy of cells to multiprocessors for a two-dimensional system. For deeper levels, cells are local to a processor. For lower levels, cell information is made global to all the processors.

assigned cell for $l < \log_2[\max(p_x, p_y, p_z)]$, using global information for the multipoles. For $l \geq \log_2[\max(p_x, p_y, p_z)]$, each node computes local expansions for $8^l/p$ cells using the augmented multipoles. A pseudocode for one MD step of the MRMD algorithm is given in the Appendix.

The problem of idle processors at lower levels is inherent in all hierarchical algorithms. It reduces the theoretical minimum parallel execution time to be $O(\log N)$ for an N particle system using N processors [16,20]. For multimillion-particle simulations on less than 512 processors, however, the number of the finest cells, N_{cell}, is much larger than p. For such large grain sizes, we found that the idle processor problem is negligible and the program is practically scalable as demonstrated by the parallel efficiency analysis in the next section.

4. Performance of multiresolution MD algorithm on parallel computers

We have carried out performance tests of the MRMD algorithm on parallel computers. The physical system is SiO_2 modeled with a realistic interatomic potential [3]. The machines we use are the 512-node Intel Touchstone Delta at Caltech and the 128-node IBM SP1 at Argonne. Both are distributed-memory MIMD (multiple instruction multiple data) machines. The SP1 has a high-speed switch for interprocessor communication. The low-latency EUIH interface is used for switch-based communications [22]. The program is based on the message-passing programming style. On the Delta, we use the iPSC Fortran which uses the native NX message passing [18]. On the SP1, we use the Chameleon programming system [19] which is portable to many parallel computers. All the calculations are done in 64-bit precision.

The accuracy and speed of the MRMD program depend on the parameters, n_1, n_2, and $\lambda_{\max}$, in the MTS method. The dependence of the accuracy and speed on $\lambda_{\max}$ is analyzed in Ref. [11]. Larger n_1 and n_2 achieve a higher speed but with less accuracy. There is a trade-off between speed and accuracy, as shown in Fig. 7, where the speed and accuracy of the MRMD program are plotted as a function of n_2. The speed is expressed in terms of MD steps executed per hour for a 393, 216 particle SiO_2 system on 256 nodes of the Delta. The accuracy is the deviation in the total energy during 1,000 MD steps normalized by the total energy. Here we choose the other MTS parameters to be $n_1 = 15$ and $\lambda_{\max} = 3$.

Here and in the following benchmark tests, we use $K = 2$, i.e., we retain terms up to quadrupoles. The dependence of the accuracy of the FMM and RCMM on the order of multipole expansions, K, is systematically studied in Refs. [8] and [17]. To test the accuracy of the MRMD algorithm, we have implemented another parallel MD algorithm in which the long-range Coulomb interaction is computed

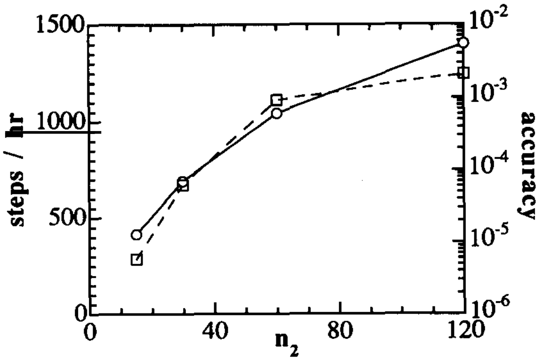

Fig. 7. Speed of the MRMD program in terms of MD steps executed per hour solid curve). The results are for 393, 216-particle SiO_2 system on 256 nodes of the Delta. Also shown is the accuracy in terms of the change in the total energy during 1,000 MD steps normalized by the total energy dashed curve).

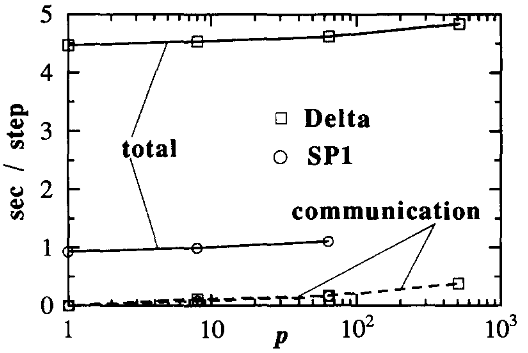

Fig. 8. Execution time of the MRMD program solid curves) for 8232p particle SiO_2 systems on p processors. Open squares and open circles are the results on the Delta and SP1, respectively. Communication overhead dashed curves) of the same program.

using the conventional Ewald method [7]. The system we consider is densified (4.28 g/cm^3) liquid SiO_2 at 3000 K [23]. For a 10080-particle MD configuration, the potential energy calculated with the Ewald method is -1.5953×10^{-18} J/particle, while it is -1.5972×10^{-18} J/particle with the MRMD. The relative error in the potential energy calculated by the MRMD is thus 1×10^{-3}. For the same configuration, the microscopic stress tensor (Tr(π)/3) is calculated as 83.3 GPa with the Ewald method and 81.3 GPa (relative error is 2×10^{-2}) with the MRMD.

Fig. 8 shows the execution time of the MRMD program on the Delta and SP1 as a function of the number of processors, p. The MTS parameters are chosen as $n_1 = 15$, $n_2 = 120$, and $\lambda_{max} = 1$. Here we scale the system size linearly with the number of processors so that the number of particles, $N = 8232p$. For a 4.2-million particle SiO_2 system on the 512 nodes Delta, one MD step takes only 4.84 seconds. The execution time increases only slightly for larger systems.

Fig. 8 also shows the time spent for communication expressed as a fraction of the total execution time. For the 512-node Delta, communication is only 8.1% of the total execution time. The small communication overhead makes the program efficient even for larger p.

On the SP1 the computation is more than four times faster than on the Delta, but the communication performance is comparable, see Fig. 8. For a 0.53-million particle SiO_2 system on 64 nodes of the SP1, one MD step takes 1.11 seconds and the communication overhead is 15.2%.

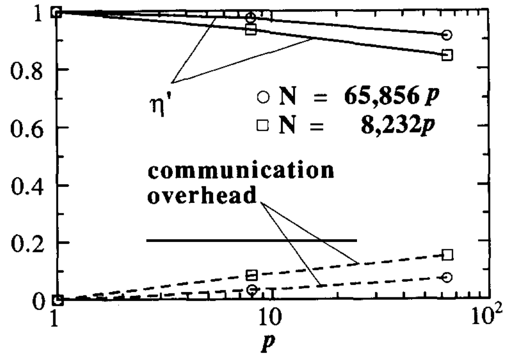

Fig. 9. Constant-grain parallel efficiency, η', of the MRMD program for SiO_2 systems solid curves) on p processors of the SP1. Open circles and open squares are the results for $8232p$ particles and $65856p$ particles, respectively. Communication overheads dashed curves) of the same program are also shown.

Next we examine the parallel efficiency of the MRMD program. We first define the speed of the MRMD program as a product of the total number of particles and time steps executed in a second. The constant-grain speedup is given by the ratio between the speed of p processors and that of one processor. The constant-grain efficiency, η', is the constant-grain speedup divided by p [11].

For a 4.2-million particle SiO_2 system on the 512 nodes Delta, we estimate $\eta' = 0.924$. Since the processing power of a SP1 node is greater than that of Delta but the communication speed remains similar (see Fig. 8), the parallel efficiency of the MRMD program on the SP1 is smaller than on the Delta. In Fig. 9, open squares represent the parallel efficiency of the MRMD program as a function of p for a system of $8232p$ particles on the SP1. The parallel efficiency decreases as the number of processors increases.

The above low efficiency is due to the small grain size; larger grain sizes result in higher efficiencies. In Fig. 9, open circles represent the efficiency for systems with $65856p$ particles. Also shown is the communication overhead of the MRMD program as a function of p. The larger grain size reduces the communication overhead. For a 4.2-million particle SiO_2 system on 64 nodes of the SP1, we estimate $\eta' = 0.914$ with 7.3% communication overhead.

Figure 10 shows the decomposition of the execution time into core computation tasks. Here, *up* represents the upper pass of the FMM to calculate the multipoles, while *ewald* represents the calculation of the well-separated image contribution to the Coulomb potential and microscopic stress tensor. Both on the Delta and SP1, these computations consume negligible time. *down* is the calculation of the local expansions in the FMM by the downward pass.

In our implementation of the MRMD program, *direct* – the direct-force calculation – is the most time consuming. Other tasks include the predictor and corrector procedures in Gear's algorithm to integrate the equations of motion (denoted as *p&c*). Message passing of the data associated with the boundary

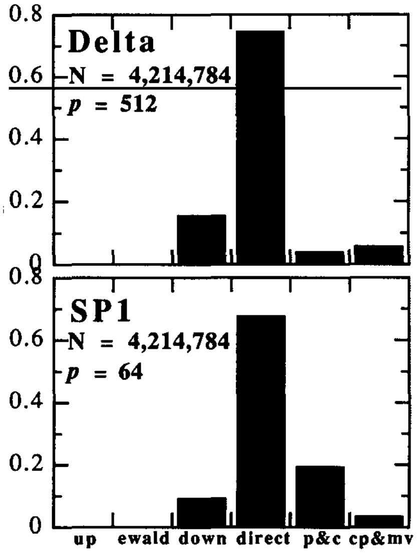

Fig. 10. Decomposition of the execution time per MD step into core computation tasks of the MRMD program on the Delta and SP1. *up* and *down* represent the upward and downward passes of the FMM algorithm. The necessary message passing of multipoles is also included in *up*. *ewald* is the computation of the Ewald summation of the well-separated image contribution to the Coulomb potential based on the RCMM. *direct* represents the direct force calculation within the nearest-neighbor finest cells. *p&c* is the predictor and corrector steps in Gear's predictor-corrector method to integrate the equations of motion. The copy and receive operations of the boundary-atom information necessary for the direct force calculation are denoted by *cp&mv*.

particles and those leaving the boundary (denoted by *cp&mv*) is a small fraction of the execution time. Message passing associated with the multipoles of the two skin-layer cells is included in the *up* category in Fig. 10, and is negligibly small.

5. Conclusion

In conclusion, the multiresolution MD algorithm is applicable for large-scale simulations with realistic interatomic potential including the long-range Coulomb and three-body potentials. More specifically, multimillion particle simulations take only a few seconds per step on advanced parallel computers such as the Intel Touchstone Delta and IBM SP1.

Acknowledgements

This work was supported by the U.S. Department of Energy, Grant No. DE-FG05-92ER45477. This research was performed in part using the 512-node Intel Touchstone Delta operated by Caltech on behalf of the Concurrent Supercomputing Consortium and the 128-node IBM SP1 computer at Argonne National Laboratory. The computations were also performed using the Intel iPSC/860 in the Concurrent Computing Laboratory for Materials Simulations (CCLMS) at Louisiana State University. The facilities in the CCLMS were acquired with the Equipment Enhancement Grants awarded by the Louisiana Board of Regents through Louisiana Education Quality Support Fund (LEQSF).

Appendix. Multiresolution MD algorithm

```
A. Predictor
predictor for particles, $r_i$
if $steps$ mod $n_1 = 1$
    predictor for the MD box, $\mathbf{h}$
endif

B. Far-field interaction
Update the local expansion, $\Psi_{l,c}$ and $\Pi_{l,c}$, at every $n_2$ steps
if $steps$ mod $n_2 = 1$
    B.1. Upward pass
    Form multipole expansions at the levels
    for $c = 0, \ldots, 8^L - 1$
        Calculate $\Phi_{L,c}$ at the leaf level
    end
    Form multipole expansions at lower tree levels
    for $l = L - 1, L - 2, \cdots, 1, 0$
        for $c = 0, \ldots, 8^l - 1$
            for $c' \in \{\mathrm{child}(c)\}$
                Multipole-to-multipole shift $(T_1)$ $\Phi_{l+1,c'}$ and add it to $\Phi_{l,c}$
            end
        end
    end
    Skin-layer-cell copy
    Augment multipoles with the skin-layer cells of the neighbor nodes
    B.2 Ewald contribution to the local expansion from well-separated images
    Determine the $M$ reduced charges using the multipoles, $\Phi_{0,0}$, of the unit MD box
    Calculate local expansions, $\Psi_{0,0}$ and $\Pi_{0,0}$, by the Ewald summation on the $M$ reduced charges based
    on the RCMM

    B.3. Downward pass
    Recursively calculate the local expansions
    for $l = 1, \cdots, L$
      for $c = 0, \ldots, 8^l - 1$
        Inherited local expansion from the parent
        Local-to-local transform ($T_3$ and $S_3$) of the parent local field, $\Psi_{l-1,\,\text{parent}\,(c)}$ and
        $\Pi_{l-1,\,\text{parent}\,(c)}$, to obtain the inherited components of $\Psi_{l,c}$ and $\Pi_{l,c}$
        Contribution from interactive set
        for $c' \in \{\text{interactive}(c)\}$
          Multipole-to-local shift ($T_2$ and $S_2$) of $\Phi_{l,c'}$ and add them to $\Psi_{l,c}$ and $\Pi_{l,c}$
        end
      end
    end
endif
Evaluation of the far-field interaction at the leaf level
Calculate the Coulombic contribution to the potential energy, $V$, interatomic forces, $\boldsymbol{F}_i$, and pressure
tensor, $\boldsymbol{\pi}$, using the local expansions, $\Psi_{L,c(i)}$ and $\Pi_{L,c(i)}$


C. Boundary-atom copy
    if $steps$ mod $n_1 = 1$
      Copy the atomic positions and derivatives of the primary and secondary boundary atoms
    else
      Copy the atomic positions of the primary boundary atoms
    endif


D. Direct calculation of the medium-range interaction
    if $steps$ mod $n_2 = 1$
      Compute the primary, secondary, and tertiary forces directly
      Construct a primary-neighbor list
      Update the secondary and tertiary forces and store the time derivatives, $\boldsymbol{F}^{(\alpha)}_{i,s}$ and $\boldsymbol{F}^{(\alpha)}_{i,t}$
    else if $steps$ mod $n_1 = 1$
      Compute the primary and secondary forces directly
      Construct a primary-neighbor list
      Update only the secondary forces and store the time derivatives, $\boldsymbol{F}^{(\alpha)}_{,s}$
      Taylor extrapolation of the tertiary forces
    else
      Calculate the binary, primary forces, $\boldsymbol{F}_{i,p}$, using the primary-neighbor list
      Taylor extrapolation of the secondary and tertiary forces
    endif
    Calculate the three-body forces using the primary-neighbor list
```

```
E. Corrector
  Corrector for particles, $\boldsymbol{x}_i$
  **if** *steps* mod $n_1 = 1$
    Corrector for the MD box, $\boldsymbol{h}$
  **endif**

F. Move atoms
  **if** *steps* mod $n_1 = 0$
    Move atoms which have cross the subsystem border to proper processors
  **endif**
```

References

[1] R.W. Siegel, in: Atomic-Level Properties of Interface Materials, eds. D. Wolf and S. Yip (Chapman and Hall, London, 1992). H. Gleiter, Nanostructured Materials 1 (1992) 1.

[2] D.C. Rapaport, Comput. Phys. Commun. 62 (1991) 217.

[3] P. Vashishta, R.K. Kalia, J.P. Rino and I. Ebbsjö, Phys. Rev. B 41 (1990) 12197.

[4] A.W. Appel, SIAM J. Sci. Stat. Comput. 6 (1985) 85.

[5] J. Barnes and P. Hut, Nature 324 (1986) 446.

[6] L. Greengard and V. Rokhlin, J. Comput. Phys. 73 (1987) 325.

[7] R.K. Kalia, S.W. de Leeuw, A. Nakano and P. Vashishta, Comput. Phys. Commun. 74 (1993) 316.

[8] H.-Q. Ding, N. Karasawa and W.A. Goddard, Chem. Phys. Lett. 196 (1992) 6.

[9] W.B. Street, D.J. Tildesley and G. Saville, Mol. Phys. 35 (1978) 639.

[10] R.K. Kalia, A. Nakano, D.L. Greenwell, P. Vashishta and S.W. de Leeuw, Supercomputer 54(X-2) (1993) 11.

[11] A. Nakano, P. Vashishta and R.K. Kalia, Comput. Phys. Commun. 77 (1993) 302.

[12] D. Frenkel, in: Simple Molecular Systems at Very High Density, eds. A. Polian and P. Loubeyre (Plenum, New York, 1989) p. 411.

[13] M.P. Allen and D.J. Tildesley, Computer Simulation of Liquids (Oxford Univ. Press, Oxford, 1987).

[14] M. Parrinello and A. Rahman, Phys. Rev. Lett. 45 (1980) 1196; J. Appl. Phys. 52 (1981) 7182.

[15] C.W. Gear, Numerical Initial Value Problem in Ordinary Differential Equations (Prentice-Hall, Englewood Cliffs, NJ, 1971).

[16] F. Zhao and S. Lennart Johnsson, SIAM J. Sci. Stat. Comput. 12 (1991) 1420.

[17] H.-Q. Ding, N. Karasawa, and W.A. Goddard, J. Chem. Phys. 97 (1992) 4309.

[18] iPSC®/2 and iPSC®/860 Programmers's Reference Manual, Technical Report 311708-004, Intel Corporation (1991).

[19] W. Gropp and B. Smith, Users Manual for the Chameleon Parallel Programming Tools, Technical Report ANL-93/23, Argonne National Laboratory (1993).

[20] L. Greengard and W.D. Gropp, in: Parallel Processing for Scientific Computing, ed. G. Rodrigue (SIAM, Philadelphia, 1989) p. 213.

[21] J.A. Board et al., Chem. Phys. Lett. 198 (1992) 89.

[22] W. Gropp and E. Lusk, Users Guide for the ANL IBM SP-1 Draft, Technical Report ANL/MCS-TM-00, Argonne National Laboratory (1994).

[23] W. Jin, R.K. Kalia, P. Vashishta and J.P. Rino, Phys. Rev. Lett. 71 (1993) 3146.