



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

SCIENCE @ DIRECT®

Computer Physics Communications 167 (2005) 151–164

Computer Physics  
Communications

[www.elsevier.com/locate/cpc](http://www.elsevier.com/locate/cpc)

# Embedded divide-and-conquer algorithm on hierarchical real-space grids: parallel molecular dynamics simulation based on linear-scaling density functional theory

Fuyuki Shimojo<sup>a,b</sup>, Rajiv K. Kalia<sup>a</sup>, Aiichiro Nakano<sup>a,\*</sup>, Priya Vashishta<sup>a</sup>

<sup>a</sup> *Collaboratory for Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Materials Science & Engineering, University of Southern California, Los Angeles, CA 90089-0242, USA*

<sup>b</sup> *Department of Physics, Kumamoto University, Kumamoto 860-8555, Japan*

Received 5 October 2004; received in revised form 24 January 2005; accepted 26 January 2005

Available online 14 March 2005

## Abstract

A linear-scaling algorithm has been developed to perform large-scale molecular-dynamics (MD) simulations, in which interatomic forces are computed quantum mechanically in the framework of the density functional theory. A divide-and-conquer algorithm is used to compute the electronic structure, where non-additive contribution to the kinetic energy is included with an embedded cluster scheme. Electronic wave functions are represented on a real-space grid, which is augmented with coarse multigrids to accelerate the convergence of iterative solutions and adaptive fine grids around atoms to accurately calculate ionic pseudopotentials. Spatial decomposition is employed to implement the hierarchical-grid algorithm on massively parallel computers. A converged solution to the electronic-structure problem is obtained for a 32,768-atom amorphous CdSe system on 512 IBM POWER4 processors. The total energy is well conserved during MD simulations of liquid Rb, showing the applicability of this algorithm to first principles MD simulations. The parallel efficiency is 0.985 on 128 Intel Xeon processors for a 65,536-atom CdSe system.

© 2005 Elsevier B.V. All rights reserved.

PACS: 02.70.-c; 02.70.Ns; 71.15.-m

Keywords: Parallel computing; Molecular dynamics; Density functional theory; Linear scaling algorithm

\* Corresponding author.

E-mail address: [anakano@usc.edu](mailto:anakano@usc.edu) (A. Nakano).

## 1. Introduction

There is growing interest in large-scale molecular dynamics (MD) simulations involving multimillion atoms [1–3], in which interatomic forces are computed quantum mechanically [4,5] in the framework of the density functional theory (DFT) [6–8] to accurately describe chemical reactions. Such large DFT-based MD simulations would provide requisite coupling of chemical reactions, atomistic processes, and long-range stress phenomena for broad applications. Examples are energetic materials, in which chemical reactions sustain shock waves [9,10], and stress corrosion, where chemical reactions at the crack tip need to be coupled with long-range stress fields [11]. Unfortunately, DFT-based MD simulations are rarely performed over  $N \sim 10^2$  atoms because of their  $O(N^3)$  computational complexity, which severely limits their applicability.

One computational approach toward quantum mechanically informed million-atom MD simulations employs environment-dependent interatomic potentials [12], based on variable atomic charges [13–17] to describe charge transfers and bond orders [18,19] to describe bond formation and breakage. In the reactive force field (ReaxFF) approach [12], the parameters in the interatomic potential are “trained” to best fit thousands of DFT calculations on small ( $N \sim 10$ ) clusters of various atomic-species combinations [10]. Thus, the determination of the potential parameters prior to MD simulations constitutes the most critical and nontrivial part of the ReaxFF-MD approach.

Alternative to this sequential DFT-informed MD approach is to perform a number of small DFT calculations “on the fly” to compute interatomic forces quantum mechanically during an MD simulation [20–22]. The concurrent DFT-based MD approach may be realized using a divide-and-conquer (DC) algorithm [23–25]. The DC-DFT algorithm represents the physical system as a union of overlapping spatial domains, and physical properties are computed as linear combinations of domain properties. This algorithm is based on a data locality principle called the quantum near sightedness [26], which naturally leads to  $O(N)$  DFT calculations [27–29].

In this paper, we develop an extension of the DC algorithm, i.e. an embedded divide-and-conquer (EDC) algorithm, in which non-additive correction to the kinetic energy is included with an embedded cluster approach [30,31]. DFT calculations of atomic clusters (or domains) involved in the EDC algorithm are efficiently performed using a real-space approach, in which electronic wave functions are numerically represented on grid points [32,33]. The real-space grid is augmented with coarser multigrids to accelerate the convergence of iterative solutions [5,22,29,34–36]. Furthermore, a finer grid is adaptively generated near every atom [11,37], in order to accurately operate ionic pseudopotentials [38–40] to describe electron–ion interactions. We include electron–electron interactions using the generalized gradient approximation [41] to the exchange–correlation energy. Since the EDC-DFT algorithm involves solutions to small (i.e. the average number of atoms per domain,  $n < 50$ ) electronic-structure problems, it solves some of the convergence problems often associated with iterative solutions in large electronic-structure calculations. For example, it is difficult to achieve any convergence for nontrivial electronic-structure problems such as amorphous materials with a large number ( $> 10^4$ ) of atoms using a large ( $10^5$  grid points per electronic wave function) basis set, unless very good initial guess for the wave functions and the electron number density is available [29]. Our numerical tests show that the EDC-DFT algorithm has robust convergence properties such that, starting from random initial wave functions, it is able to obtain converged solutions to such problems.

The EDC-DFT algorithm on the hierarchical real-space grids is implemented on massively parallel computers, based on spatial decomposition [3,29], in which the physical system is divided into subsystems of equal volume and each subsystem is assigned to a processor in a parallel computer. Each subsystem (or processor) contains one or more domains of the EDC algorithm. For each domain, an  $O(n^3)$  DFT algorithm is used to calculate its electronic structures, with little information needed from other processors. The resulting large computation/communication ratio makes this approach highly scalable on parallel computers.

This paper is organized as follows. The next section describes the EDC-DFT algorithm for quantum mechanically based MD simulations, and its parallelization is discussed in Section 3. Numerical results are presented in Section 4, and finally Section 5 contains summary.

## 2. Embedded divide-and-conquer density-functional-theory algorithm on hierarchical real-space grids

In the Kohn–Sham formulation [7] of the density-functional theory (DFT) [6,8], the energy of a physical system is expressed as a functional of  $N$  atomic positions,  $\vec{R}^N = \{\vec{R}_I \mid I = 1, \dots, N\}$ , and  $N_{\text{band}}$  one-electron wave functions (or Kohn–Sham orbitals),  $\psi^{N_{\text{band}}} = \{\psi_n(\vec{r}) \mid n = 1, \dots, N_{\text{band}}\}$ . The energy functional in the atomic unit reads

$$E[\rho] = T_s[\rho] + \int d\vec{r} \rho(\vec{r}) v_{\text{loc}}(\vec{r}) + E_{\text{nl}}[\rho] + \frac{1}{2} \int d\vec{r} \int d\vec{r}' \frac{\rho(\vec{r})\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} + E_{\text{xc}}[\rho] + \sum_{I < J} \frac{Z_I Z_J}{|\vec{R}_I - \vec{R}_J|}, \quad (1)$$

where  $T_s[\rho]$  is the kinetic energy of a noninteracting electron gas in its ground state with the number density,  $\rho(\vec{r})$ , of valence electrons,  $E_{\text{xc}}[\rho]$  is the exchange–correlation energy, the last term is the electrostatic energy between ions with valence,  $\{Z_I \mid I = 1, \dots, N\}$ , and

$$v_{\text{loc}}(\vec{r}) = \sum_I v_I^{\text{loc}}(\vec{r} - \vec{R}_I) \quad (2)$$

is the local ionic pseudopotential. In Eq. (1), the nonlocal pseudopotential energy [38–40],  $E_{\text{nl}}[\rho]$ , is given by

$$E_{\text{nl}}[\rho] = \sum_n \sum_I \sum_{lm} \int d\vec{r} \psi_n^*(\vec{r}) \xi_{lm}^I(\vec{r} - \vec{R}_I) \int d\vec{r}' \xi_{lm}^{I*}(\vec{r}' - \vec{R}_I) \psi_n(\vec{r}'), \quad (3)$$

where  $\xi_{lm}^I(\vec{r} - \vec{R}_I)$  is a projection state localized at  $\vec{R}_I$  with the angular momentum quantum numbers,  $l$  and  $m$ . The valence electron number density,  $\rho(\vec{r})$ , is calculated as

$$\rho(\vec{r}) = 2 \sum_{n=1}^{N_{\text{band}}} |\psi_n(\vec{r})|^2, \quad (4)$$

where only doubly occupied states are considered.

The ground-state energy of the system, with given atomic positions,  $\vec{R}^N$ , is obtained by minimizing  $E[\rho]$  with respect to  $\psi^{N_{\text{band}}}$ , subjected to orthonormality constraints,

$$\int d\vec{r} \psi_n^*(\vec{r}) \psi_{n'}(\vec{r}) = \delta_{nn'} \equiv \begin{cases} 1 & (n = n'), \\ 0 & (n \neq n'). \end{cases} \quad (5)$$

The constrained minimization leads to the Kohn–Sham equations,

$$\hat{H} \psi_n(\vec{r}) = \varepsilon_n \psi_n(\vec{r}), \quad (6)$$

where the Kohn–Sham Hamiltonian operator,  $\hat{H}$ , is defined through

$$\hat{H} \psi_n(\vec{r}) = \left[ -\frac{1}{2} \nabla^2 + v_{\text{loc}}(\vec{r}) + v_{\text{H}}(\vec{r}) + v_{\text{xc}}(\vec{r}) \right] \psi_n(\vec{r}) + \sum_I \sum_{lm} \xi_{lm}^I(\vec{r} - \vec{R}_I) \int d\vec{r}' \xi_{lm}^{I*}(\vec{r}' - \vec{R}_I) \psi_n(\vec{r}'). \quad (7)$$

In Eq. (7),  $\nabla^2$  is the Laplacian operator, the Hartree potential is given by

$$v_{\text{H}}(\vec{r}) = \int d\vec{r}' \frac{\rho(\vec{r}')}{|\vec{r} - \vec{r}'|}, \quad (8)$$

and we use the generalized gradient approximation [41] to the exchange–correlation potential,  $v_{\text{xc}}(\vec{r}) = \delta E_{\text{xc}} / \delta \rho(\vec{r})$ , which is a function of the local value of  $\rho(\vec{r})$  and its gradient,  $\nabla \rho(\vec{r})$ .

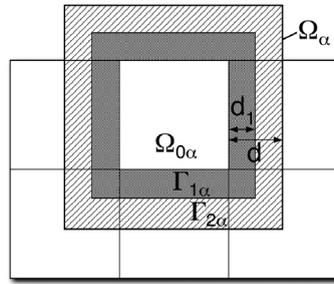


Fig. 1. Schematic of the divide-and-conquer algorithm in 2D. The physical space  $\Omega$  is a union of overlapping domains,  $\Omega = \bigcup_{\alpha} \Omega_{\alpha}$ . Each domain is further decomposed into the non-overlapping core,  $\Omega_{0\alpha}$ , the primary buffer layer,  $\Gamma_{1\alpha}$  (see the shaded area), and the secondary buffer layer,  $\Gamma_{2\alpha}$  (see the hatched area). The depth of the primary and total (= primary + secondary) buffer layers are  $d_1$  and  $d$ , respectively.

### 2.1. Divide-and-conquer density-functional-theory algorithm

In the divide-and-conquer (DC) algorithm [23–25,42], the three-dimensional space  $\Omega$  is covered with overlapping domains (see Fig. 1),

$$\Omega = \bigcup_{\alpha} \Omega_{\alpha}. \quad (9)$$

Each domain,  $\Omega_{\alpha}$ , is further decomposed into its sub-volumes,

$$\Omega_{\alpha} = \Omega_{0\alpha} \cup \Gamma_{1\alpha} \cup \Gamma_{2\alpha}, \quad (10)$$

where  $\Omega_{0\alpha}$  is the non-overlapping core,

$$\Omega = \bigcup_{\alpha} \Omega_{0\alpha}; \quad \Omega_{0\alpha} \cap \Omega_{0\beta} = 0 \quad (\alpha \neq \beta), \quad (11)$$

and  $\Gamma_{1\alpha}$  and  $\Gamma_{2\alpha}$  are the primary and secondary buffer layers, respectively [25].

For each domain  $\alpha$ , we define a domain support function,  $p^{\alpha}(\vec{r})$ , such that it is only nonzero within the core and primary buffer layer of the domain,

$$p^{\alpha}(\vec{r}) = 0 \quad \text{if } \vec{r} \notin \Omega_{0\alpha} \cup \Gamma_{1\alpha}, \quad (12)$$

with the sum rule,

$$\sum_{\alpha} p^{\alpha}(\vec{r}) = 1, \quad (13)$$

satisfied at every spatial position,  $\vec{r}$ . Because of the sum rule, the valence electron number density is exactly decomposed into

$$\rho(\vec{r}) = \sum_{\alpha} \rho^{\alpha}(\vec{r}), \quad (14)$$

where

$$\rho^{\alpha}(\vec{r}) = p^{\alpha}(\vec{r})\rho(\vec{r}) \quad (15)$$

is the partial contribution to the electron density from domain  $\alpha$ .

The essential approximation in the DC-DFT algorithm is the replacement of the self-consistent Kohn–Sham Hamiltonian,  $\hat{H}$ , by its subspace approximation,  $\hat{H}_{\alpha}$ , which is formally identical to Eq. (7), but the Kohn–Sham

equations, Eq. (6), are solved in each domain,  $\Omega_\alpha$ , to obtain locally orthonormal Kohn–Sham orbitals,  $\{\psi_n^\alpha(\vec{r})\}$ :

$$\int_{\Omega_\alpha} d\vec{r} \psi_n^{\alpha*}(\vec{r}) \psi_{n'}^\alpha(\vec{r}) = \delta_{nn'} \equiv \begin{cases} 1 & (n = n'), \\ 0 & (n \neq n'). \end{cases} \quad (16)$$

Boundary conditions on  $\{\psi_n^\alpha(\vec{r})\}$  are imposed at the domain boundary,  $\partial\Omega_\alpha$ . We use either the rigid-wall boundary condition, in which the wave function vanishes at the boundary, or the periodic boundary condition. The wave function values in the secondary buffer layer,  $\Gamma_{2\alpha}$ , may be contaminated by the artificial boundary conditions at  $\partial\Omega_\alpha$ . This is the reason why the domain support function,  $p^\alpha(\vec{r})$ , is made zero in  $\Gamma_{2\alpha}$  [25], so that the contaminated wave function values do not contribute to the density,  $\rho(\vec{r})$ .

To determine the number of occupied local Kohn–Sham orbitals in the DC-DFT algorithm, we first note that Eq. (4) is an expansion of the valence electron density,

$$\rho(\vec{r}) = 2|\vec{r}\rangle\Theta(\varepsilon_F - \hat{H})|\vec{r}\rangle, \quad (17)$$

in terms of the eigenstates of the Kohn–Sham Hamiltonian,  $\hat{H}$ . In Eq. (17),  $|\vec{r}\rangle$  is the coordinate eigenstate, the step function,  $\Theta(x)$ , is 1 for  $x > 0$  and is 0 otherwise, and the Fermi energy,  $\varepsilon_F$ , is determined from the number of valence electrons,  $N_{\text{el}}$ , through the relation,

$$N_{\text{el}} = \int d\vec{r} \rho(\vec{r}). \quad (18)$$

In the DC-DFT algorithm, the local Kohn–Sham orbitals,  $\{\psi_n^\alpha(\vec{r})\}$ , are compactly supported on each domain,  $\Omega_\alpha$ , and thus the partial density,  $\rho^\alpha(\vec{r})$ , is expanded in terms of them as

$$\rho^\alpha(\vec{r}) = p^\alpha(\vec{r}) \sum_n f(\varepsilon_n^\alpha) |\psi_n^\alpha(\vec{r})|^2, \quad (19)$$

where  $\varepsilon_n^\alpha$  is the  $n$ th eigenvalue of  $\hat{H}_\alpha$ , and  $f(\varepsilon_n^\alpha) = 2\Theta(\varepsilon_F - \varepsilon_n^\alpha)$ . Accordingly, the normalization condition, Eq. (18), reads

$$N_{\text{el}} = \sum_\alpha \sum_n f(\varepsilon_n^\alpha) \int d\vec{r} p^\alpha(\vec{r}) |\psi_n^\alpha(\vec{r})|^2. \quad (20)$$

In MD simulations, we numerically integrate Newton's equations of motion,

$$M_I \frac{d^2}{dt^2} \vec{R}_I = \vec{F}_I, \quad (21)$$

where the force,  $\vec{F}_I$ , acting on the  $I$ th ion, is calculated according to the Hellmann–Feynman theorem as

$$\begin{aligned} \vec{F}_I &= \vec{F}_I^{\text{ion}} + \vec{F}_I^{\text{loc}} + \vec{F}_I^{\text{nl}} \\ &= \sum_{J(\neq I)} Z_I Z_J \frac{\vec{R}_I - \vec{R}_J}{|\vec{R}_I - \vec{R}_J|^3} + \int d\vec{r} \rho(\vec{r}) (\vec{r} - \vec{R}_I) \frac{dv_{\text{loc}}}{d|\vec{r} - \vec{R}_I|} \\ &\quad + \sum_\alpha \sum_n f(\varepsilon_n^\alpha) \sum_{lm} \left[ \int d\vec{r} p^\alpha(\vec{r}) \psi_n^{\alpha*}(\vec{r}) \frac{\partial \xi_{lm}^I}{\partial(\vec{r} - \vec{R}_I)} \int d\vec{r}' \xi_{lm}^{I*}(\vec{r}' - \vec{R}_I) \psi_n^\alpha(\vec{r}') \right. \\ &\quad \left. + \int d\vec{r} p^\alpha(\vec{r}) \psi_n^{\alpha*}(\vec{r}) \xi_{lm}^I(\vec{r} - \vec{R}_I) \int d\vec{r}' \frac{\partial \xi_{lm}^{I*}}{\partial(\vec{r}' - \vec{R}_I)} \psi_n^\alpha(\vec{r}') \right]. \end{aligned} \quad (22)$$

In Eq. (22), the ionic contribution,  $\vec{F}_I^{\text{ion}}$ , can be computed efficiently with  $O(N)$  operations using the fast multipole method (FMM) [43]. Scalable FMM algorithms to compute  $\vec{F}_I^{\text{ion}}$  on parallel computers are described in our previous publications [44,45]. For systems with the periodic boundary condition, methods based on the Ewald summation are effective, including the  $O(N \log N)$  particle mesh Ewald method [46]. (A parallel Ewald algorithm is described in Ref. [47].)

## 2.2. Embedded-cluster scheme for non-additive kinetic energy correction

The DC-DFT algorithm amounts to approximating the kinetic energy functional,  $T_s[\rho]$ , as

$$T_s[\rho] = \sum_{\alpha} T_s^*[\rho^{\alpha}], \quad (23)$$

where

$$T_s^*[\rho^{\alpha}] = \sum_n f(\varepsilon_n^{\alpha}) \int d\vec{r} \rho^{\alpha}(\vec{r}) \psi_n^{\alpha*}(\vec{r}) \left( -\frac{1}{2} \nabla^2 \right) \psi_n^{\alpha}(\vec{r}), \quad (24)$$

i.e. the kinetic energy is additive with respect to domain contributions. Its non-additive correction can be introduced using an embedded-cluster scheme [30,31], in which the kinetic energy of the total system is expressed as

$$T_s[\rho] = \sum_{\alpha} T_s^*[\rho^{\alpha}] + T_s[\rho] - \sum_{\alpha} T_s^*[\rho^{\alpha}], \quad (25)$$

and an explicit functional of the electron density is used for the second and third terms. For example, the Thomas–Fermi approximation leads to an expression for the kinetic energy,

$$T_s[\rho] = \sum_{\alpha} T_s^*[\rho^{\alpha}] + C_{\text{TF}} \left( \int d\vec{r} \rho(\vec{r})^{5/3} - \sum_{\alpha} \int d\vec{r} \rho^{\alpha}(\vec{r})^{5/3} \right), \quad (26)$$

where  $C_{\text{TF}} = 3(3\pi^2)^{2/3}/10$ . Accordingly, the local Kohn–Sham Hamiltonian becomes

$$\begin{aligned} \hat{H}^{\alpha} \psi_n^{\alpha}(\vec{r}) = & \left[ -\frac{1}{2} \nabla^2 + v_{\text{loc}}(\vec{r}) + v_{\text{H}}(\vec{r}) + v_{\text{xc}}(\vec{r}) + \frac{5}{3} C_{\text{TF}} (\rho(\vec{r})^{2/3} - \rho^{\alpha}(\vec{r})^{2/3}) \right] \psi_n^{\alpha}(\vec{r}) \\ & + \sum_l \sum_{lm} \xi_{lm}^l(\vec{r} - \vec{R}_l) \int d\vec{r}' \xi_{lm}^{l*}(\vec{r}' - \vec{R}_l) \psi_n^{\alpha}(\vec{r}'). \end{aligned} \quad (27)$$

## 2.3. Hierarchical real-space grids

For efficient parallel implementation of DFT, we have developed a hierarchical real-space grid method, based on higher-order finite differencing [32,33] and multigrid acceleration [34,35]. In the hierarchical grid method, the real-space multigrids are adaptively refined [37] near each atom to accurately operate the ionic pseudopotentials on the electronic wave functions (see Fig. 2).

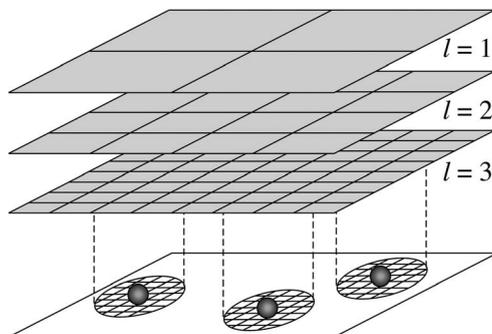


Fig. 2. Schematic of hierarchical real-space grids in 2D. Coarse multigrids (gray) are used to accelerate iterative solutions of the DFT problem on the original real-space grid (corresponding to the grid refinement level,  $l = 3$ ). The bottom panel shows that fine grids are adaptively generated near the atoms (spheres) to accurately operate the ionic pseudopotentials on the wave functions.

In the high-order finite difference method for calculating the derivatives (i.e. kinetic-energy operator) in Eq. (27),  $\psi_n^\alpha(\vec{r})$  and  $\rho(\vec{r})$  are represented by numerical values on real-space grid points. The kinetic energy operator is expanded using the finite difference method as [32]

$$\left. \frac{\partial^2 \psi_n(\vec{r})}{\partial x^2} \right|_{\vec{r}=(x_i, y_j, z_k)} = \sum_{\lambda=-L}^L C_\lambda \psi_n^\alpha(x_i + \lambda h, y_j, z_k) + O(h^{2L+2}), \quad (28)$$

where  $h$  is the grid spacing, and  $L$  is the order of the finite difference method. Since the calculations are performed entirely in real space, this method is most suitable for localized configurations such as clusters, which is the case in the EDC-DFT algorithm. Since the finite difference expansion involves only short-ranged operations, an efficient implementation on parallel computers is possible [28,29].

The constrained minimization of  $E[\rho]$  with respect to  $\{\psi_n^\alpha\}$  is performed by searching for an energy-minimum iteratively based on the conjugate-gradient (CG) method [48]. The self-consistent minimization loop consists of the following steps:

- (i) Calculate the electronic potentials,  $v_{\text{loc}}(\vec{r})$ ,  $\{\xi_{lm}^I(\vec{r} - \vec{R}_I)\}$ ,  $v_{\text{H}}(\vec{r})$ , and  $v_{\text{xc}}(\vec{r})$ , using the density,  $\rho(\vec{r})$ , from the previous step or the starting value (in case of the first step); these potentials are common to all the domains; the Hartree potential is obtained by iterative solution to the Poisson equation,  $\nabla^2 v_{\text{H}}(\vec{r}) = -4\pi\rho(\vec{r})$ , using the multigrid method [49].
- (ii) For all domains  $\alpha$ , perform the following.
  - (a) Perform a unitary transformation of  $\{\psi_n^\alpha\}$  to diagonalize the Hamiltonian matrix,  $H_{nn'}^\alpha = \int_{\Omega_\alpha} d\vec{r} \psi_n^{\alpha*}(\vec{r}) \times \hat{H}^\alpha \psi_{n'}^\alpha(\vec{r})$ .
  - (b) Improve  $\{\psi_n^\alpha(\vec{r})\}$  iteratively.
  - (c) Orthonormalize  $\{\psi_n^\alpha(\vec{r})\}$  according to Eq. (16) with the Gram–Schmidt method [48].
- (iii) Determine the Fermi energy,  $\varepsilon_{\text{F}}$ , to reproduce the number of valence electrons,  $N_{\text{el}}$ , by solving Eq. (20) by the Newton–Raphson method [49].
- (iv) Calculate the new density,  $\rho(\vec{r})$ , according to Eqs. (14) and (19), using the updated  $\{\psi_n^\alpha(\vec{r})\}$ ; refine  $\rho(\vec{r})$  by mixing the new and old densities using the Pulay charge-mixing scheme [50].

The above steps, (i)–(iv), are repeated until the self-consistency between  $\{\psi_n^\alpha\}$  and  $\rho(\vec{r})$  is achieved within a prescribed error tolerance.

The step (ii)(b) in the self-consistent loop includes two inner loops: One is associated with the band index  $n$  and the other is the CG iteration for each band involving a loop index,  $i_{\text{CG}}$ . To reduce long wavelength components of the residual, we use the multigrid method [11,29,34]. On a coarser grid, the following residual equations are solved,

$$\left\{ -\frac{1}{2}\nabla^2 + v(\vec{r}) \right\} \varphi_n^\alpha(\vec{r}) = g_n^\alpha(\vec{r}). \quad (29)$$

In this method, the electronic potential,  $v(\vec{r})$ , as well as the residual vector,  $g_n^\alpha(\vec{r})$ , on a fine grid are restricted to a coarser grid by using restriction operations. Here,

$$v(\vec{r}) = v_{\text{loc}}(\vec{r}) + v_{\text{H}}(\vec{r}) + v_{\text{xc}}(\vec{r}) + \frac{5}{3}C_{\text{TF}}(\rho(\vec{r})^{2/3} - \rho^\alpha(\vec{r})^{2/3}) \quad (30)$$

excludes the nonlocal pseudopotential, since it is short-ranged and does not influence the solutions on coarser grids. The solution,  $\varphi_n^\alpha(\vec{r})$ , of Eq. (29) on the coarse grid is prolonged to the fine grid, and is added to  $\psi_n^\alpha(\vec{r})$ .

The computational complexity of the EDC-DFT algorithm is  $O(N)$  for  $N$  atoms, as analyzed below. Since the most time consuming part is step (ii), we concentrate on the computational complexity of this step. The orthogonalization in step (ii)(c) requires  $O(N_{\text{domain}} n_{\text{band}}^2 n_{\text{grid}})$  operations, where  $N_{\text{domain}}$  is the number of domains, and  $n_{\text{band}}$  and  $n_{\text{grid}}$  are the number of Kohn–Sham orbitals and that of real-space grid points per domain, respectively. Since we choose  $N_{\text{domain}} = O(N)$ , both  $n_{\text{band}}$  and  $n_{\text{grid}}$  are  $O(n) = O(1)$ , where  $n = N/N_{\text{domain}}$  is the average number

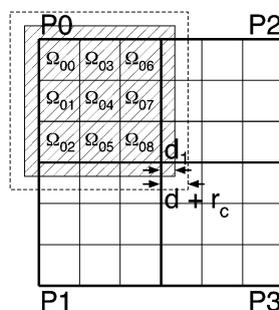


Fig. 3. Schematic of the parallel divide-and-conquer algorithm in 2D. The physical system is divided into subsystems,  $P_0, \dots, P_3$ , of equal volume, and each subsystem is assigned to a processor in a parallel computer. Each subsystem, in turn, consists of multiple non-overlapping core domains,  $\Omega_{00}, \dots, \Omega_{08}$ . To perform electronic-structure calculations on overlapping domains,  $\Omega_0, \dots, \Omega_8$ , on processor  $P_0$ , the contributions to the total electron density,  $\rho(\vec{r})$ , within the primary buffer-layer depth  $d_1$  (see the hatched area), need to be cached from the nearest-neighbor processors. In addition, the ionic positions within depth  $d + r_c$  (enclosed by dashed lines) need to be cached from the nearest-neighbor processors to compute ionic pseudopotentials. Here,  $d$  and  $r_c$  are the total buffer-layer depth and the range of nonlocal pseudopotentials, respectively.

of atoms per domain. Consequently, this operation scales as  $O(Nn^3) = O(N)$ . All calculations in step (ii) except for the orthogonalization require  $O(N_{\text{domain}}n_{\text{band}}n_{\text{grid}}) = O(Nn^2) = O(N)$  operations. The orthogonalization step, which scales as  $O(Nn^3)$ , is negligible [29] for relatively small domains ( $n < 100$ ) used in our calculations.

### 3. Parallelization

The hierarchical grid EDC-DFT algorithm has been implemented on parallel computers based on spatial decomposition [3,29], in which the physical system is divided into subsystems of equal volume, and each subsystem is assigned to a processor in a parallel computer (see Fig. 3). The processors are logically arranged as a 3D mesh of size  $P = P_x \times P_y \times P_z$ . Each subsystem, in turn, consists of multiple domains.

In this parallelization scheme, each domain is local to a processor, and the Kohn–Sham orbitals,  $\{\psi_n^\alpha(\vec{r})\}$ , need not be exchanged between processors. Consequently, there is no need for the massive communication required for orthonormalization in conventional parallel DFT algorithms [29]. Instead, there are two major sources of interprocessor communications in the parallel EDC-DFT algorithm. First, for the computation of the total electron density,  $\rho(\vec{r})$ , in a given domain,  $\Omega_\alpha$ , partial contributions,  $\{\rho^{\alpha'}(\vec{r})\}$ , within the primary buffer-layer depth  $d_1$ , need to be cached from the nearest-neighbor processors,  $\alpha'$ . Second, the positions of ions within depth  $d + r_c$  need to be cached from the nearest-neighbor processors to compute ionic pseudopotentials. Here,  $d$  and  $r_c$  are the total buffer-layer depth and the range of nonlocal pseudopotentials, respectively.

The computational time of this parallel EDC-DFT algorithm scales as  $O((N_{\text{domain}}/P)n^3) = O((N/P)n^3)$ . The communication time for the above two tasks, on the other hand, scales as  $O((N/P)^{2/3})$ . The communication overhead of this algorithm,  $O((N/P)^{-1/3}n^{-3})$ , is extremely low, because of the small surface-to-volume ratio and the absence of communicating  $O(n)$  Kohn–Sham orbitals. The parallel multigrid algorithm to compute the Hartree potential from  $\rho(\vec{r})$  requires additional interprocessor communications, which scales as  $O(\log P)$  [51]. For coarse-grained applications, in which  $N_{\text{grid}}/P \gg P$ , however, this logarithmic overhead is negligible.

### 4. Numerical results

We have tested the convergence properties and scalability of the parallel EDC-DFT algorithms, as well as its applicability to MD simulations, for a number of materials, including amorphous cadmium selenide (CdSe) and

liquid rubidium (Rb). For these numerical tests, we have used periodic boundary conditions to obtain local Kohn–Sham orbitals, because a faster convergence has been achieved compared with the rigid wall boundary conditions. For the support function, we use a cubic interpolation function such that both the function value and its derivative is continuous at  $p^\alpha(\vec{r}) = 0$  and 1. Since we discard the wave functions within the secondary buffer layers, numerical results are likely to be insensitive to the choice of the support function. The program is written in Fortran 90 with MPI (Message Passing Interface) for message passing.

Numerical tests have been performed on an IBM SP4 computer at the Naval Oceanographic Office (NAVO) Major Shared Resource Center and Intel Xeon-based Linux clusters at the Collaboratory for Advanced Computing and Simulations (CACs) and the Research Computing Facility (RCF) at the University of Southern California (USC). The 1184-processor SP4 consists of 148 nodes or LPARs (Logical Partitions), each containing eight 1.3 GHz POWER4 processors. (Four LPARs, i.e. 32 processors, in turn constitute a physical unit.) An LPAR contains four dual-processor chips, which share 8 GB of memory. In each chip, the two processors share 1.4 MB L2 cache. The SP4 uses a proprietary network and IBM’s Colony II switch to communicate between nodes. The network switch provides 2.88 Gb/s bi-directional bandwidth, with 21  $\mu$ s latency. The SP4 runs the AIX 5.1 operating system. The 266-processor CACS Linux cluster consists of 133 nodes, with two 2.8 GHz Intel Xeon processors and 2GB DDR memory per node, 512 KB L2 cache per processor, and 533 MHz bus. These nodes are interconnected by Myricom’s Myrinet with 4 Gb/s of point-to-point bandwidth in the duplex mode and 7–9  $\mu$ s latency. The CACS cluster runs Red Hat Linux 8.0 operating system, Myricom GM network software, MPICH-GM for message passing, and Intel Fortran compiler. The 1976-processor RCF Linux clusters consists of Intel Xeon and Pentium III processors, including 464 3.06 GHz Xeon processors, connected by Myrinet. The RCF cluster also runs Red Hat Linux and Intel Fortran.

We first test the convergence of the EDC-DFT algorithm, with respect to the domain size. Fig. 4 shows the potential energy as a function of the domain size for an amorphous CdSe system containing 512 atoms in a cubic cell of length, 45.664 a.u. The amorphous configuration is prepared by a melt-quench procedure [52] in MD simulation based on an empirical interatomic potential. Here, and in the following, the grid spacing is chosen as 0.476 a.u. The total buffer size,  $d$ , is fixed as 2.854 a.u. In all the numerical tests, the primary buffer size is chosen as  $d_1 = d/2$ . The potential energy converges within 0.003 a.u. per atom above the domain size 6 a.u. The number of self-consistent iterations required for convergence decreases with increased domain size.

Next, we test the convergence of the potential energy as a function of the buffer size,  $d$ , for an amorphous CdSe system containing 512 atoms in a cubic cell of length, 45.664 a.u. (see Fig. 5). The domain size is fixed

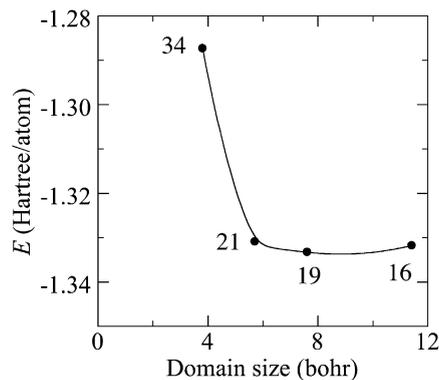


Fig. 4. Potential energy per atom as a function of the domain size for an amorphous CdSe system (512 atoms in a cubic cell of side length, 45.664 a.u.). The buffer size is fixed as 2.854 a.u. The atomic units are used for both energy and length. Numerals in the figure indicate the number of self-consistent iterations required for the convergence of the electron density within  $\langle\langle(\rho_i(\vec{r}) - \rho_{i-1}(\vec{r}))/\rho_0\rangle\rangle^2 \leq 10^{-4}$ , where  $\rho_i(\vec{r})$  is the electron density at  $i$ th iteration,  $\rho_0$  is the average electron density, 0.0215 a.u., and the brackets denote the average over the grid in the entire system.

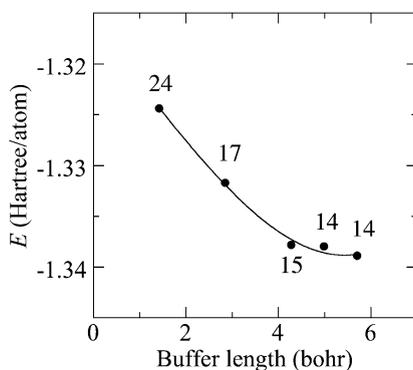


Fig. 5. Potential energy as a function of the buffer length,  $d$ , for an amorphous CdSe system (512 atoms in a cubic cell of side length, 45.664 a.u.). The domain size is fixed as 11.416 a.u. The atomic units are used for both energy and length. Numerals in the figure indicate the number of self-consistent iterations required for the convergence of the electron density within  $\langle \{(\rho_i(\vec{r}) - \rho_{i-1}(\vec{r})) / \rho_0\}^2 \rangle \leq 10^{-4}$ , where  $\rho_i(\vec{r})$  is the electron density at  $i$ th iteration,  $\rho_0$  is the average electron density, 0.0215 a.u., and the brackets denote the average over the grid for the entire system.

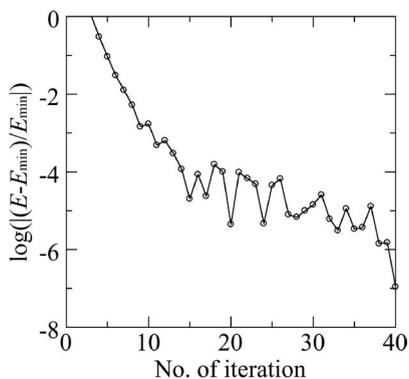


Fig. 6. The potential energy as a function of the number of self-consistent iterations for an amorphous CdSe system (32,768 atoms in a cubic cell of side length, 182.656 a.u.), where  $E_{\min}$  denotes the converged energy (or the potential energy obtained at the 50th iteration). The domain and buffer sizes are 11.416 and 5.708 a.u., respectively. The atomic units are used for both energy and length. The calculation was performed on 512 IBM POWER4 processors.

as 11.416 a.u. The potential energy converges within 0.001 a.u. per atom above  $d = 4$  a.u. The number of self-consistent iterations required for convergence decreases with increased buffer length.

To test the convergence of the EDC-DFT algorithm for larger systems with irregular atomic distributions, Fig. 6 shows the potential energy as a function of the number of self-consistent iterations for an amorphous CdSe system containing 32,768 atoms in a cubic cell of length, 182.656 a.u. In Fig. 6,  $E_{\min}$  denotes the converged energy (or the potential energy obtained at the 50th iteration). The domain and buffer sizes are 11.416 and 5.708 a.u., respectively. This result shows that the algorithm is convergent for disordered  $10^4$ -atom systems with nontrivial electronic structures. This test has been performed on 512 IBM POWER4 processors.

Although convergent for the 32,768-atom amorphous CdSe, the convergence rate is slower than those for smaller systems. Fig. 7 compares the convergence of the electron density for 32,768-atom and 512-atom amorphous CdSe systems. The left panel shows the convergence of the electron density in the self-consistent iterations for an amorphous CdSe system containing 32,768 atoms in a cubic cell of length, 182.656 a.u. Here,  $\rho_i(r)$  is the electron density at  $i$ th iteration, and  $\rho_0$  is the average electron density, 0.0215 a.u. The figure plots the electron density averaged over the grid in the entire system. The domain and buffer sizes are 11.416 and 5.708 a.u., respectively. The

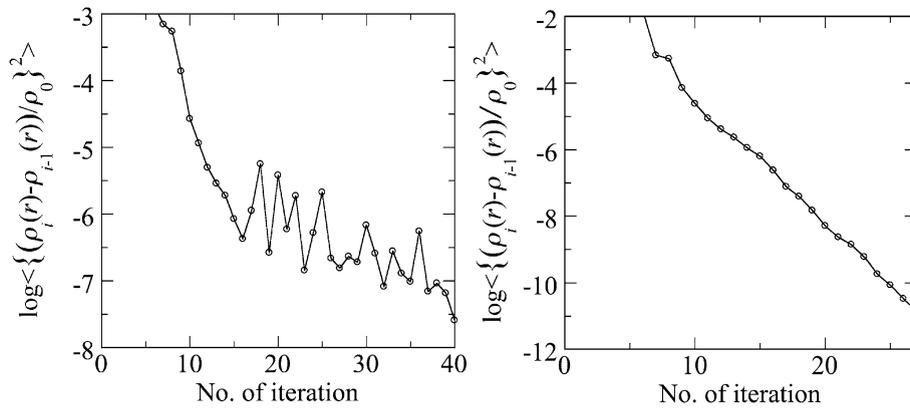


Fig. 7. (Left) Convergence of the electron density in the self-consistent iterations for an amorphous CdSe system (32,768 atoms in a cubic cell of side length, 182.656 a.u.).  $\rho_i(r)$  is the electron density at  $i$ th iteration,  $\rho_0$  is the average electron density, 0.0215 a.u., and the brackets denote the average over the grid in the entire system. The domain and buffer sizes are 11.416 and 5.708 a.u., respectively. (Right) Convergence of the electron density in the self-consistent iterations for an amorphous CdSe system (512 atoms in a cubic cell of side length, 45.664 a.u.). The domain and buffer sizes are the same as in the left panel.

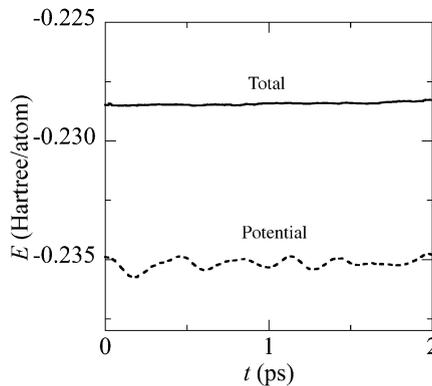


Fig. 8. Total (solid curve) and potential (dashed curve) energies in the atomic unit as a function of time in a molecular-dynamics simulation of liquid Rb at 1400 K (432 atoms in a cubic cell of side length, 65.696 a.u.). The domain and buffer sizes are 16.424 and 8.212 a.u., respectively.

right panel shows the same quantity for a 512-atom amorphous CdSe system in a cubic cell of length, 45.664 a.u. The convergence for the 32,768-atom system is slightly slower than for the 512-atom system. However, the convergence rate of the EDC-DFT algorithm is sufficiently high to allow practical electronic-structure calculations for disordered  $10^4$ -atom systems.

To test the applicability of the EDC-DFT algorithm for first principles MD simulations, Fig. 8 shows the total and potential energies as a function of time in an MD simulation of liquid Rb at 1400 K. The system contains 432 atoms in a cubic cell of length, 65.696 a.u. The domain and buffer sizes are 16.424 and 8.212 a.u., respectively. The time discretization unit is  $\Delta t = 4$  fs. The total energy is conserved within  $2 \times 10^{-4}$  a.u. per atom for 500 MD time steps. The figure demonstrates the applicability of the EDC-DFT algorithm for large-scale first principles MD simulations on parallel computers.

Finally, we test the scalability of the parallel EDC-DFT algorithm. Fig. 9 shows the wall-clock time per self-consistent iteration with scaled workloads—512  $P$  atom CdSe system on  $P$  processors of an Intel Xeon-based Linux cluster. By increasing the number of atoms linearly with the number of processors, the wall-clock time

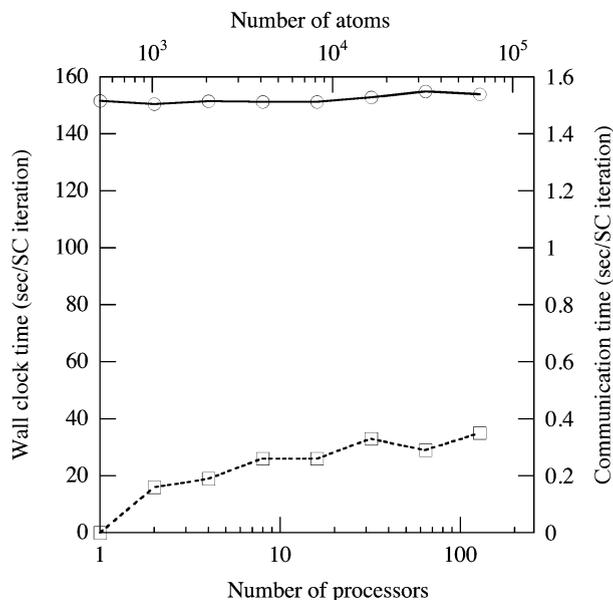


Fig. 9. Wall-clock (circles) and communication (squares) times per self-consistent iteration of the parallel EDC-DFT algorithm, with scaled workloads—512  $P$  atom CdSe system on  $P$  processors ( $P = 1, \dots, 128$ ) on an Intel Xeon-based Linux cluster.

increases little, indicating an excellent scalability. To quantify the parallel efficiency, we first define the speed of the EDC-DFT algorithm as a product of the total number of atoms and the number of self-consistent iterations executed per 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 is the constant-grain speedup divided by  $P$ . For a 65,536-atom CdSe system, the parallel efficiency is 0.985 on 128 Intel Xeon processors. Fig. 9 also shows the communication time per self-consistent iterations as a function of  $P$ . The communication overhead increases little for larger numbers of processors, and is negligible compared with the computational time.

## 5. Summary

We have developed a linear-scaling algorithm to perform large-scale first principles MD simulations, in which interatomic forces are computed quantum mechanically in the framework of the DFT. The embedded divide-and-conquer algorithm computes the electronic structure in  $O(N)$  time and includes non-additive contribution to the kinetic energy. Electronic wave functions are represented on a real-space grid, which is augmented with coarse multigrids to accelerate the convergence of iterative solutions and adaptive fine grids around atoms to accurately calculate ionic pseudopotentials. Spatial decomposition is employed to implement the hierarchical-grid EDC-DFT algorithm on massively parallel computers. A converged solution is obtained for a 32,768-atom amorphous CdSe system on 512 IBM Power 4 processors, demonstrating the applicability of the EDC-DFT algorithm for large-scale electronic structure calculations with irregular atomic distributions. The total energy is well conserved during MD simulations of liquid Rb, showing the applicability of this algorithm for first principles MD simulations. The parallel EDC-DFT algorithm exhibits an excellent scalability: parallel efficiency is 0.985 on 128 Intel Xeon processors for a 65,536-atom CdSe system. Such  $O(N)$  algorithms are indispensable for application software to keep up with the Moore's law of computer hardware, i.e. compute power of a processor doubles every 18 months.

## Acknowledgements

This work was partially supported by AFOSR–DURINT, ARO–MURI, DARPA–PROM, DOE, NSF, and Okawa Foundation. Numerical tests were performed at Department of Defense’s Major Shared Resource Centers under Challenge and CHSSI projects, and at the University of Southern California using the 1976-processor Linux cluster at the Research Computing Facility and the 432-processor Linux clusters at the Collaboratory for Advanced Computing and Simulations.

## References

- [1] F.F. Abraham, R. Walkup, H.J. Gao, et al., *Proc. Natl Acad. Sci. USA* 99 (2002) 5777.
- [2] T.C. Germann, P.S. Lomdahl, *IEEE Comput. Sci. Engrg.* 1 (2) (1999) 10.
- [3] A. Nakano, R.K. Kalia, P. Vashishta, et al., *Sci. Programming* 10 (2002) 263.
- [4] R. Car, M. Parrinello, *Phys. Rev. Lett.* 55 (1985) 2471.
- [5] J.-L. Fattebert, F. Gygi, *Comput. Phys. Comm.* 162 (2004) 24.
- [6] P. Hohenberg, W. Kohn, *Phys. Rev.* 136 (1964) B864.
- [7] W. Kohn, L.J. Sham, *Phys. Rev.* 140 (1965) A1133.
- [8] W. Kohn, P. Vashishta, in: N.H. March, S. Lundqvist (Eds.), *Inhomogeneous Electron Gas*, Plenum, 1983, p. 79.
- [9] D.W. Brenner, D.H. Robertson, M.L. Elert, et al., *Phys. Rev. Lett.* 70 (1993) 2174.
- [10] A. Strachan, A.C.T. van Duin, D. Chakraborty, et al., *Phys. Rev. Lett.* 91 (2003) 098301.
- [11] S. Ogata, F. Shimojo, R.K. Kalia, et al., *J. Appl. Phys.* 95 (2004) 5316.
- [12] A.C.T. van Duin, S. Dasgupta, F. Lorant, et al., *J. Phys. Chem. A* 105 (2001) 9396.
- [13] A.K. Rappe, W.A. Goddard, *J. Phys. Chem. A* 95 (1991) 3358.
- [14] F.H. Streitz, J.W. Mintmire, *Phys. Rev. B* 50 (1994) 11996.
- [15] S.W. Rick, S.J. Stuart, B.J. Berne, *J. Chem. Phys.* 101 (1994) 6141.
- [16] A. Nakano, *Comput. Phys. Comm.* 104 (1997) 59.
- [17] T. Campbell, R.K. Kalia, A. Nakano, et al., *Phys. Rev. Lett.* 82 (1999) 4866.
- [18] J. Tersoff, *Phys. Rev. B* 37 (1988) 6991.
- [19] D.W. Brenner, *Phys. Stat. Sol. (b)* 217 (2000) 23.
- [20] H. Kikuchi, R.K. Kalia, A. Nakano, et al., in: *Proc. Supercomputing '02, IEEE*, 2002.
- [21] S. Ogata, E. Lidorikis, F. Shimojo, et al., *Comput. Phys. Comm.* 138 (2001) 143.
- [22] S. Ogata, F. Shimojo, R.K. Kalia, et al., *Comput. Phys. Comm.* 149 (2002) 30.
- [23] W. Yang, *Phys. Rev. Lett.* 66 (1991) 1438.
- [24] W. Yang, T.-S. Lee, *J. Chem. Phys.* 103 (1995) 5674.
- [25] S.L. Dixon, K.M. Merz, *J. Chem. Phys.* 107 (1997) 879.
- [26] W. Kohn, *Phys. Rev. Lett.* 76 (1996) 3168.
- [27] S. Goedecker, *Rev. Mod. Phys.* 71 (1999) 1085.
- [28] F. Shimojo, T.J. Campbell, R.K. Kalia, et al., *Future Generation Comput. Syst.* 17 (2000) 279.
- [29] F. Shimojo, R.K. Kalia, A. Nakano, et al., *Comput. Phys. Comm.* 140 (2001) 303.
- [30] T. Wesolowski, A. Warshel, *J. Phys. Chem.* 98 (1994) 5183.
- [31] N. Govind, Y.A. Wang, E.A. Carter, *J. Chem. Phys.* 110 (1999) 7677.
- [32] J.R. Chelikowsky, N. Troullier, K. Wu, et al., *Phys. Rev. B* 50 (1994) 11355.
- [33] J.R. Chelikowsky, Y. Saad, S. Ögüt, et al., *Phys. Stat. Sol. (b)* 217 (2000) 173.
- [34] E.L. Briggs, D.J. Sullivan, J. Bernholc, *Phys. Rev. B* 54 (1996) 14362.
- [35] J.-L. Fattebert, J. Bernholc, *Phys. Rev. B* 62 (2000) 1713.
- [36] T.L. Beck, *Rev. Mod. Phys.* 72 (2000) 1041.
- [37] T. Ono, K. Hirose, *Phys. Rev. Lett.* 82 (1999) 5016.
- [38] L. Kleinman, D.M. Bylander, *Phys. Rev. Lett.* 48 (1982) 1425.
- [39] N. Troullier, J.L. Martins, *Phys. Rev. B* 43 (1991) 8861.
- [40] R.D. Kingsmith, M.C. Payne, J.S. Lin, *Phys. Rev. B* 44 (1991) 13063.
- [41] J.P. Perdew, K. Burke, M. Ernzerhof, *Phys. Rev. Lett.* 77 (1996) 3865.
- [42] D.E. Keyes, Y. Saad, D.G. Truhlar, *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, SIAM, Philadelphia, 1995.
- [43] L. Greengard, V. Rokhlin, *J. Comput. Phys.* 73 (1987) 325.

- [44] A. Nakano, R.K. Kalia, P. Vashishta, *Comput. Phys. Comm.* 83 (1994) 197.
- [45] S. Ogata, T.J. Campbell, R.K. Kalia, et al., *Comput. Phys. Comm.* 153 (2003) 445.
- [46] T. Darden, D. York, L. Pederson, *J. Chem. Phys.* 98 (1993) 10089.
- [47] R.K. Kalia, S.W. de Leeuw, A. Nakano, et al., *Comput. Phys. Comm.* 74 (1993) 316.
- [48] M.C. Payne, M.P. Teter, D.C. Allan, et al., *Rev. Mod. Phys.* 64 (1992) 1045.
- [49] W.H. Press, S.A. Teukolsky, W.T. Vetterling, et al., *Numerical Recipes*, second ed., Cambridge University Press, Cambridge, UK, 1992.
- [50] P. Pulay, *Chem. Phys. Lett.* 73 (1980) 393.
- [51] A. Nakano, P. Vashishta, R.K. Kalia, *Comput. Phys. Comm.* 83 (1994) 181.
- [52] J.P. Rino, I. Ebbsjo, P.S. Branicio, et al., *Phys. Rev. B* 70 (2004) 045207.