



Linear-scaling density-functional-theory calculations of electronic structure based on real-space grids: design, analysis, and scalability test of parallel algorithms

Fuyuki Shimojo¹, Rajiv K. Kalia, Aiichiro Nakano*, Priya Vashishta

Concurrent Computing Laboratory for Materials Simulations, Biological Computation and Visualization Center, Department of Physics and Astronomy, Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA

Received 10 April 2001; accepted 26 April 2001

Abstract

We have implemented parallel algorithms for density-functional-theory (DFT) based electronic-structure calculations. These include a plane-wave based algorithm, a real-space-grid algorithm based on a high-order finite difference method, and a linear-scaling real-space algorithm using localized orbitals. Parallelization schemes are described for these algorithms, and the computational complexity and the communications involved in the resulting parallel algorithms are analyzed. Scalability tests of these algorithms on massively parallel computers show that the linear-scaling DFT algorithm is highly scalable. For a 110,592-atom gallium arsenide system on 1024 IBM SP3 processors, the parallel efficiency is as high as 93%. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Parallel computing; Electronic structure; Density functional theory; Linear scaling algorithm

1. Introduction

Density-functional-theory (DFT) has reduced the exponentially complex quantum N -body problem to a self-consistent eigenvalue problem that can be solved with $O(N^3)$ operations [1,2]. Electronic-structure calculations based on DFT have been applied to various materials and have had tremendous impact on solid state physics, chemistry and other areas. In particular, Car and Parrinello [3] have combined DFT with molecular-dynamics (MD) simulation to obtain the phase space trajectories of the system (positions and velocities of all atoms at all time), thereby enabling first-principles study of dynamic processes in materials. In this approach, the electronic structure is calculated within the Born–Oppenheimer approximation, which separates ionic and electron degrees-of-freedom, and electron–ion interaction is calculated using the pseudopotential method [4,5]. The DFT-based MD approach not only obtains accurate interatomic forces from the Hellmann–Feynman theorem but also calculates electronic information such as charge distribution as a function of time.

* Corresponding author.

E-mail address: nakano@bit.csc.lsu.edu (A. Nakano).

¹ Also at Faculty of Integrated Arts and Sciences, Hiroshima University, Higashi-Hiroshima 739-8521, Japan.

In the calculation of electronic states of condensed matter, the plane-wave (PW) basis has been used most widely [6–13], because the formulation is simple and the atomic forces are easily calculated. In the PW algorithm, the fast Fourier transform (FFT) is usually used to calculate convolution. However, implementation of the FFT algorithm on parallel computers involves global operations, and hence its scalability becomes problematic on massively parallel computers. Because of the limited scalability, the number of atoms studied by the PW-based DFT algorithm has been limited to a few hundred. Chelikowsky, Troullier, and Saad [14–16] have developed a DFT algorithm that is more scalable on parallel computers. Their real-space-grid algorithm uses a high-order finite difference method to calculate derivatives such as the kinetic-energy operator. In this real-space (RS) algorithm, all operations are inherently short-ranged, resulting in superior scalability on parallel computers. As a result, the RS algorithm has made it possible to carry out DFT calculations involving ~ 1000 atoms [17]. Implementation of the real-space grid based DFT algorithm is discussed extensively in Refs. [18–25].

Because of the orthonormalization constraint on one-electron wave functions, computations in both PW and RS algorithms scale asymptotically as $O(N^3)$. The $O(N^3)$ orthonormalization procedure becomes a bottleneck for systems containing more than a few hundred atoms. To overcome this bottleneck, various linear-scaling DFT algorithms have been proposed recently [26–38]. These algorithms exploit the locality of physical quantities such as electronic wavefunctions and density matrices. In a linear-scaling (LS) DFT algorithm based on localized wave functions [29–32], each one-electron wave function is constrained in a local region, and orthogonalization of wave functions is avoided. For predictive materials simulations involving more than thousands of atoms, scalable and portable parallel implementation of DFT is indispensable.

In this paper, we develop a parallel $O(N)$ DFT algorithm based on localized wave functions combined with a real-space grid approach. We present parallelization strategies for the PW, RS, and LS algorithms for DFT calculations and analyze their scalability on parallel computers. Benchmark tests are performed on 1024 Cray T3E and IBM SP3 processors. This paper is organized as follows. In Section 2, we describe the DFT algorithm based on the real-space grid approach. Section 3 presents the linear scaling algorithm based on localized wave functions on a real-space grid. Parallel implementation is discussed for the real-space algorithms as well as for a PW method in Section 4. The results of numerical tests are given in Section 5, and Section 6 contains conclusions.

2. Real-space density-functional-theory calculations

In DFT, the energy of a system is expressed as a functional of N atomic positions, $\vec{r}^N = \{\vec{r}_I | I = 1, \dots, N\}$ and one-electron wave functions for N_{band} bands, $\psi^{N_{\text{band}}} = \{\psi_n(\vec{x}; \vec{r}^N) | n = 1, \dots, N_{\text{band}}\}$:

$$\begin{aligned}
 E(\vec{r}^N; \psi^{N_{\text{band}}}) = & 2 \sum_{n=1}^{N_{\text{band}}} \left\{ \int d\vec{x} \psi_n^*(\vec{x}; \vec{r}^N) \left(-\frac{\hbar^2}{2m_e} \frac{\partial^2}{\partial \vec{x} \partial \vec{x}} \right) \psi_n(\vec{x}; \vec{r}^N) \right. \\
 & \left. + \int d\vec{x} \psi_n^*(\vec{x}; \vec{r}^N) V_{\text{ion}}(\vec{x}) \psi_n(\vec{x}; \vec{r}^N) \right\} \\
 & + \frac{1}{2} \int d\vec{x} \rho(\vec{x}; \vec{r}^N) V_{\text{H}}(\vec{x}) + E_{\text{XC}}[\rho(\vec{x}; \vec{r}^N)] + \sum_{I=1}^{N-1} \sum_{J=I+1}^N \frac{Z_I Z_J e^2}{|\vec{r}_I - \vec{r}_J|}, \quad (1)
 \end{aligned}$$

where, m_e is the mass of an electron and Z_I is the valence of the I th ion. In Eq. (1), $V_{\text{ion}}(\vec{x})$, $V_{\text{H}}(\vec{x})$, and $E_{\text{XC}}(\rho)$ are the valence electron–ion interaction potential, Hartree potential, and exchange–correlation energy functional, respectively. The electron number density, $\rho(\vec{x}; \vec{r}^N)$, is calculated as

$$\rho(\vec{x}; \vec{r}^N) = 2 \sum_{n=1}^{N_{\text{band}}} |\psi_n(\vec{x}; \vec{r}^N)|^2. \quad (2)$$

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

$$\int d\vec{x} \psi_n^*(\vec{x}; \vec{r}^N) \psi_{n'}(\vec{x}; \vec{r}^N) = \delta_{nn'} \equiv \begin{cases} 1 & (n = n'), \\ 0 & (n \neq n'). \end{cases} \quad (3)$$

Under the orthonormality constraint, variation with respect to $\psi^{N_{\text{band}}}$ leads to the following Kohn–Sham equations,

$$\hat{H} \psi_n(\vec{x}; \vec{r}^N) = \varepsilon_n \psi_n(\vec{x}; \vec{r}^N) \quad (n = 1, \dots, N_{\text{band}}), \quad (4)$$

where ε_n is the eigenvalue for n th electronic state. In Eq. (4), the Hamiltonian operator, \hat{H} , is obtained as

$$\begin{aligned} \hat{H} \psi_n(\vec{x}; \vec{r}) &\equiv \frac{\delta E(\vec{r}^N; \psi^{N_{\text{band}}})}{\delta \psi_n^*(\vec{x}; \vec{r})} \\ &= \left\{ -\frac{\hbar^2}{2m_e} \frac{\partial^2}{\partial \vec{x} \partial \vec{x}} + V_{\text{ion}}(\vec{x}) + V_{\text{H}}(\vec{x}) + V_{\text{XC}}(\vec{x}) \right\} \psi_n(\vec{x}; \vec{r}^N), \end{aligned} \quad (5)$$

where $V_{\text{XC}}(\vec{x}) = \delta E_{\text{XC}}[\rho]/\delta \rho$ is the exchange-correlation potential. Using the eigenvalues, ε_n , the ground-state energy is expressed as

$$\begin{aligned} E(\vec{r}^N) &= \sum_n \varepsilon_n - \frac{1}{2} \int d\vec{x} \rho(\vec{x}; \vec{r}^N) V_{\text{H}}(\vec{x}) + E_{\text{XC}}[\rho] \\ &\quad - \int d\vec{x} \rho(\vec{x}; \vec{r}^N) V_{\text{XC}}(\vec{x}) + \sum_{I=1}^{N-1} \sum_{J=I+1}^N \frac{Z_I Z_J e^2}{|\vec{r}_I - \vec{r}_J|}, \end{aligned} \quad (6)$$

where the first term is the band-structure energy. Our DFT algorithms use either the local density approximation [39] or the generalized gradient approximation [40] for $E_{\text{XC}}(\rho)$ and norm-conserving pseudopotentials [5] for $V_{\text{ion}}(\vec{x})$ in a fully separable form [41].

Our DFT algorithms use a real-space grid approach combined with a high-order finite difference method for calculating the derivatives (i.e. kinetic-energy operator) in Eq. (1) [14–16]. In this algorithm, $\psi_n(\vec{x}; \vec{r}^N)$ and $\rho(\vec{x}; \vec{r}^N)$ are represented by numerical values on real space grid points. The kinetic energy operator is expanded using the finite difference method as

$$\left. \frac{\partial^2 \psi_n(\vec{x}; \vec{r}^N)}{\partial x^2} \right|_{\vec{x}=(x_i, y_j, z_k)} = \sum_{l=-L}^L C_l \psi_n(x_i + lh, y_j, z_k; \vec{r}^N) + O(h^{2L+2}), \quad (7)$$

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 suitable for localized configurations such as clusters. Since the finite difference expansion involves only short-ranged operations, an efficient implementation on parallel computers is possible.

The minimization of $E(\vec{r}^N; \psi^{N_{\text{band}}})$ with respect to $\psi^{N_{\text{band}}}$ is performed by searching for an energy-minimum iteratively based on the conjugate-gradient (CG) method [8,9]. The minimization loop consists of the following steps:

- (i) Calculate the electronic potentials, $V_{\text{ion}}(\vec{x})$, $V_{\text{H}}(\vec{x})$, and $V_{\text{XC}}(\vec{x})$, using $\rho(\vec{x}; \vec{r}^N)$ from the previous step or starting value.
- (ii) Perform a unitary transformation of $\psi^{N_{\text{band}}}$ to diagonalize the corresponding Hamiltonian matrix, $\int d\vec{x} \psi_n^* H \psi_m$.
- (iii) Improve $\{\psi_n(\vec{x}; \vec{r}^N) | n = 1, \dots, N_{\text{band}}\}$ iteratively.
- (iv) Calculate $\rho(\vec{x}; \vec{r}^N)$ using the updated $\psi^{N_{\text{band}}}$.

These steps, (i)–(iv), are repeated until the self-consistency between $\psi_n(\vec{x}; \vec{r}^N)$ and $\rho(\vec{x}; \vec{r}^N)$ is achieved within an error tolerance. The step (iii) 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 :

```

1  for  $n \leftarrow 1$  to  $N_{\text{band}}$  do
2     $i \leftarrow 1$ 
3    orthogonalize  $\psi_n^{(i)}$  to  $\psi_m$  ( $m = 1, 2, \dots, n - 1$ )
4    calculate  $\varepsilon_n^{(i)} \leftarrow \int d\vec{x} \psi_n^{(i)*} H \psi_n^{(i)}$ 
5    while  $\varepsilon_n^{(i)}$  is not convergent do
6      calculate the gradient,  $g_n^{(i)} \leftarrow (\varepsilon_n^{(i)} - H) \psi_n^{(i)}$ 
7      calculate the preconditioning vector,  $\tilde{g}_n^{(i)} \leftarrow \hat{P} g_n^{(i)}$ 
8      calculate the conjugate gradient,  $h_n^{(i)} \leftarrow \tilde{g}_n^{(i)} + \beta h_n^{(i-1)}$ ,
      where
          
$$\beta = \begin{cases} \int d\vec{x} g_n^{(i)*} g_n^{(i)} / \int d\vec{x} g_{n'}^{(i-1)*} g_{n'}^{(i-1)} & \text{for } i > 1, \\ 0 & \text{for } i = 1 \end{cases}$$

9      orthogonalize  $h_n^{(i)}$  to  $\psi_m$  ( $m = 1, 2, \dots, n - 1$ )
10     improve  $\psi_n^{(i+1)} \leftarrow C(\lambda)(\psi_n^{(i)} + \lambda h_n^{(i)})$  so as to minimize the band energy,
        where  $C(\lambda)$  is the normalization constant
11     calculate the new band energy,  $\varepsilon_n^{(i+1)} \leftarrow \int d\vec{x} \psi_n^{(i+1)*} \hat{H} \psi_n^{(i+1)}$ 
12      $i \leftarrow i + 1$ 

```

To accelerate the convergence, the gradients (residual vectors), $g_n(\vec{x})$, are preconditioned by taking an average over the neighbor grid points as

$$\begin{aligned} \tilde{g}_n(x_i, y_j, z_k) &= \hat{P} g_n(x_i, y_j, z_k) \\ &\equiv \sum_{l_1=-1}^1 \sum_{l_2=-1}^1 \sum_{l_3=-1}^1 d_{l_1 l_2 l_3} g_n(x_i + l_1 h, y_j + l_2 h, z_k + l_3 h), \end{aligned} \quad (8)$$

where $d_{l_1 l_2 l_3}$ are constants. This enhances the convergence rate of short-wavelength components of the residual.

To reduce long wavelength components of the residual, we use the multigrid method [20,21,23,42]. On a coarser grid, the following equations are solved,²

$$\left\{ -\frac{\hbar^2}{2m_e} \frac{\partial^2}{\partial \vec{x} \partial \vec{x}} + V(\vec{x}) \right\} \varphi_n(\vec{x}) = g_n(\vec{x}). \quad (9)$$

In this method, the electronic potential, $V(\vec{x})$, as well as the residual vector, $g_n(\vec{x})$, on a fine grid are restricted to a coarser grid by using restriction operations [42]. The solution, $\varphi_n(\vec{x})$, is prolonged to the fine grid, and is added to $\psi_n(\vec{x}; \vec{r}^N)$. In our algorithm, not only the Kohn–Sham equations but also the Poisson equations to obtain $V_H(\vec{x})$ are solved by the multigrid method. Recent developments involving the multigrid-accelerated solver for the Kohn–Sham equations are found in Refs. [23,43]. Wang and Beck [43] have demonstrated a higher-order multigrid scheme by adopting a nonlinear full approximation scheme with the full multigrid algorithm [42]. In their scheme, the convergence rate in the self-consistency iterations is relatively insensitive to the initial values of the wave functions and charge density. Starting with random orthonormalized functions, the ground-state electronic structure is obtained in 2 to 4 self-consistency iterations [43].

We analyze the computational complexity of the real-space (RS) DFT algorithm. Since the most time consuming part is step (iii), we concentrate on the computational complexity of this step. The orthogonalization in step (iii)

² In previous studies [20,21], only the kinetic energy operator (the first term of the left-hand side of Eq. (9)) has been taken into account for the multigrid procedure.

requires $O(N_{\text{band}}^2 N_{\text{grid}})$ operations, where N_{grid} is the number of real-space grid points. Since both N_{band} and N_{grid} are proportional to N , this operation scales as $O(N^3)$. All calculations except for the orthogonalization in step (iii) require $O(N_{\text{band}} N_{\text{grid}}) \sim O(N^2)$ operations. Although the orthogonalization step, which scales as $O(N^3)$, is negligible for relatively small systems ($N_{\text{band}} < 100$), it becomes the bottleneck for larger systems containing over 1000 electronic bands.

3. Linear-scaling real-space density-functional-theory calculations

In the conventional real-space DFT algorithm described in Section 2, we obtain a set of eigenfunctions of the Hamiltonian, which extend over the entire system. However, it is always possible to transform the eigenfunctions to a set of localized orbitals by a unitary transformation. If a localization constraint is imposed on electron orbitals, each orbital overlaps only with a finite number of neighboring orbitals, independent of the size of the system, thereby enabling a linear-scaling algorithm.

We use the energy-functional minimization approach proposed by Mauri and Galli [29], which requires neither explicit orthogonalization nor inversion of an overlap matrix. It can be proven that *unconstrained* minimization of the energy functional,

$$\tilde{E}(\vec{r}^N; \psi^{N_{\text{wf}}}) = 2 \sum_{m=1}^{N_{\text{wf}}} \sum_{n=1}^{N_{\text{wf}}} Q_{mn} (H_{nm} - \eta S_{nm}) + \eta N_{\text{el}}, \quad (10)$$

with

$$H_{mn} = \int d\vec{x} \psi_m^*(\vec{x}; \vec{r}^N) \hat{H} \psi_n(\vec{x}; \vec{r}^N), \quad (11)$$

$$S_{mn} = \int d\vec{x} \psi_m^*(\vec{x}; \vec{r}^N) \psi_n(\vec{x}; \vec{r}^N), \quad (12)$$

$$Q_{mn} = 2\delta_{mn} - S_{mn}, \quad (13)$$

produces the identical solution to that in constrained minimization of Eq. (1) [29–31]. It can also be proven that the minimized \tilde{E} coincides with the band structure energy in Eq. (6) [29–31]. The N_{wf} , η and N_{el} are the number of wave functions, the chemical potential and the number of electrons, respectively. In this algorithm, each wave function is localized in a local region and therefore interacts with only those wave functions in the neighboring local regions. The energy functional is minimized iteratively using the CG method, and the minimization loop consists of the following steps:

- (i) Calculate the electronic potentials, $V_{\text{ion}}(\vec{x})$, $V_{\text{H}}(\vec{x})$, and $V_{\text{XC}}(\vec{x})$, using $\rho(\vec{x}; \vec{r}^N)$ from previous step or starting value.
- (ii) Improve $\{\psi_n(\vec{x}; \vec{r}^N) | n = 1, \dots, N_{\text{wf}}\}$ iteratively.
- (iii) Calculate $\rho(\vec{x}; \vec{r}^N)$ using the updated $\psi^{N_{\text{wf}}}$.

In contrast to the conventional RS algorithm in Section 2, step (ii) in this linear-scaling (LS) algorithm includes one inner loop for the CG iteration (controlled by a loop index, i); all wave functions are improved concurrently in this algorithm rather than in a sequential manner as in the RS algorithm:

- 1 $i \leftarrow 1$
- 2 calculate $H_{mn}^{(i)}$, $S_{mn}^{(i)}$, and $\tilde{E}^{(i)}$
- 3 **while** the energy $\tilde{E}^{(i)}$ is not convergent **do**
- 4 calculate the gradient,

$$g_n^{(i)} \leftarrow -2 \sum_m [(\hat{H} - \eta) \psi_m^{(i)} Q_{mn}^{(i)} - \psi_m^{(i)} (H_{mn}^{(i)} - \eta S_{mn}^{(i)})] \quad (n = 1, \dots, N_{\text{wf}})$$

- 5 calculate the preconditioning vectors, $\tilde{g}_n^{(i)} \leftarrow \widehat{P} g_n^{(i)}$ ($n = 1, \dots, N_{\text{wf}}$)
 6 calculate the conjugate gradient, $h_n^{(i)} \leftarrow \tilde{g}_n^{(i)} + \beta h_n^{(i-1)}$ ($n = 1, \dots, N_{\text{wf}}$)
 where

$$\beta = \begin{cases} \frac{\sum_n \int d\vec{x} g_n^{(i)*} g_n^{(i)} / \sum_{n'} \int d\vec{x} g_{n'}^{(i-1)*} g_{n'}^{(i-1)}}{0} & \text{for } i > 1, \\ 0 & \text{for } i = 1 \end{cases}$$

- 7 improve $\psi_n^{(i+1)} \leftarrow \psi_n^{(i)} + \lambda h_n^{(i)}$ ($n = 1, \dots, N_{\text{wf}}$) so as to minimize the band structure energy
 8 calculate $H_{mn}^{(i+1)}$, $S_{mn}^{(i+1)}$, and $\tilde{E}^{(i+1)}$
 9 $i \leftarrow i + 1$

We now analyze computational complexity of the LS algorithm. All operations in step (ii) scale as $O(N_{\text{wf}} \tilde{N}_{\text{wf}} \tilde{N}_{\text{grid}})$, where \tilde{N}_{wf} is the number of wave functions interacting with each wave function, and \tilde{N}_{grid} is the number of real-space grid points in the local region. Note that both \tilde{N}_{wf} and \tilde{N}_{grid} are independent of the system size. Since the energy-minimization loop contains neither a subspace diagonalization nor the Gram–Schmidt orthonormalization, the computation scales as $O(N)$.

4. Design and analysis of parallel density-functional-theory algorithms

In this section, we describe the parallelization of the RS DFT algorithm (Section 2), the LS DFT algorithm (Section 3), as well as a PW-basis DFT algorithm.

As mentioned in Section 1, $\psi_n(\vec{x}; \vec{r}^N)$ is expanded in terms of plane waves in most DFT algorithms. To implement the PW method, the FFT is usually used to calculate convolutions [3,6–13]. There are at least two ways to parallelize the PW method [12,13]: The band decomposition and the reciprocal-space decomposition. In the band decomposition scheme, wave functions with different band indices are assigned to different processors. The CG iteration to improve the wave function for each band can be performed independently. However, there is a drawback in the memory requirement, since all processors must have the charge density of the entire system redundantly. Also, enormous communication is required to calculate the scalar products between the wave functions for the Gram–Schmidt orthonormalization, since entire components of each band have to be sent to all processors. On the other hand, in the reciprocal-space decomposition scheme, the reciprocal vectors of each band as well as the charge density are distributed to processors. Although none of the drawbacks in the band decomposition scheme mentioned above is involved, the reciprocal-space decomposition requires the FFT algorithm on parallel computers that involves complex communications with varying strides.

In this study, we combine these two schemes in a manner similar to the one used by Tuckerman et al. [13]. In this hybrid decomposition scheme, the CG iterations to improve the wave functions and the calculation of the charge density are parallelized using the band decomposition scheme. On modern computers with large memory, the redundant memory allocation imposes little problem. The Gram–Schmidt orthonormalization and the unitary transformation are executed in the reciprocal-space decomposition scheme. To switch between these two schemes, all-to-all communications are required. Let us denote the number of reciprocal vector components for each band as N_{PW} . The number, N_{PW} , scales linearly with N . Since each processor sends $N_{\text{PW}} N_{\text{band}} / P$ reciprocal-vector components to the other $(P - 1)$ processors and receives the same number of reciprocal-vector components from other processors, these all-to-all communications scale as $O(N^2)$. In addition, global communication is necessary to calculate the scalar products between the wave functions. The data size in the global communications scales as $O(N^2 \log P)$, since the total pair of wave functions is N_{band}^2 , and each global sum is performed in $\log P$ steps.

The parallelization of the real-space-grid-based algorithm is relatively simple compared to that of the PW algorithm. Spatial decomposition is used to parallelize both the RS and the LS DFT algorithms. Grid points

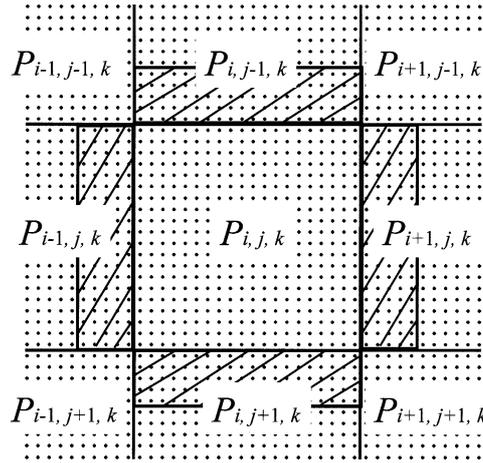


Fig. 1. Spatial decomposition scheme for the parallel implementation of the conventional RS DFT algorithm. The dots show the real-space grids. The $P_{i,j,k}$ denotes the (i, j, k) th processor for $i = 1 \dots I$, $j = 1 \dots J$, $k = 1 \dots K$ with IJK being the total number of processors. Each processor is assigned the corresponding sub-region. To perform the finite differencing operation in the $P_{i,j,k}$ processor, the grid data in the hatched regions have to be sent from the neighboring processors.

representing $\psi_n(\vec{x}; \vec{r}^N)$ and $\rho(\vec{x}; \vec{r}^N)$ are divided into spatial sub-regions, and are distributed among P processors. Finite differencing of $\psi_n(\vec{x}; \vec{r}^N)$ as well as the preconditioning operation in Eq. (8) necessitate interprocessor communication of the grid data because of non-local operations. (We use the sixth-order expansion formula for the kinetic-energy operator [15].) Message passing of grid data is required only between neighboring processors, in the same way as in the classical MD algorithm [44].

Fig. 1 illustrates the spatial decomposition scheme to parallelize the RS algorithm. Processors are arranged as a three-dimensional array, $\{P_{i,j,k} \mid 1 \leq i \leq I, 1 \leq j \leq J, 1 \leq k \leq K\}$, and each processor, $P_{i,j,k}$, is assigned the corresponding sub-region. To perform the finite differencing operation in the processor $P_{i,j,k}$, the grid data in the hatched regions in Fig. 1 have to be cached from the neighboring processors, which involves considerable communication that scales as $O(N(N/P)^{2/3})$. This communication overhead can be derived as follows: The number of surface grid points that must be cached to other processors is proportional to $(N/P)^{2/3}$, and the number of communication is αN_{band} , where α is the average number of the CG iteration for each band. In addition to the surface-data caching, global communication, which scales as $O(N^2 \log P)$, is necessary to evaluate the scalar products between different $\psi_n(\vec{x}; \vec{r}^N)$'s in the same way as in the PW method.

Although basic strategy for parallelizing the LS algorithm is similar to that for the RS algorithm, communication scales quite differently. Fig. 2 shows the spatial decomposition scheme for the parallel LS algorithm. The large circle denotes the local region for a wave function, which is divided into four regions I, II, III, and IV, and distributed to the corresponding processors. For the kinetic-energy operation in the region I, interprocessor communication of the grid data in the hatched regions is needed. In the parallel LS algorithm, data structures to store discretized $\psi_n(\vec{x}; \vec{r}^N)$ are the following arrays. We use a one-dimensional array, which has sufficient size to store the total number of grid points for $\psi_n(\vec{x}; \vec{r}^N)$ assigned to each processor. Pointers are used for each band to specify the starting and ending array indices for grid points stored in the array. The number of surface grid points per sub-region is proportional to $(N/P)^{2/3}$ in the same way as in the RS algorithm, while the number of wave functions associated with each grid point is independent of the system size. Therefore, the communication data size between neighboring nodes scales as $O((N/P)^{2/3})$, which is much smaller than the $O(N(N/P)^{2/3})$ communication in the RS algorithm. Global communication for calculating the scalar products between different $\psi_n(\vec{x}; \vec{r}^N)$'s, which scales as $N^2 \log P$ in the RS algorithm, is unnecessary in the LS algorithm.

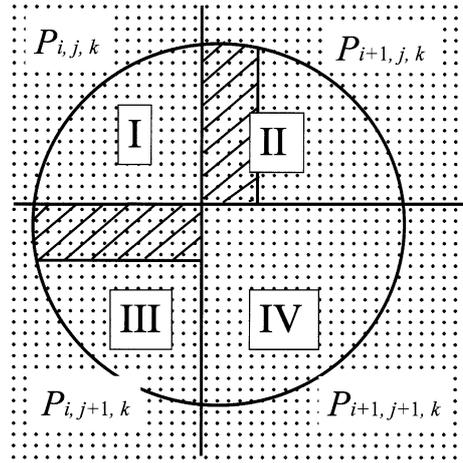


Fig. 2. Spatial decomposition scheme for the parallel LS method. The large circle shows the local region for a wave function, which is divided into four sub-regions I, II, III, and IV, and distributed to the corresponding processors. For the kinetic-energy operation in the sub-region I, the interprocessor communication of the grid data in the hatched regions is needed.

Table 1
Runtime (computation and communication times) and the memory requirement of the PW, RS, and LS algorithms for an N atom problem on P processors

	Computation	Interprocessor communication	Global communication	Memory requirement
PW	N^3/P	N^2	$N^2 \log P$	N^2
RS	N^3/P	$N(N/P)^{2/3}$	$N^2 \log P$	N^2/P
LS	N/P	$(N/P)^{2/3}$	–	N/P

Table 1 summarizes the numbers of computational operations and communications required in the PW, RS, and LS algorithms. Table 1 also shows the memory requirement of the three algorithms.

5. Performance tests

Benchmark tests of the three parallel DFT algorithms — PW, RS, and LS — have been performed on the Cray T3E and the IBM SP3 computers at the U.S. Naval Oceanographic Office (NAVO) Major Shared Resource Center using up to 1024 processors. All programs are written in Fortran 77 with MPI (Message Passing Interface) for message passing [45]. (As a compiler option, the -O3 optimization level is used.) The T3E at NAVO consists of 1088 Digital Alpha processors with clock speed 450 MHz and 258 GB memory. It runs UNICOS/mk 2.0.3 operating system and cf90 3.3.0 Fortran compiler. The IBM SP3 at NAVO is configured with 375 MHz Power3 CPUs and has 334 nodes with 4 CPUs and 4 GB of memory per node. It runs AIX 4.3 operating system and IBM XL Fortran 7.1 compiler.

We have performed electronic-structure calculations for gallium arsenide (GaAs) crystals in the zinc-blende structure [46]. The grid spacing is chosen as $a/24$ ($a = 5.6532 \text{ \AA}$ is the size of the zinc-blende cubic unit cell). This grid corresponds to a PW cutoff of 12.4 Ry.

First, we performed the DFT electronic-structure calculation with the PW basis, which was applied to a 216 atom GaAs system as shown in Fig. 3. The speedup for 256 processors reaches only 19 when compared with

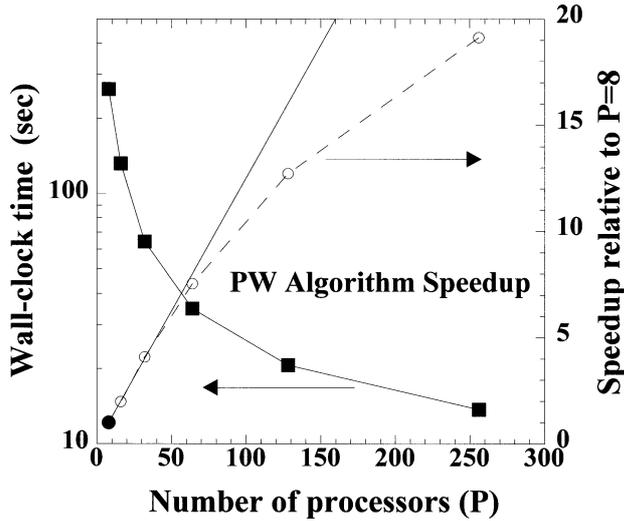


Fig. 3. Wall-clock time per self-consistent iteration (squares) and speed-up over 8 processors (circles) of the parallel PW algorithm as a function of the number of processors. The system is a 216-atom supercell of gallium arsenide. The straight solid line represents the perfect speed-up.

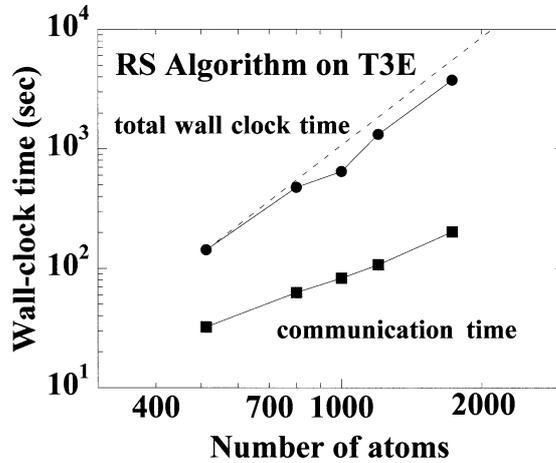


Fig. 4. Wall-clock (circles) and communication (squares) times per self-consistent iteration as a function of the number of atoms for the parallel RS algorithm on Cray T3E. The system is gallium arsenide crystal in the zinc-blende structure. The number of processors is 1024. The dashed line shows the $O(N^3)$ scaling. Communication time excludes the $O(N^2)$ -scaled global communication.

8 processors (the ideal value is 32). This implies computational efficiency of only 59% when the number of processors is increased from 8 to 256. Thus, the scalability of the PW method becomes problematic on massively parallel computers because of the global all-to-all communications. It is also difficult to scale up the problem size with the present parallel implementation, because each processor must store information on all the bands; see Table 1.

Fig. 4 shows the performance of the RS algorithm on 1024 T3E processors. The circles and squares are the wall-clock and interprocessor-communication times, respectively, per one self-consistent loop. (They are estimated assuming four CG iterations for each band.) The dashed line shows the $O(N^3)$ scaling. The asymptotic $O(N^3)$ behavior becomes dominant for $N > 1000$ in the total wall-clock time. The best fit of the measured data shows that

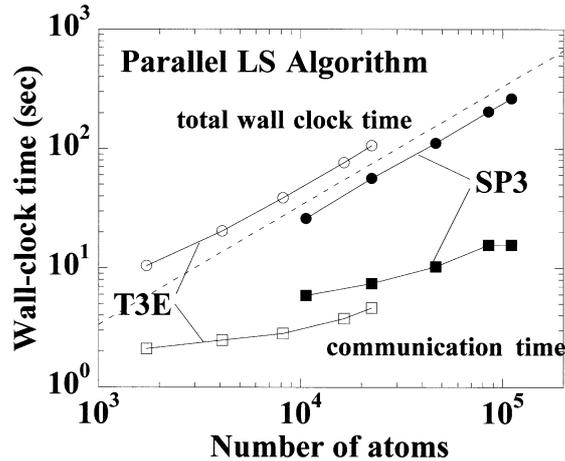


Fig. 5. Wall-clock (circles) and communication (squares) times per CG step as a function of the number of atoms for the parallel LS algorithm on Cray T3E and IBM SP3 computers. The system is gallium arsenide crystal in the zinc-blende structure. The number of processors is 1024. The dashed line shows the linear scaling.

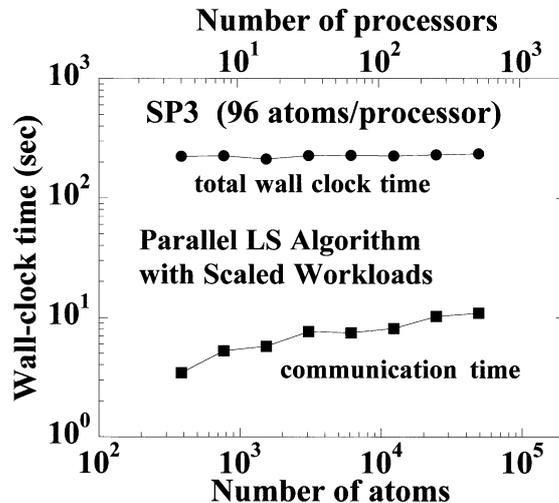


Fig. 6. Wall-clock (circles) and communication (squares) times of the parallel LS algorithm with scaled workloads — $96 \cdot P$ atom gallium arsenide systems on P processors ($P = 4, \dots, 1024$).

the interprocessor communication time scales with N as $O(N^{1.57})$, which agree well with the theoretical scaling, $O(N^{5/3})$, shown in Table 1.

In the LS calculations for GaAs systems, the localization region for the wave functions is defined as a sphere with radius 4.4 \AA , where the center of each localization region is placed at the center of a Ga–As bond. The performance of the LS algorithms on 1024 T3E and SP3 processors is shown in Fig. 5. The circles and squares are the wall-clock and interprocessor-communication times, respectively, for one CG iteration. The open and solid symbols denote the timings on T3E and SP3, respectively. The dashed line shows the linear scaling. The wall-clock time of the LS algorithm scales linearly with N above $N \sim 10,000$. For the largest system ($N = 110,592$), one CG iteration takes 280 s. For this system, the parallel efficiency is estimated to be 93%. The interprocessor communication time scales as $O(N^{0.6})$ for $N > 10,000$. When the number of atoms is less than ~ 4000 , the wall-clock time deviates from the

linear scaling because the communication time is not negligible compared with the computation time. Fig. 6 shows the wall-clock (circles) and communication (squares) times of the LS algorithm with scaled workloads — $96 \cdot P$ atom GaAs systems on P processors. The number of processors is varied from 4 to 1024. It is shown that the wall-clock time remains nearly constant, with the communication time increasing with P only slightly.

6. Conclusion

We have developed parallel density-functional-theory algorithms for electronic-structure calculations based on real-space grids and analyzed their scaling properties. Both the conventional and linear-scaling real-space algorithms scale superior to a plane-wave-based parallel DFT algorithm. In the conventional parallel real-space DFT method, where the number of operations is proportional to $O(N^3/P)$, the interprocessor communication scales as $O(N(N/P)^{2/3})$ and $O(N^2 \log P)$ global communication is involved. In the linear-scaling parallel DFT algorithm, where the number of operation is $O(N/P)$, interprocessor communication scales as $O((N/P)^{2/3})$. Scalability of both the conventional and the linear-scaling real-space DFT algorithms as well as that of a plane-wave based DFT algorithm have been tested for crystalline gallium arsenide. The parallel real-space DFT algorithms have been shown to be highly scalable. Using the linear-scaling real-space DFT algorithm, large electronic-structure calculation containing more than hundred thousand atoms are feasible.

Acknowledgements

This work is partially supported by AFOSR, DOE, NASA, NSF, and Louisiana Board of Regents. Benchmark tests were performed using the 1088-node Cray T3E and the 1280-processor IBM SP3 computers at Department of Defense's NAVO (Naval Oceanographic Office) Major Shared Resource Center under a DoD Challenge Project Award.

References

- [1] P. Hohenberg, W. Kohn, Phys. Rev. 136 (1964) B864;
W. Kohn, L.J. Sham, Phys. Rev. 140 (1965) A1133.
- [2] W. Kohn, P. Vashishta, in: S. Lundqvist, N.H. March (Eds.), *Inhomogeneous Electron Gas*, Plenum, New York, 1983, pp. 79–184.
- [3] R. Car, M. Parrinello, Phys. Rev. Lett. 55 (1985) 2471.
- [4] D.R. Hamann, M. Schlüter, C. Chiang, Phys. Rev. Lett. 43 (1979) 1494.
- [5] N. Troullier, J.L. Martins, Phys. Rev. B 43 (1991) 1993.
- [6] J. Ihm, A. Zunger, M.L. Cohen, J. Phys. C 12 (1979) 4409.
- [7] W.E. Pickett, Comput. Phys. Rep. 9 (1989) 115.
- [8] M.P. Teter, M.C. Payne, D.C. Allan, Phys. Rev. B 40 (1989) 12 255.
- [9] M.C. Payne, M.P. Teter, D.C. Allan, T.A. Arias, J.D. Joannopoulos, Rev. Mod. Phys. 64 (1992) 1045.
- [10] F. Shimojo, Y. Zempo, K. Hoshino, M. Watabe, Phys. Rev. B 52 (1995) 9320.
- [11] G. Kresse, J. Furthmuller, Phys. Rev. B 54 (1996) 11 169.
- [12] C. Cavazzoni, G.L. Chiarotti, Comput. Phys. Commun. 123 (1999) 56.
- [13] M.E. Tuckerman, D.A. Yarne, S.O. Samuelson, A.L. Hughes, G.J. Martyna, Comput. Phys. Commun. 128 (2000) 333.
- [14] J.R. Chelikowsky, N. Troullier, Y. Saad, Phys. Rev. Lett. 72 (1994) 1240.
- [15] J.R. Chelikowsky, N. Troullier, K. Wu, Y. Saad, Phys. Rev. B 50 (1994) 11 355.
- [16] J.R. Chelikowsky, N. Troullier, X. Jing, D. Dean, N. Binggeli, K. Wu, Y. Saad, Comput. Phys. Commun. 85 (1995) 325.
- [17] S. Ögüt, J.R. Chelikowsky, S.G. Louie, Phys. Rev. Lett. 79 (1997) 1770–1773.
- [18] T. Hoshi, M. Arai, T. Fujiwara, Phys. Rev. B 52 (1995) R5459.
- [19] A.P. Seitsonen, M.J. Puska, R.M. Nieminen, Phys. Rev. B 51 (1995) 14 057.
- [20] E.L. Briggs, D.J. Sullivan, J. Bernholc, Phys. Rev. B 52 (1995) R5471.
- [21] E.L. Briggs, D.J. Sullivan, J. Bernholc, Phys. Rev. B 54 (1996) 14 362.

- [22] J.R. Chelikowsky, *Phys. Rev. B* 57 (1998) 3333.
- [23] F. Ancilotto, P. Blandin, F. Toigo, *Phys. Rev. B* 59 (1999) 7868.
- [24] J.R. Chelikowsky, Y. Saad, S. Ögüt, I. Vashilev, A. Stathopoulos, *Phys. Stat. Sol.* 217 (2000) 173.
- [25] J.-L. Fattebert, J. Bernholc, *Phys. Rev. B* 63 (2000) 1713.
- [26] S. Goedecker, *Rev. Mod. Phys.* 71 (1999) 1085.
- [27] P. Ordejón, *Comput. Mat. Sci.* 12 (1998) 157.
- [28] G. Galli, M. Parrinello, *Phys. Rev. Lett.* 69 (1992) 3547.
- [29] F. Mauri, G. Galli, *Phys. Rev. B* 50 (1994) 4316.
- [30] J. Kim, F. Mauri, G. Galli, *Phys. Rev. B* 52 (1995) 1640.
- [31] P. Ordejón, D.A. Drabold, R.M. Martin, M.P. Grumbach, *Phys. Rev. B* 51 (1995) 1456.
- [32] S. Itoh, P. Ordejón, R.M. Martin, *Comput. Phys. Commun.* 88 (1995) 173.
- [33] X.-P. Li, R.W. Nunes, D. Vanderbilt, *Phys. Rev. B* 47 (1993) 10 891.
- [34] C.M. Goringe, E. Hernández, M.J. Gillan, I.J. Bush, *Comput. Phys. Commun.* 102 (1997) 1.
- [35] P.D. Haynes, M.C. Payne, *Comput. Phys. Commun.* 102 (1997) 17.
- [36] D.R. Bowler, M.J. Gillan, *Comput. Phys. Commun.* 112 (1998) 103.
- [37] U. Stephan, *Phys. Rev. B* 62 (2000) 16 412.
- [38] E.B. Stechel, in: D.E. Keyes, Y. Saad, D.G. Truhlar (Eds.), *Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, SIAM, Philadelphia, 1995, p. 217.
- [39] D.M. Ceperley, B.J. Alder, *Phys. Rev. Lett.* 45 (1980) 566;
J.P. Perdew, A. Zunger, *Phys. Rev. B* 23 (1981) 5048.
- [40] J.P. Perdew, K. Burke, M. Ernzerhof, *Phys. Rev. Lett.* 77 (1996) 3865.
- [41] L. Kleinman, D.M. Bylander, *Phys. Rev. Lett.* 48 (1982) 1425.
- [42] A. Brandt, *Math. Comp.* 31 (1977) 333.
- [43] J. Wang, T.L. Beck, *J. Chem. Phys.* 112 (2000) 9223.
- [44] D.C. Rapaport, *The Art of Molecular Dynamics Simulation*, Cambridge Univ. Press, Cambridge, 1995.
- [45] W. Gropp, E. Lusk, A. Skjellum, *Using MPI*, MPI Press, Cambridge, MA, 1994.
- [46] M.L. Cohen, J.R. Chelikowsky, *Electronic Structure and Optical Properties of Semiconductors*, Springer, Berlin, 1988.