

Efficient low-order scaling method for large-scale electronic structure calculations with localized basis functions

Taisuke Ozaki

Research Center for Integrated Science (RCIS), Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan

(Received 9 May 2010; revised manuscript received 5 July 2010; published 20 August 2010)

An efficient low-order scaling method is presented for large-scale electronic structure calculations based on the density-functional theory using localized basis functions, which directly computes selected elements of the density matrix by a contour integration of the Green's function evaluated with a nested dissection approach for resultant sparse matrices. The computational effort of the method scales as $O[N(\log_2 N)^2]$, $O(N^2)$, and $O(N^{7/3})$ for one-, two-, and three-dimensional systems, respectively, where N is the number of basis functions. Unlike $O(N)$ methods developed so far the approach is a numerically exact alternative to conventional $O(N^3)$ diagonalization schemes in spite of the low-order scaling, and can be applicable to not only insulating but also metallic systems in a single framework. It is also demonstrated that the well separated data structure is suitable for the massively parallel computation, which enables us to extend the applicability of density-functional calculations for large-scale systems together with the low-order scaling.

DOI: [10.1103/PhysRevB.82.075131](https://doi.org/10.1103/PhysRevB.82.075131)

PACS number(s): 71.15.Mb, 71.15.Nc

I. INTRODUCTION

During the last three decades continuous efforts^{1–29} have been devoted to extend applicability of the density-functional theory (DFT) (Refs. 30 and 31) to large-scale systems, which leads to realization of more realistic simulations being close to experimental conditions. In fact, lots of large-scale DFT calculations have already contributed for comprehensive understanding of a vast range of materials,^{32–37} although widely used functionals such as local-density approximation (LDA) (Ref. 38) and generalized gradient approximation (GGA) (Ref. 39) have limitation in describing strong correlation in transition-metal oxides and van der Waals interaction in biological systems.

The efficient methods developed so far within the conventional DFT can be classified into two categories in terms of computational complexity^{1–26} while the other methods, which deviate from the classification, have been also proposed.^{27–29} The first category consists of $O(N^3)$ methods,^{1–10} where N is the number of basis functions, as typified by the Householder-QR method,^{9,10} the conjugate gradient method,^{2,6,7} and the Pulay method,^{4,5} which have currently become standard methods. The methods can be regarded as numerically exact methods, and the computational cost scales as $O(N^3)$ even if only valence states are calculated because of the orthonormalization process. On the other hand, the second category involves approximate $O(N)$ methods such as the density-matrix methods,^{19–21} the orbital minimization methods,^{18,23} and the Krylov subspace methods^{14–17,25} of which computational cost is proportional to the number of basis functions N . The linear scaling of the computational effort in the $O(N)$ methods can be achieved by introducing various approximations like the truncation of the density matrix¹⁹ or Wannier functions^{18,23} in real space. Although the $O(N)$ methods have been proven to be very efficient, the applications must be performed with careful consideration due to the introduction of the approximations, which might be one of reasons that the $O(N)$ methods have

not been widely used compared to the $O(N^3)$ methods. From the above reason one may think of whether a numerically exact but low-order scaling method can be developed by utilizing the resultant sparse structure of the Hamiltonian and overlap matrices expressed by localized basis functions. Recently, a direction toward the development of $O(N^{2\sim})$ methods has been suggested by Lin *et al.*⁴⁰ in which diagonal elements of the density matrix is computed by a contour integration of the Green's function calculated by making full use of the sparse structure of the matrix. Also, efficient schemes have been developed to calculate certain elements of the Green's function for electronic transport calculations,^{41,42} which are based on the algorithm by Takahashi *et al.*⁴³ and Erisman and Tinney.⁴⁴ However, except for the methods mentioned above the development of numerically exact $O(N^{2\sim})$ methods, which are positioned in between the $O(N)$ and $O(N^3)$ methods, has been rarely explored yet for large-scale DFT calculations.⁴⁵

In this paper we present a numerically exact but low-order scaling method for large-scale DFT calculations of insulators and metals using localized basis functions such as pseudoatomic orbital (PAO),⁴⁶ finite element (FE),⁴⁷ and wavelet basis functions.⁴⁸ The computational effort of the method scales as $O[N(\log_2 N)^2]$, $O(N^2)$, and $O(N^{7/3})$ for one-, two-, and three-dimensional (1D, 2D, and 3D) systems, respectively. In spite of the low-order scaling, the method is a numerically exact alternative to the conventional $O(N^3)$ methods. The key idea of the method is to directly compute selected elements of the density matrix by a contour integration of the Green's function evaluated with a set of recurrence formulas. It is shown that a contour integration method based on a continued fraction representation of the Fermi-Dirac function⁴⁹ can be successfully employed for the purpose, and that the number of poles used in the contour integration does not depend on the size of the system. We also derive a set of recurrence formulas based on the nested dissection⁵⁰ of the sparse matrix and a block LDL^T factorization using the Schur complement¹⁰ to calculate selected ele-

ments of the Green's function. The computational complexity is governed by the calculation of the Green's function. In addition to the low-order scaling, the method can be particularly advantageous to the massively parallel computation because of the well separated data structure.

This paper is organized as follows: in Sec. II the theory of the proposed method is presented together with detailed analysis of the computational complexity. In Sec. III several numerical calculations are shown to illustrate practical aspects of the method within a model Hamiltonian and DFT calculations using the PAO basis functions. In Sec. IV we summarize the theory and applicability of the numerically exact but low-order scaling method.

II. THEORY

A. Density-matrix approach

Let us assume that the Kohn-Sham (KS) orbital ϕ_ν is expressed by a linear combination of localized basis functions $\{\chi\}$ such as PAO,⁴⁶ FE,⁴⁷ and wavelet basis functions⁴⁸ as

$$\phi_\nu(\mathbf{r}) = \sum_{i=1}^N c_{\nu i} \chi_i(\mathbf{r}), \quad (1)$$

where N is the number of basis functions. Throughout the paper, we consider the spin restricted and \mathbf{k} -independent KS orbitals for simplicity of notation. However, the generalization of our discussion for these cases is straightforward. By introducing LDA or GGA for the exchange-correlation functional, the KS equation is written in a sparse matrix form

$$Hc_\nu = \varepsilon_\nu S c_\nu, \quad (2)$$

where ε_ν is the eigenvalue of state ν , c_ν a vector consisting of coefficients $\{c_{\nu i}\}$, and H and S are the Hamiltonian and overlap matrices, respectively. Due to both the locality of basis functions and LDA or GGA for the exchange-correlation functional, both the matrices possess the same sparse structure. It is also noted that the charge density $n(\mathbf{r})$ can be calculated by the density matrix ρ ,

$$n(\mathbf{r}) = \sum_{i,j} \rho_{ij} \chi_j(\mathbf{r}) \chi_i(\mathbf{r}). \quad (3)$$

By remembering that χ is localized in real space, one may notice that the product $\chi_i \chi_j$ is nonzero only if they are closely located each other. Thus, the number of elements in the density matrix required to calculate the charge-density scales as $O(N)$. As well as the calculation of the charge density, the total energy is computed by only the corresponding elements of the density matrix within the conventional DFT as

$$E_{\text{tot}}[n, \rho] = \text{Tr}(\rho H_{\text{kin}}) + \int d\mathbf{r} n(\mathbf{r}) v_{\text{ext}}(\mathbf{r}) + \int \int d\mathbf{r} d\mathbf{r}' \frac{n(\mathbf{r}) n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + E_{\text{xc}}[n], \quad (4)$$

where H_{kin} is the matrix for the kinetic operator, v_{ext} an ex-

ternal potential, and E_{xc} an exchange-correlation functional. Since the matrix H_{kin} possesses the same sparse structure as that of S , one may find an alternative way that the selected elements of the density matrix, corresponding to the nonzero products $\chi_i \chi_j$, are directly computed without evaluating the KS orbitals. The alternative way enables us to avoid an orthogonalization process such as Gram-Schmidt method for the KS orbitals, of which computational effort scales as $O(N^3)$ even if only the occupied states are taken into account. The direct evaluation of the selected elements in the density matrix is the starting point of the method proposed in the paper. In this sense, the low-order scaling method is similar to $O(N)$ Green's-function methods such as recursion and bond-order potential methods¹⁴⁻¹⁶ while the Green's function is approximately evaluated in these $O(N)$ methods. The density matrix ρ can be calculated through the Green's function G as follows:

$$\rho = -\frac{2}{\pi} \text{Im} \int_{-\infty}^{\infty} dE G(E + i0^+) f\left(\frac{E - \mu}{k_B T}\right), \quad (5)$$

where the factor 2 is due to the spin degeneracy, f the Fermi-Dirac function, μ chemical potential, T electronic temperature, k_B the Boltzmann factor, and 0^+ a positive infinitesimal. Also the matrix expression of the Green's function is given by

$$G(Z) = (ZS - H)^{-1}, \quad (6)$$

where Z is a complex number. Therefore, from Eqs. (5) and (6), our problem is cast to two issues: (i) how the integration of the Green's function can be efficiently performed and (ii) how the selected elements of the Green's function in the matrix form can be efficiently evaluated. In the subsequent sections we discuss the two issues in detail.

B. Contour integration of the Green's function

We perform the integration of the Green's function, Eq. (5), by a contour integration method using a continued fraction representation of the Fermi-Dirac function.⁴⁹ In the contour integration the Fermi-Dirac function is expressed by

$$\frac{1}{1 + \exp(x)} = \frac{1}{2} - \frac{\frac{x}{4}}{1 + \frac{\left(\frac{x}{2}\right)^2}{3 + \frac{\left(\frac{x}{2}\right)^2}{5 + \frac{\left(\frac{x}{2}\right)^2}{\dots}}}} \quad (7)$$

$$= \frac{1}{2} + \sum_{p=1}^{\infty} \frac{R_p}{x - iz_p} + \sum_{p=1}^{\infty} \frac{R_p}{x + iz_p},$$

where $x = \beta(Z - \mu)$ with $\beta = \frac{1}{k_B T}$, z_p , and R_p are poles of the continued fraction representation and the associated residues,

respectively. The representation of the Fermi-Dirac function is derived from a hypergeometric function and can be regarded as a Padé approximant when terminated at the finite continued fraction. The poles z_p and residues R_p can be easily obtained by solving an eigenvalue problem as shown in Ref. 49. By making use of the expression of Eq. (7) for Eq. (5) and considering the contour integration, one obtain the following expression for the integration of Eq. (5):

$$\rho = M^{(0)} + \text{Im} \left[-\frac{4i}{\beta} \sum_{p=1}^{\infty} G(\alpha_p) R_p \right], \quad (8)$$

where $\alpha_p = \mu + i\frac{z_p}{\beta}$ and $M^{(0)}$ is the zeroth-order moment of the Green's function which can be computed by $iR_d G(iR_d)$ with a large real number R_d of 10^{10} hartree. The structure of the poles distribution, that all the poles are located on the imaginary axis like the Matsubara pole but the density of the poles becomes smaller as the poles go away from the real axis, has been found to be very effective for the efficient integration of Eq. (5). It has been shown that only the use of the 100 poles at 600 K gives numerically exact results within double precision.⁴⁹ Thus, the contour integration method can be regarded as a numerically exact method even if the summation is terminated at a practically modest number of poles.

Moreover, it should be noted that the number of poles to achieve convergence is independent of the size of system. Giving the Green's function in the Lehmann representation, Eq. (8) can be rewritten by

$$\begin{aligned} \rho &= M^{(0)} + \text{Im} \left(-\frac{4i}{\beta} \sum_{p=1}^{\infty} \sum_{\nu} \frac{|\phi_{\nu}\rangle\langle\phi_{\nu}|}{\alpha_p - \varepsilon_{\nu}} R_p \right) \\ &= M^{(0)} + \sum_{\nu} \text{Im} \left(-\frac{4i}{\beta} \sum_{p=1}^{\infty} \frac{|\phi_{\nu}\rangle\langle\phi_{\nu}|}{\alpha_p - \varepsilon_{\nu}} R_p \right). \end{aligned} \quad (9)$$

Although the expression in the second line is obtained by just exchanging the order of the two summations, the expression clearly shows that the number of poles for convergence does not depend on the size of system if the spectrum radius is independent of the size of system. Since the independence of the spectrum radius can be found in general cases, it can be concluded that the computational effort is determined by that for the calculation of the Green's function.

The energy density matrix e , which is needed to calculate forces on atoms within nonorthogonal localized basis functions, can also be calculated by the contour integration method⁴⁹ as follows:

$$\begin{aligned} e &= -\frac{2}{\pi} \text{Im} \int_{-\infty}^{\infty} dE E G(E + i0^+) f\left(\frac{E - \mu}{k_B T}\right), \\ &= M^{(1)} + \kappa M^{(0)} + \text{Im} \left[-\frac{4i}{\beta} \sum_{p=1}^{\infty} G(\alpha_p) R_p \alpha_p \right] \end{aligned} \quad (10)$$

with κ defined by

$$\kappa = \frac{4}{\beta} \sum_{p=1}^{\infty} R_p, \quad (11)$$

where $M^{(0)}$ and $M^{(1)}$ are the zeroth- and first-order moments of the Green's function, and can be computed by solving the following simultaneous linear equation:

$$\begin{pmatrix} 1 & z_0^{-1} \\ 1 & z_1^{-1} \end{pmatrix} \begin{pmatrix} M^{(0)} \\ M^{(1)} \end{pmatrix} = \begin{pmatrix} z_0 G(z_0) \\ z_1 G(z_1) \end{pmatrix}. \quad (12)$$

The equation is derived by terminating the summation over the order of the moments in the moment representation of the Green's function. By letting z_0 and z_1 be iR_e^2 and iR_e ,⁵¹ respectively, $M^{(0)}$ and $M^{(1)}$ are explicitly given by

$$M^{(0)} = \frac{iR_e}{R_e - 1} [R_e^2 G(iR_e^2) - G(iR_e)], \quad (13)$$

$$M^{(1)} = \frac{R_e^3}{R_e - 1} [R_e G(iR_e^2) - G(iR_e)], \quad (14)$$

where R_e should be a large real number and 10^7 hartree is used in this study so that the higher order terms can be negligible in terminating the summation in the moment representation of the Green's function. Inserting Eqs. (13) and (14) into Eq. (10), we obtain the following expression which is suitable for the efficient implementation in terms of memory consumption:

$$e = \lambda G(iR_e^2) + \gamma G(iR_e) + \text{Im} \left[-\frac{4i}{\beta} \sum_{p=1}^{\infty} G(\alpha_p) R_p \alpha_p \right] \quad (15)$$

with λ and γ defined by

$$\lambda = \frac{R_e^3}{R_e - 1} (R_e + i\kappa), \quad (16)$$

$$\gamma = -\frac{R_e}{R_e - 1} (R_e^2 + i\kappa). \quad (17)$$

One may notice that the number of poles for convergence does not depend on the size of system even for the calculation of the energy density matrix because of the same reason as for the density matrix.

C. Calculation of the Green's function

It is found from the above discussion that the computational effort to compute the density matrix is governed by that for the calculation of the Green's function, consisting of an inversion of the sparse matrix of which computational effort by conventional schemes such as the Gauss elimination or LU factorization-based methods scales as $O(N^3)$. Thus, the development of an efficient method of inverting a sparse matrix is crucial for efficiency of the proposed method.

Here we present an efficient low-order scaling method, based on a nested dissection approach,⁵⁰ of computing only

selected elements in the inverse of a sparse matrix. The low-order scaling method proposed here consists of two steps: (1) *nested dissection*: by noting that a matrix $X \equiv (ZS-H)$ is sparse, a structured matrix is constructed by a nested dissection approach. In practice, just reordering the column and row indices of the matrix X yields the structured matrix. (2) *Inverse by recurrence formulas*: by recursively applying a block LDL^T factorization¹⁰ to the structured matrix, a set of recurrence formulas is derived. Using the recurrence formulas, only the selected elements of the inverse matrix $X^{-1} \equiv G(Z)$ are directly computed. The computational effort to calculate the selected elements in the inverse matrix using the steps (i) and (ii) scales as $O[N(\log_2 N)^2]$, $O(N^2)$, and $O(N^{7/3})$ for 1D, 2D, and 3D systems, respectively, as shown later. First, we discuss the nested dissection of a sparse matrix, and then derive a set of recurrence formulas of calculating the selected elements of the inverse matrix.

1. Nested dissection

As an example the right panel of Fig. 1(c) shows a structured matrix obtained by the nested dissection approach for a finite chain model consisting of ten atoms, where we consider a s -valent nearest-neighbor tight-binding (NNTB) model. When one assigns the number to the ten atoms as shown in the left panel of Fig. 1(a), then X is a tridiagonal matrix, of which diagonal and off-diagonal terms are assumed to be a and b , respectively, as shown in the right panel of Fig. 1(a). As the first step to generate the structured matrix shown in the right panel of Fig. 1(c), we make a *dissection* of the system into the left and right *domains*⁵² by renumbering for the ten atoms, and obtain a dissected matrix shown in the right panel of Fig. 1(b). The left and right domains interact with each other through only a *separator* consisting of an atom 10. As the second step we apply a similar dissection for each domain generated by the first step, and arrive at a *nested-dissected* matrix given by the right panel of Fig. 1(c). The subdomains, which consist of atoms 1 and 2 and atoms 3 and 4, respectively, in the left domain interact with each other through only a separator consisting of an atom 5. The similar structure is also found in the right domain consisting of atoms 6, 7, 9, and 8. It is worth mentioning that the resultant nested structure of the sparse matrix can be mapped to a binary tree structure which indicates hierarchical interactions between (sub)domains as shown in Fig. 1(d). By applying the above procedure to a sparse matrix, one can convert any sparse matrix into a nested and dissected matrix in general. However in practice there is no obvious way to perform the nested dissection for general sparse matrices while a lot of efficient and effective methods have been already developed for the purpose.^{53,54} Here we propose a rather simple but effective way for the nested dissection by taking account of a fact that the basis function we are interested in is localized in real space, and that the sparse structure of the resultant matrix is very closely related to the position of basis functions in real space. The method bisects a system into two domains interacting through only a separator, and recursively applies to the resultant subdomains, leading to a binary tree structure for the interaction. Our algorithm for the

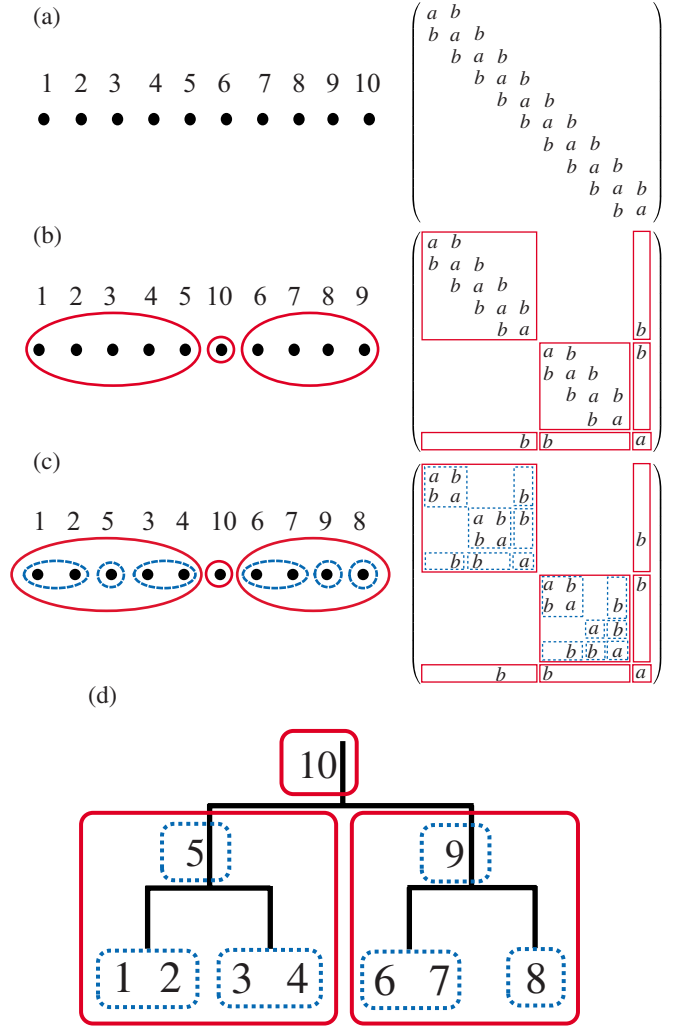


FIG. 1. (Color online) (a) The initial numbering for atoms in a linear-chain molecule consisting of ten atoms described by the s -valent NNTB and its corresponding matrix, (b) the renumbering for atoms by the first step in the nested dissection and its corresponding matrix, (c) the renumbering for atoms by the second step in the nested dissection and its corresponding matrix, and (d) the binary tree structure representing hierarchical interactions between domains in the structured matrix by the numbering shown in (c).

nested dissection of a general sparse matrix is summarized as follows.

(i) *Ordering*. Let us assume that there are N_d basis functions in a domain we are interested in. We order the basis functions in the domain by using the fractional coordinate for the central position of localized basis functions along \mathbf{a} , axis, where $i=1, 2$, and 3 . As a result of the ordering, each basis function can be specified by the *ordering number*, which runs from 1 to N_d in the domain of the central unit cell. The ordering number in the periodic cells specified by $l\mathbf{a}_i$, where $l=0, \pm 1, \pm 2, \dots$, is given by lN_d+q , where q is the corresponding ordering number in the central cell. In isolated systems, one can use the Cartesian coordinate instead of the fractional coordinate without losing any generality.

(ii) *Screening of basis functions with a long tail*. The basis functions with a long tail tend to make an efficient dissection

difficult. The sparse structure formed by the other basis functions with a short tail hides behind the dense structure caused by the basis functions with a long tail. Thus, we classify the basis functions with a long tail in the domain as members in the separator before performing the dissection process. By the screening of the basis functions with a long tail, it is possible to expose concealed sparse structure when atomic basis functions with a variety of tails are used while a systematic basis set such as the FE basis functions may not require the screening.

(iii) *Finding of a starting nucleus.* Among the localized basis functions in the domain, we search a basis function which has the smallest number of nonzero overlap with the other basis functions. Once the basis function is found, we set it as a starting *nucleus* of a subdomain.

(iv) *Growth of the nucleus.* Starting from a subdomain (nucleus) given by the procedure (iii), we grow the subdomain by increasing the size of nucleus step by step. The growth of the nucleus can be easily performed by managing the minimum and maximum ordering numbers, m_{\min} and m_{\max} , which ranges from 1 to N_d , and the grown subdomain is defined by basis functions with the successive ordering numbers between the minimum and maximum ordering numbers m_{\min} and m_{\max} . At each step in the growth of the subdomain, we search two basis functions which have the minimum ordering number n_{\min} and maximum ordering number n_{\max} among basis functions overlapping with the subdomain defined at the growth step. In the periodic boundary condition, n_{\min} can be smaller than zero and n_{\max} can be larger than the number of basis functions N_d . Then, the number of basis functions in the subdomain, the separator, and the other subdomain can be calculated by $N_0 \equiv m_{\max} - m_{\min} + 1$, $N_s \equiv n_{\max} - n_{\min} + 1 - N_0$, and $N_1 \equiv N_d - N_0 - N_s$, respectively, at each growth step. By the growth process one can minimize $(|N_0 - N_1| + N_s)$ being a measure for quality of the dissection, where the measure $(|N_0 - N_1| + N_s)$ takes equal bisection size of the subdomains and minimization of the size of the separator into account. Also if $(n_{\max} - n_{\min} + 1)$ is larger than N_d , then this situation implies that the proper dissection can be difficult along the axis.

(v) *Dissection.* By applying the above procedures (i)–(iv) to each \mathbf{a}_i axis, where $i=1, 2$, and 3 , and we can find an axis which gives the minimum $(|N_0 - N_1| + N_s)$. Then, the dissection along the axis is performed by renumbering for basis functions in the domain, and two subdomains and one separator are obtained. Evidently, the same procedures can be applied to each subdomain, and recursively continued until the size of domain reaches a threshold. As a result of the recursive dissection, a structured matrix specified by a binary tree is obtained.

As an illustration we apply the method for the nested dissection to the finite chain molecule shown in Fig. 1. We first set all the system as *domain* and start to apply the series of procedures to the domain. The procedure (i) is trivial for the case, and we obtain the numbering of atoms and the corresponding matrix shown in Fig. 1(a). Also it is noted that the screening of the basis functions with a long tail is unnecessary and that we only have to search the chain direction. In the procedure (iii), atoms 1 and 10 in Fig. 1(a) satisfy the condition. Choosing the atom 1 as a starting nucleus of the

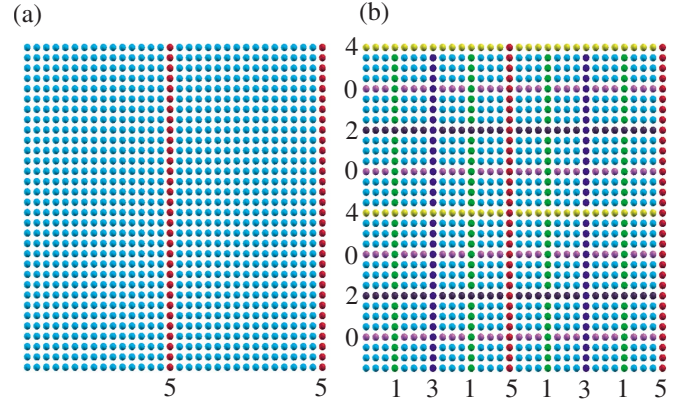


FIG. 2. (Color online) (a) The square lattice model, described by the s -valent NNTB, of which unit cell contains 1024 atoms with periodic boundary condition. The right blue and red circles correspond to atoms in two domains and a separator, respectively, at the first step in the nested dissection. (b) The square lattice model at the final step in the nested dissection. The separator at the innermost and the outermost levels are labeled as separators 0 and 5, respectively, and the separators at each level are constructed by atoms with a same color.

domain, and we gradually increase the size of the domain according to the procedure (iv). Then, it is found that the division shown in Fig. 1(b) gives the minimum $(|N_0 - N_1| + N_s)$. Renumbering for the basis functions based on the analysis yields the dissected matrix shown in the right panel of Fig. 1(b). By applying the similar procedures to the left and right subdomains, one will immediately find the result of Fig. 1(c). In addition to the finite chain molecule, as an example of more general cases, the above algorithm for the nested dissection is applied to a s -valent NNTB square lattice model of which unit cell contains 1024 atoms with periodic boundary condition. At the first step in the nested dissection, the separator is found to be red atoms as shown in Fig. 2(a). Due to the periodic boundary condition, the separator consists of two *lines*. At the final step, the system is dissected by the recursive algorithm as shown in Fig. 2(b). The separator at the innermost and the outermost levels are labeled as separators 0 and 5, respectively, and each subdomain at the innermost level includes nine atoms. As demonstrated for the square lattice model, the algorithm can be applied for systems with any dimensionality, and provides a well structured matrix for our purpose in a single framework.

2. Inverse by recurrence formulas

We directly compute the selected elements of the inverse matrix using a set of recurrence formulas which can be derived by recursively applying a block LDL^T factorization to the structured matrix obtained by the nested dissection method as shown below. To derive the recurrence formulas, we first introduce the block LDL^T factorization¹⁰ for a symmetric square matrix X ,

$$X = \begin{pmatrix} A & B^T \\ B & C \end{pmatrix}$$

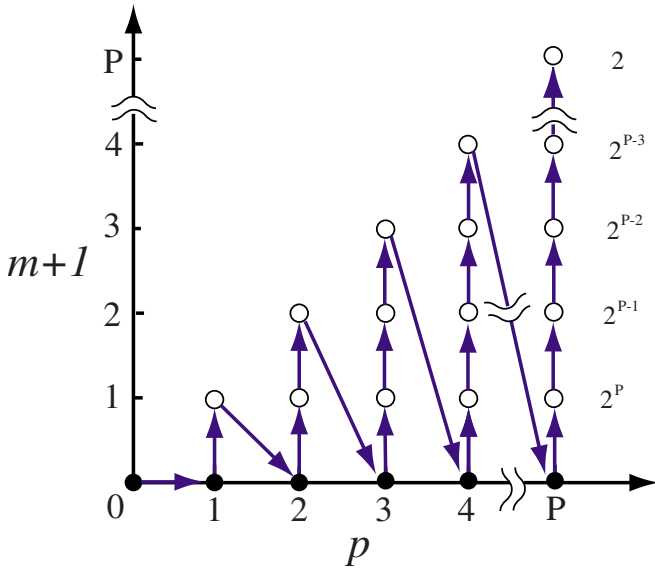


FIG. 3. (Color online) The development of recurrence formulas, Eqs. (28)–(32), which implies that the recurrence starts from $p = m+1=0$ and ends at $p=m+1=P$. The number on the right-hand side is the multiplicity for similar calculations by Eq. (29) due to the index n at each $m+1$.

rather intermediate one, being dependent on m . The indices n in Eq. (30) is dependent on p and q and they run as follows:

$$n = q(2^p), \dots, (q+1)(2^p) - 1, \quad (36)$$

$$q = 0, \dots, 2^{p+1-p} - 1. \quad (37)$$

Since the set of the recurrence formulas Eqs. (28)–(32) proceed according to Eqs. (33)–(35), the development of recurrence can be illustrated as in Fig. 3. The recurrence starts from Eq. (30) with $p=0$, and Eqs. (31) and (32) follow. Then, p is incremented by one, and $m+1$ climbs up to 1. The increment of p and the climbing of $m+1$ are repeated until $p=P$ and $m+1=P$. At $m+1=p$ for each p , L and S are evaluated by Eqs. (31) and (32), and the inverse of S is calculated by a conventional method such as LU factorization, which are used in the next recurrence for the higher level of hierarchy. The numbers on the right-hand side of Fig. 3 give the multiplicity for similar calculations by Eq. (29) coming from the index n at each $m+1$ since n runs from 0 to $2^{P-m}-1$ as given in Eq. (35). The computational complexity can be estimated by Fig. 3, and we will discuss its details later.

We are now ready to calculate the selected elements of the Green's function using the inverses of the Schur complements S and L calculated by the recurrence formulas of Eqs. (28)–(32). By noting that Eq. (21) has a recursive structure and the matrix X is structured by the nested dissection, one can derive the following recurrence formula:

$$X_{p+1,n}^{-1} = \begin{pmatrix} X_{p,2n}^{-1} & & \\ & X_{p,2n+1}^{-1} & \\ & & 0 \end{pmatrix} + \begin{pmatrix} Y_{p,2n}^T L_{p,2n} & & -Y_{p,2n}^T \\ & Y_{p,2n+1}^T L_{p,2n+1} & -Y_{p,2n+1}^T \\ -Y_{p,2n} & -Y_{p,2n+1} & S_{p,n}^{-1} \end{pmatrix}, \quad (38)$$

where

$$Y_{p,2n}^T = L_{p,2n}^T S_{p,n}^{-1},$$

$$Y_{p,2n+1}^T = L_{p,2n+1}^T S_{p,n}^{-1}. \quad (39)$$

The recurrence formula, Eq. (38), starts with $X_{0,n}^{-1} = (A_{0,n})^{-1}$, adds contributions at $m+1=p$ for every p , and at last yields the inverse of the matrix X as $X^{-1} = G(Z) = X_{P+1,0}^{-1}$. Since the calculation of each element for the inverse of X can be independently performed, only the selected elements can be computed without calculating all the elements. The selected elements to be calculated are elements in the block matrices A , B , and C , each of which corresponds to a nonzero overlap matrix as discussed before. Thus, we can easily compute only the selected elements using a table function which stores the position for the nonzero elements in the block matrices A , B , and C .

A simple but nontrivial example is given in Appendix A to illustrate how the inverse of matrix is computed by the recurrence formulas, and also a similar way is presented to calculate a few eigenstates around a selected energy in Appendix B while the proposed method can calculate the total energy of system without calculating the eigenstates.

3. Finding chemical potential

As well as the conventional DFT calculations, in the proposed method the chemical potential has to be adjusted so that the number of electrons can be conserved. However, there is no simpler way to know the number of electrons under a certain chemical potential before the contour integration by Eq. (8) with the chemical potential. Thus, we search the chemical potential by iterative methods for the charge conservation. Since the contour integration is the time-consuming step in the method, a smaller number of the iterative step directly leads to the faster calculation. Therefore, we develop a careful combination of several iterative methods to minimize the number of the iterative step for sufficient convergence. In general, the procedure for searching the chemical potential can be performed by a sequence (1)-(2) or (5)-(1)-(3)-(1)-(4)-(1)-(4)-(1)... in terms of the following procedures. As shown later, the procedure enables us to obtain the chemical potential conserving the number of electrons within 10^{-8} electron/system by less than five iterations on an average.

(1) *Calculation of the difference ΔN_i in the total number of electrons.* The difference ΔN_i in the total number of electrons is defined with $\rho(\mu_i)$ calculated using Eq. (8) at a chemical potential μ_i by

$$\Delta N_i = \text{Tr}[\rho(\mu_i)S] - N_{\text{ideal}}, \quad (40)$$

where N_{ideal} is the number of electrons that the system should possess for the charge conservation. If ΔN_i is zero, the chemical potential μ_i is the desired one of the system.

(2) *Using the retarded Green's function.* If the difference ΔN_i is large enough so that the interpolation schemes (3) and (4) can fail to guess a good chemical potential, the next trial chemical potential is estimated by using the retarded Green's function. When the chemical potential of μ_{tri} is considered, the correction δN_{tri} estimated by the retarded Green's function to ΔN_i is given by

$$\delta N_{\text{tri}} = \int_{E_{\text{min}}}^{E_{\text{max}}} dE \delta\rho(E) \Delta f(E, \mu_{\text{tri}}), \quad (41)$$

where $\delta\rho(E)$ and $\Delta f(E, \mu_{\text{tri}})$ are defined by

$$\delta\rho(E) = -\frac{2}{\pi} \text{Im} \text{Tr}[G(E + i\eta)S] \quad (42)$$

with a small number η (0.01 eV in this study) and

$$\Delta f(E, \mu_{\text{tri}}) = f\left(\frac{E - \mu_{\text{tri}}}{k_B T}\right) - f\left(\frac{E - \mu_i}{k_B T}\right). \quad (43)$$

The integration in Eq. (41) is numerically evaluated by a simple quadrature scheme such as trapezoidal rule with a similar number of points as for that of poles in Eq. (8), and the integration range can be determined by considering the surviving range of $\Delta f(E, \mu_{\text{tri}})$. The search of μ_{tri} is performed by a bisection method until $\Delta N_{\text{cri}} > (\Delta N_i + \delta N_{\text{tri}})$, where ΔN_{cri} is a criterion for the convergence and 10^{-8} electron/system is used in this study. It should be noted that the evaluation of Green's function being the time-consuming part can be performed before the bisection method and a set of $\delta\rho(E)$ is stored for computational efficiency.

(3) *Linear interpolation/extrapolation.* In searching the chemical potential μ , if two previous results $(\mu_i, \Delta N_i)$ and $(\mu_j, \Delta N_j)$ are available, a trial chemical potential μ_{tri} is estimated by a linear interpolation/extrapolation method as

$$\mu_{\text{tri}} = \frac{\mu_j \Delta N_i - \mu_i \Delta N_j}{\mu_i - \mu_j}. \quad (44)$$

(4) *Muller method.*^{55,56} In searching the chemical potential μ , if three previous results $(\mu_i, \Delta N_i)$, $(\mu_j, \Delta N_j)$, and $(\mu_k, \Delta N_k)$ are available, they can be fitted to a quadratic equation,

$$\Delta N = a\mu^2 + b\mu + c, \quad (45)$$

where a , b , and c are found by solving a simultaneous linear equation of 3×3 in size.⁵⁷ Then, μ_{tri} giving $\Delta N=0$ is a solution of Eq. (45) and given by

$$\mu_{\text{tri}} = \begin{cases} \frac{-2c}{b + \sqrt{b^2 - 4ac}} & b \geq 0, \\ \frac{-b + \sqrt{b^2 - 4ac}}{2a} & b < 0. \end{cases} \quad (46)$$

The selection of sign is unique because of the condition that the gradient at the solution must be positive, and the branch-

ing is taken into account to avoid the round-off error. As the iteration proceeds in search of the chemical potential, we have a situation that the number of available previous results is more than three. For the case, it is important to select three chemical potentials having smaller ΔN and the different sign of ΔN among three chemical potentials since the guess of μ_{tri} can be performed as the interpolation rather than the extrapolation.

(5) *Extrapolation of chemical potential for the second step.* During the self-consistent field (SCF) iteration, the chemical potential obtained at the last SCF step is used as the initial guess μ_1 in the current SCF step. In addition, we estimate the second trial chemical potential by fitting a set of results $(\mu_1^{(i)}, \Delta N_1^{(i)}, \mu_2^{(i)}, \Delta N_2^{(i)})$, $(\mu_1^{(j)}, \Delta N_1^{(j)}, \mu_2^{(j)}, \Delta N_2^{(j)})$, and $(\mu_1^{(k)}, \Delta N_1^{(k)}, \mu_2^{(k)}, \Delta N_2^{(k)})$, where the subscript and the superscript in $\mu_0^{(i)}$ and $\Delta N_0^{(i)}$ mean the iteration step in search of the chemical potential and the SCF step, respectively, at three previous SCF steps to the following equation:

$$\Delta N_2 = a_1 \Delta N_1 + a_2 (\mu_2 - \mu_1) + a_3 \Delta N_1 (\mu_2 - \mu_1), \quad (47)$$

where a_1 , a_2 , and a_3 are found by solving a simultaneous linear equation of 3×3 in size. Then, the chemical potential μ_2 giving $\Delta N_2=0$ can be estimated by solving Eq. (47) with respect to μ_2 as follows:

$$\mu_{\text{tri}} \equiv \mu_2 = \mu_1 - \frac{a_1 \Delta N_1}{a_2 + a_3 \Delta N_1}. \quad (48)$$

It is found from numerical calculations that Eq. (48) provides a very accurate guess in most cases as the SCF calculation converges.

D. Computational complexity

We analyze the computational complexity of the proposed method. As discussed in Sec. II B, the number of poles for the contour integration is independent of the size of system. Thus, we focus on the computational complexity of the calculation of the Green's function. For simplicity of the analysis we consider a finite chain, a finite square lattice, and a finite cubic lattice as representatives of 1D, 2D, and 3D systems, respectively, which are described by the s -valent NNTB models as in the explanation of the nested dissection. Note that the results in the analysis might be valid for more general cases with periodic boundary conditions. Since the computational cost is governed by Eq. (29), let us first analyze the computational cost of Eq. (29) while those of the other equations will be discussed later. Considering that the recurrence formula of Eq. (29) develops as shown in Fig. 3, and that the calculation of Eq. (29) corresponds to the open circle in the figure, the computational cost t can be estimated by

$$t \propto \sum_{p=1}^P \sum_{m=0}^{p-1} \sum_{n=0}^{2^{p-m}-1} N_m^{(1)} N_m^{(2)} N_p^{(3)}, \quad (49)$$

where $N_m^{(1)}$ and $N_m^{(2)}$ are the dimension of row and column in the matrix,

TABLE I. Some of $N_m^{(2)}$ and $N_p^{(3)}$ in Eq. (49) for a finite 1D chain, a finite 2D square lattice, and a finite 3D cubic lattice described by the s -valent NNTB model. They depends on m or p for the 2D and 3D systems in a rather complicated way, while $N_{p,m,n}^{(1)} = \frac{N}{2^{P-m}}$ for all the cases. The unit for each case is given in parenthesis.

$m+1$ or p	P	$P-1$	$P-2$	$P-3$	$P-4$	$P-5$	$P-6$	$P-7$	$P-8$	$P-9$	$P-10$
1D (1)	1	1	1	1	1	1	1	1	1	1	1
2D ($N^{1/2}$)	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{32}$
3D ($N^{2/3}$)	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{128}$

$$\begin{pmatrix} L_{m,2n}^T \\ L_{m,2n+1}^T \\ -I \end{pmatrix}$$

and $N_p^{(3)}$ is the dimension of column in the matrix $Q_{p,m+1,n}^T$. Since Eq. (29) consists of a matrix product, the computational cost is simply given by $N_m^{(1)}N_m^{(2)}N_p^{(3)}$. Also it is noted that $N_m^{(1)}$ and $N_m^{(2)}$ depend on only m , and $N_p^{(3)}$ has dependency on only p because of the simplicity of the systems we consider.

For the finite 1D chain system, we see that $N_m^{(1)} = N/(2^{P-m})$ and $N_m^{(2)} = N_p^{(2)} = 1$ as listed in Table I. Thus, the computational cost t_{1D} for the 1D system is estimated as

$$\begin{aligned} t_{1D} &\propto \sum_{p=1}^P \sum_{m=0}^{p-1} \sum_{n=0}^{2^{P-m-1}} \frac{N}{2^{P-m}} \\ &= \frac{1}{2}NP(P+1). \end{aligned} \quad (50)$$

Noting $N \propto 2^P$, we see that the computational cost for the 1D system scales as $O\{N[\log_2(N)]^2\}$.

For the finite 2D square lattice system, we see $N_m^{(1)} = N/(2^{P-m})$, and $N_m^{(2)}$ and $N_p^{(3)}$ depend on m and p , respectively as shown in Table I. To estimate the order of the computational cost we approximate $N_m^{(2)}$ and $N_p^{(3)}$ as $N_m^{(2)} \approx N^{1/2}/2^{1/2(P-m-1)}$ and $N_p^{(3)} \approx N^{1/2}/2^{(1/2)(P-p)}$ which are equal to or more than the corresponding exact number. Then, the computational cost t_{2D} for the 2D system can be estimated as follows:

$$\begin{aligned} t_{2D} &\propto \sum_{p=1}^P \sum_{m=0}^{p-1} \sum_{n=0}^{2^{P-m-1}} \frac{N}{2^{P-m}} N_{p,m,n}^{(2)} N_{p,m,n}^{(3)} \\ &< \sum_{p=1}^P \sum_{m=0}^{p-1} \sum_{n=0}^{2^{P-m-1}} \frac{N}{2^{P-m}} \frac{N^{1/2}}{2^{(1/2)(P-m-1)}} \frac{N^{1/2}}{2^{(1/2)(P-p)}} \\ &= \frac{2N^2}{(\sqrt{2}-1)^2} \left(2 - \sqrt{2} + \frac{\sqrt{2}}{2^P} - \frac{1}{2^P} - \frac{1}{2^{P/2}} \right). \end{aligned} \quad (51)$$

Since the first two terms in parenthesis of the last line are the leading term, we see that the computational cost for the 2D system scales as $O(N^2)$.

For the finite 3D cubic lattice system we have $N_m^{(1)} = N/(2^{P-m})$ as well as the 1D and 2D systems. As shown in the analysis of the 2D systems, by approximating $N_m^{(2)}$ and

$N_p^{(3)}$ as $N_m^{(2)} \approx N^{2/3}/2^{(2/3)(P-m-1)}$ and $N_p^{(3)} \approx N^{2/3}/2^{(2/3)(P-p)}$, which are equal to or more than the corresponding exact number, we can estimate the computational cost t_{3D} for the 3D system as follows:

$$\begin{aligned} t_{3D} &\propto \sum_{p=1}^P \sum_{m=0}^{p-1} \sum_{n=0}^{2^{P-m-1}} \frac{N}{2^{P-m}} N_{p,m,n}^{(2)} N_{p,m,n}^{(3)} \\ &< \sum_{p=1}^P \sum_{m=0}^{p-1} \sum_{n=0}^{2^{P-m-1}} \frac{N}{2^{P-m}} \frac{N^{2/3}}{2^{(2/3)(P-m-1)}} \frac{N^{2/3}}{2^{(2/3)(P-p)}} \\ &= \frac{4N^{7/3}}{2^{2/3}6-9} \left(-1 + 2^{2/3} - \frac{1}{2^{2/3}2^{4P/3}} \right. \\ &\quad \left. + \frac{1}{2^{2/3}2^{2P/3}} - \frac{2^{2/3}}{2^{2P/3}} + \frac{1}{2^{4P/3}} \right). \end{aligned} \quad (52)$$

Since we see that the first two terms in parenthesis of the last line are the leading term, it is concluded that the computational cost for the 3D system scales as $O(N^{7/3})$.

We further analyze the computational cost of the other Eqs. (28), (30), (32), (38), and (39) which are the primary equations for the calculation of the Green's function. Although the detailed derivations are not shown here, they can be derived in the same way as for Eq. (29). Table II shows the order of the computational cost for each equation. It is found that the computational cost is governed by Eq. (29) while the computational cost of Eq. (39) is similar to that of Eq. (29).

In addition to the analysis of the computational cost for the inverse calculations by Eq. (28)–(32), (38), and (39), we

TABLE II. Computational order of Eqs. (28)–(30), (32), (38), and (39), where the calculation of the inverse of the matrix S is also included in estimating the computational cost of Eq. (32), and the sparse structure in the matrix B is taken into account for Eqs. (28) and (32).

	1D	2D	3D
Equation (28)	$(\log_2 N)^2$	$N^{3/2} \log_2 N$	N^2
Equation (29)	$N(\log_2 N)^2$	N^2	$N^{7/3}$
Equation (30)	$N \log_2 N$	$N^{3/2}$	$N^{5/3}$
Equation (32)	$\log_2 N$	N	$N^{4/3}$
Equation (38)	N	$N^{3/2}$	$N^{5/3}$
Equation (39)	$N \log_2 N$	N^2	$N^{7/3}$

examine that for the nested dissection. The step (i) in the nested dissection involves a sorting procedure by the quick-sort algorithm of which computational cost scales as $\frac{N}{2^p} \log_2 \frac{N}{2^p}$ for a sequence with length of $\frac{N}{2^p}$, and thereby the computational cost for d -dimensional system can be estimated by

$$d \sum_{p=0}^P \left(\frac{N}{2^p} \log_2 \frac{N}{2^p} \right) 2^p = \frac{d}{2} N (\log_2 N)^2. \quad (53)$$

Also, considering that the steps (iii) and (iv) consist of search for the sequence with length of $\frac{N}{2^p}$, the computational cost is found to be

$$d \sum_{p=0}^P \frac{N}{2^p} 2^p = dN \log_2 N. \quad (54)$$

The analysis indicates that the computational cost for the nested dissection scales as $O[N(\log_2 N)^2]$ for systems with any dimensionality. Thus, it is concluded that as a whole the proposed method scales as $O[N(\log_2 N)^2]$, $O(N^2)$, and $O(N^{7/3})$ for 1D, 2D, and 3D systems, respectively.⁵⁸

III. NUMERICAL RESULTS

In this section several numerical calculations for the s -valent NNTB model and DFT are presented to illustrate practical aspects of the low-order scaling method. All the DFT calculations in this study were performed by the DFT code OPENMX.⁵⁹ The PAO basis functions⁴⁶ used in the DFT calculations are specified by H4.5- $s1$, C5.0- $s1p1$, N4.5- $s1p1$, O4.5- $s1p1$, and P6.0- $s1p1d1$ for deoxyribonucleic acid (DNA), C4.0- $s1p1$ for a single C₆₀ molecule, and Pt7.0- $s2p2d1$ for a single Pt₆₃ cluster, respectively, where the abbreviation of basis function such as C5.0- $s1p1$ represents that C stands for the atomic symbol, 5.0 the cutoff radius (bohr) in the generation by the confinement scheme, $s1p1$ means the employment of one primitive orbitals for each of s and p orbitals.⁴⁶ Since the PAO basis functions are pseudoatomic orbitals with different cutoff radii depending on atomic species, the resultant Hamiltonian and overlap matrices have a disordered sparse structure, reflecting the geometrical structure of the system. Norm-conserving pseudopotentials are used in a separable form with multiple projectors to replace the deep core potential into a shallow potential.⁶⁰ Also a LDA to the exchange-correlation potential is employed.³⁸

A. Scaling

As shown in the previous section, it is possible to reduce the computational cost from $O(N^3)$ to the low-order scaling without losing numerical accuracy. Here we validate the theoretical scaling property of the computational effort by numerical calculations. Figure 4 shows the elapsed time required for the calculation of inverse of a 1D linear chain, a 2D square lattice, and a 3D cubic lattice systems as a function of number of atoms in the unit cell under periodic boundary condition, which are described by the s -valent NNTB models. The last three points for each system are

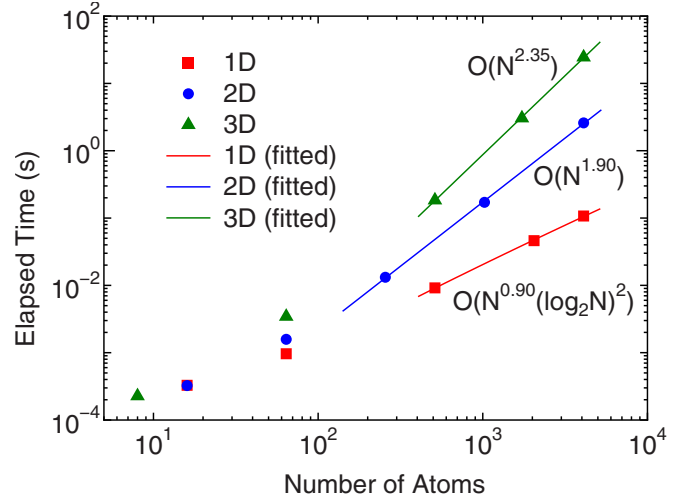


FIG. 4. (Color online) The elapsed time of the inverse calculation by Eqs. (28)–(32) for a 1D linear chain, a 2D square lattice, and a 3D cubic lattice systems as a function of number of atoms in the unit cell under periodic boundary condition. The Hamiltonian of the systems is described by the s -valent NNTB models. The line for each system is obtained by a least-square method, and the computational orders obtained from the fitted curves are $O[N^{0.90}(\log_2 N)^2]$, $O(N^{1.90})$, and $O(N^{2.35})$ for the 1D, 2D, and 3D systems, respectively. The size of domains at the innermost level is set to 20 for all the cases.

fitted to a function by a least square method, and the obtained scalings of the elapsed time are found to be $O[N^{0.90}(\log_2 N)^2]$, $O(N^{1.90})$, and $O(N^{2.35})$ for the 1D, 2D, and 3D systems, respectively. Thus, it is confirmed that the scaling of the computational cost is nearly the same as that of the theoretical estimation.

B. SCF calculation

To demonstrate that the proposed method is a numerically exact method even if the summation in Eq. (8) is terminated at a modest number of poles, we show the convergence in the SCF calculations calculated by the conventional diagonalization and the proposed methods for DNA in Fig. 5, where 80 poles is used for the summation, and the electronic temperature is 700 K. It is clearly seen that the convergence property and the total energy are almost equivalent to those by the conventional method with only 80 poles. Table III also presents the rapid convergence of the total energy and forces in the SCF calculation as a function of the number of poles. In the case, the use of only 40 poles is enough to achieve the numerically exact results for not only the total energy but also the forces on atoms within numerical precision. Since the density matrix, total energy, and forces on atoms can be very accurately evaluated within numerical precision as shown above, the low-order scaling method can be regarded as a variational method in practice.

C. Iterative search of chemical potential

Although the computational cost of the proposed method can be reduced from the cubic to low-order scalings, the

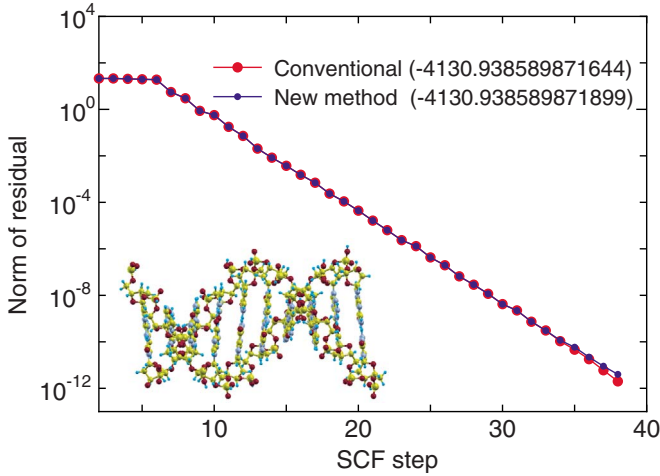


FIG. 5. (Color online) The norm of residual in the SCF calculation of DNA, with a periodic double helix structure (650 atoms/unit) consisting of cytosines and guanines, calculated by the conventional and proposed methods, where the residual is defined as the difference between the input and output charge densities in momentum space. The electric temperature of 700 K and 80 poles for the contour integration are used. The number in parenthesis is the total energy (hartree) of the system calculated by each method.

prefactor directly depends on the number of iterations in the iterative search of the chemical potential. To address how the combination of interpolation and extrapolation methods discussed before works to search a chemical potential which conserves the total number of electrons within a criterion, we show in Fig. 6 the number of iterations for finding the chemical potential, conserving the total number of electrons with a criterion of 10^{-8} electron/system, as a function of the SCF step for a C_{60} molecule, DNA, and a Pt_{63} cluster. Only few iterations are enough to achieve a sufficient convergence of the chemical potential as the SCF calculation converges while a larger number of iterations are required at the initial stage of the SCF step. It turns out that the proper chemical potential can be searched by the mean iterations of 2.1, 2.4, and 4.0 for a C_{60} molecule, DNA, and a Pt_{63} cluster, respectively. The property of the iterative search is closely related to the energy gap of systems. The energy gap between the

TABLE III. The absolute error of total energy $|E - E_{\text{ref}}|$ (hartree) and the mean absolute error (MAE) of forces on atoms (hartree/bohr) in the SCF calculation of the same DNA system as for Fig. 5 as a function of the number of poles. The reference energy E_{ref} and forces are calculated by the conventional diagonalization method. The computational conditions are the same as for the calculations of Fig. 5.

Poles	$ E - E_{\text{ref}} $	MAE of forces
10	80.698617103348	0.040703500227
20	0.122135859603	0.000111580338
40	0.00000000162	0.00000000148
60	0.00000000264	0.00000000148
80	0.00000000255	0.00000000148
100	0.00000000251	0.00000000148

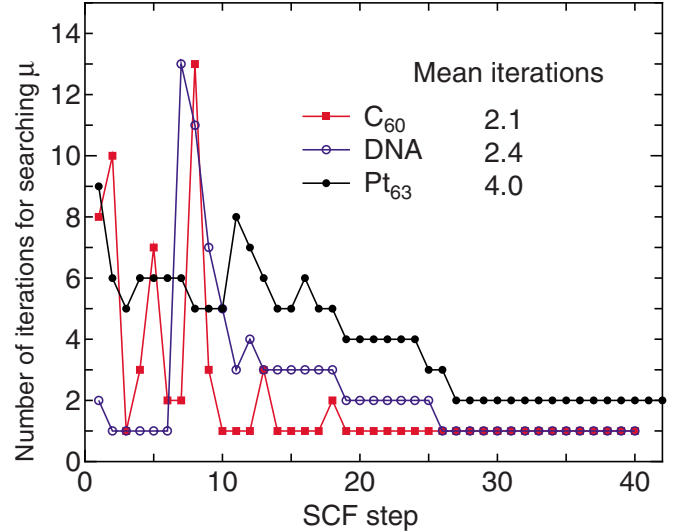


FIG. 6. (Color online) The number of iterations for searching the chemical potential which conserves the total number of electrons within a criterion of 10^{-8} electron/system for a C_{60} molecule, DNA, and a Pt_{63} cluster, where the electric temperature of 600, 700, and 1000 K, and 80, 80, and 90 poles for the contour integration are used for the C_{60} molecule, DNA, and the Pt_{63} cluster.

highest occupied and lowest unoccupied states of the C_{60} molecule, DNA, and Pt_{63} cluster are 1.95, 0.67, and 0.02 eV, respectively. For the C_{60} molecule and DNA with wide gaps the number of iterations for finding the chemical potential tends to be large up to 10 SCF iterations since the interpolation or extrapolation scheme may not work well due to the existence of the wide gap.

However, once the charge density nearly converges, the approximate chemical potential in between the gap, which is the correct chemical potential at the previous SCF step, can satisfy the criterion of 10^{-8} electron/system. The situation does correspond to a small number of iterations after ten SCF iterations. Even the trial chemical potential at the first step is the correct one within the criterion after 26 SCF iterations in these cases. For the Pt_{63} cluster with the narrow gap the number of iterations for finding the chemical potential is slightly lower than those of the a C_{60} molecule and DNA with the wide gaps at the initial stage of SCF iterations, which implies that the interpolation and extrapolation schemes by the procedures (3)–(5) can give a good estimation of the chemical potential for the nearly continuous eigenvalue spectrum. In addition to this, one may find that in contrast to the cases with the wide gap, the correct chemical potential is found by two iterations as the charge density converges since a little change in the chemical potential affects the distribution of charge density due to the narrow gap. However, the fact that only two iterations are sufficient even for the system with a narrow gap at the final stage of the SCF step suggests that the extrapolation by Eq. (48) works very well. Thus, we see from the numerical calculations that the correct chemical potential can be searched by only few iterations on an average with the combination of interpolation and extrapolation methods for systems with a wide variety of gap.

D. Parallel calculation

We demonstrate that the proposed method is suitable for the parallel computation because of the well separated data structure. It is apparent that the calculation of the Green's function at each α_p in Eq. (8) can be independently performed without data communication among processors. Thus, we parallelize the summation in Eq. (8) by using the message passing interface (MPI) in which a nearly same number of poles are distributed to each process. The summation in Eq. (8) can be partly performed in each process, and the global summation is completed after all the calculations allocated to each process finish. In most cases the global summation can be a very small fraction of the computational time even including the MPI communication since the amount of the data to be communicated is $O(N)$ due to the use of localized basis functions. In addition to the parallelization of the summation in Eq. (8), the calculation of the Green's function can be parallelized in two respects. In the recursive calculations of Eqs. (28)–(32), one may notice that the calculation for different n is independently performed, and also the calculations involving V^T and L^T in Eqs. (28)–(32) can be parallelized with respect to the column of V^T and L^T without communication until the recurrence calculations reach at $m+1=p$. For each p the MPI communication only has to be performed at $m+1=p$. In our implementation only the latter part as for the calculation of the Green's function is parallelized by a hybrid parallelization using MPI and OPENMP, which are used for internodes and intranode parallelization. As a whole, we parallelize the summation in Eq. (8) using MPI and the calculations involving V^T and L^T in Eqs. (28)–(32) using the hybrid scheme.

Figure 7 shows the speed-up ratio by the parallel calculation in the elapsed time of one SCF iteration. The speed-up ratio reaches about 350 and the elapsed time obtained is 3.76 s using 162 processes and 4 threads, which demonstrates the good scalability of the proposed method. On the other hand, the conventional diagonalization using Householder and QR methods scales up to only 21 processes, which leads to the speed-up ratio of about 10 and the elapsed time of 7.09 s. Thus, we see that the proposed method is of great advantage to the parallel computation unlike the conventional method while the comparison of the elapsed time suggests that the prefactor in the computational effort for the proposed method is larger than that of the conventional method.

IV. CONCLUSIONS

An efficient low-order scaling method has been developed for large-scale DFT calculations using localized basis functions such as the PAO, FE, and wavelet basis functions, which can be applied to not only insulating but also metallic systems. The computational effort of the method scales as $O[N(\log_2 N)^2]$, $O(N^2)$, and $O(N^{7/3})$ for 1D, 2D, and 3D systems, respectively. The method directly evaluates, based on two ideas, only selected elements in the density matrix which are required for the total-energy calculation. The first idea is to introduce a contour integration method for the integration of the Green's function in which the Fermi-Dirac function is expressed by a continued fraction. The contour integration

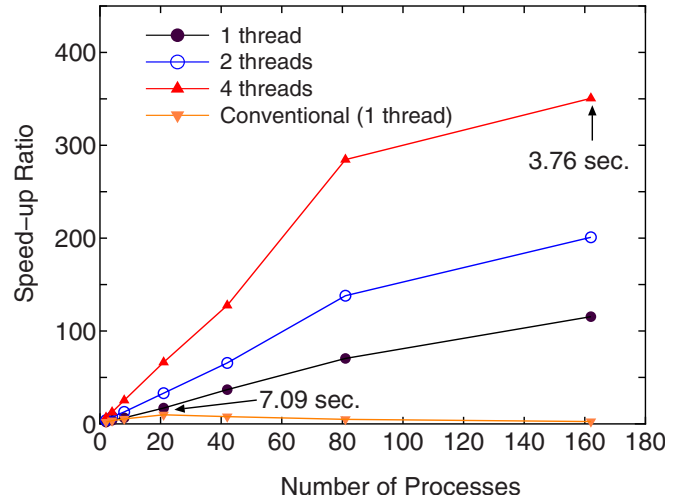


FIG. 7. (Color online) Speed-up ratio in the parallel computation of the diagonalization in the SCF calculation for DNA by a hybrid scheme using MPI and OPENMP. The speed-up ratio is defined by $2T_2/T_p$, where T_2 and T_p are the elapsed times obtained by two MPI processes and by the corresponding number of processes and threads. The structure of DNA is the same as in Fig. 5. The parallel calculations were performed on a Cray XT5 machine consisting of AMD opteron quad core processors (2.3 GHz). The electric temperature of 700 K and 80 poles for the contour integration are used. For comparison, the speed-up ratio for the parallel computation of the conventional scheme using Householder and QR methods is also shown for the case with a single thread.

enables us to obtain the numerically exact result for the integration within double precision at a modest number of poles, which allows us to regard the method as a numerically exact alternative to conventional $O(N^3)$ diagonalization methods. It is also shown that the number of poles needed for the convergence does not depend on the size of the system, but the spectrum radius of the system, which implies that the number of poles in the contour integration is unconcerned with the scaling property of the computation cost. The second idea is to employ a set of recurrence formulas for the calculation of the Green's function. The set of recurrence formulas is derived from a recursive application of a block LDL^T factorization using the Schur complement to a structured matrix obtained by a nested dissection for the sparse matrix $(ZS-H)$. The primary calculation in the recurrence formulas consists of matrix multiplications, and the computational scaling property is derived by the detailed analysis for the calculations. The chemical potential, conserving the total number of electrons, is determined by an iterative search which combines several interpolation and extrapolation methods. The iterative search permits to find the chemical potential by less than five iterations on an average for systems with a wide variety of gap. The good scalability in the parallel computation implies that the method is suitable for the massively parallel computation, and could extend the applicability of DFT calculations for large-scale systems together with the low-order scaling. It is also anticipated that the numerically exact method provides a platform in developing novel approximate efficient methods.

ACKNOWLEDGMENTS

The author was partly supported by the Fujitsu Laboratory, the Nissan Motor Co., Ltd., Nippon Sheet Glass Co., Ltd., and the Next Generation Super Computing Project, Nanoscience Program, MEXT, Japan. The author would like to thank M. Toyoda for critical reading of the manuscript.

APPENDIX A: AN EXAMPLE OF THE INVERSE CALCULATION

Since the proposed method to calculate the inverse of matrix is largely different from conventional methods, we show a simple but nontrivial example to illustrate the calculation of the inverse by using the set of recurrence formulas, Eq. (28)–(32), (38), and (39), which may be useful to understand how the calculation proceeds. We consider a finite chain molecule consisting of seven atoms described by the same s -valent NNTB model as in Sec. II C 1, where all the on-site energies and hopping integrals are assumed to be 1. After performing the nested dissection, we obtain the following structured matrix:

$$X = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix}, \quad (\text{A1})$$

where the blank means a zero element. It can be seen that the structure is same as in Eq. (22), and the system is divided into four domains with $P=1$. Then, we start from Eqs. (30) with $p=0$,

$$\begin{aligned} V_{0,0,0}^T &= (A_{0,0})^{-1}(B_{0,0}[B_{0,0}])^T \\ &= 1 \times 1 = 1 \equiv L_{0,0}^T, \end{aligned} \quad (\text{A2})$$

$$V_{0,0,1}^T = (A_{0,1})^{-1}(B_{0,1}[B_{0,1}])^T = 1 \times 1 = 1 \equiv L_{0,1}^T, \quad (\text{A3})$$

$$V_{0,0,2}^T = (A_{0,2})^{-1}(B_{0,2}[B_{0,2}])^T = 1 \times 1 = 1 \equiv L_{0,2}^T, \quad (\text{A4})$$

$$V_{0,0,3}^T = (A_{0,3})^{-1}(B_{0,3}[B_{0,3}])^T = 1 \times 1 = 1 \equiv L_{0,3}^T, \quad (\text{A5})$$

and proceed to calculate Eq. (32),

$$\begin{aligned} S_{0,0} &= C_{0,0} - B_{0,0}L_{0,0}^T - B_{0,1}L_{0,1}^T \\ &= 1 - 1 \times 1 - 1 \times 1 = -1, \end{aligned} \quad (\text{A6})$$

$$\begin{aligned} S_{0,1} &= C_{0,1} - B_{0,2}L_{0,2}^T - B_{0,3}L_{0,3}^T \\ &= 1 - 1 \times 1 - 1 \times 1 = -1. \end{aligned} \quad (\text{A7})$$

$X_{1,0}^{-1}$ and $X_{1,1}^{-1}$ which are precursors of the inverse of X can be calculated by Eq. (38) and (39) as

$$\begin{aligned} X_{1,0}^{-1} &= \begin{pmatrix} A_{0,0}^{-1} & * & * \\ * & A_{0,1}^{-1} & * \\ * & * & 0 \end{pmatrix} + \begin{pmatrix} Y_{0,0}^T L_{0,0} & * & -Y_{0,0}^T \\ * & Y_{0,1}^T L_{0,1} & -Y_{0,1}^T \\ -Y_{0,0} & -Y_{0,1} & S_{0,0}^{-1} \end{pmatrix} \\ &= \begin{pmatrix} 1 & * & * \\ * & 1 & * \\ * & * & 0 \end{pmatrix} + \begin{pmatrix} (-1) & * & -(-1) \\ * & (-1) & -(-1) \\ -(-1) & -(-1) & -1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & * & 1 \\ * & 0 & 1 \\ 1 & 1 & -1 \end{pmatrix} = X_{1,1}^{-1}, \end{aligned} \quad (\text{A8})$$

where $*$ means that the corresponding element is not calculated, and remains unknown, since these elements are not referred for further calculations. The precursor of $X_{1,1}^{-1}$ is found to be same as $X_{1,0}$ due to the same inner structure. As the next step, we set p to 1, and calculate Eq. (30),

$$V_{1,0,0}^T = (A_{0,0})^{-1}(B_{1,0}[B_{0,0}])^T = 1 \times 0 = 0, \quad (\text{A9})$$

$$V_{1,0,1}^T = (A_{0,1})^{-1}(B_{1,0}[B_{0,1}])^T = 1 \times 1 = 1, \quad (\text{A10})$$

$$V_{1,0,2}^T = (A_{0,2})^{-1}(B_{1,1}[B_{0,2}])^T = 1 \times 1 = 1, \quad (\text{A11})$$

$$V_{1,0,3}^T = (A_{0,3})^{-1}(B_{1,1}[B_{0,3}])^T = 1 \times 0 = 0, \quad (\text{A12})$$

Eq. (28),

$$\begin{aligned} Q_{1,1,0}^T &= S_{0,0}^{-1}\{B_{0,0}V_{1,0,0}^T + B_{0,1}V_{1,0,1}^T - [B_{1,0}(C_{0,0})]^T\} \\ &= (-1)(1 \times 0 + 1 \times 1 - 0) = -1, \end{aligned} \quad (\text{A13})$$

$$\begin{aligned} Q_{1,1,1}^T &= S_{0,1}^{-1}\{B_{0,2}V_{1,0,2}^T + B_{0,3}V_{1,0,3}^T - [B_{1,1}(C_{0,1})]^T\} \\ &= (-1)(1 \times 1 + 1 \times 0 - 0) = -1, \end{aligned} \quad (\text{A14})$$

Eqs. (29) and (31),

$$\begin{aligned} V_{1,1,0}^T &= \begin{pmatrix} V_{1,0,0}^T \\ V_{1,0,1}^T \\ 0 \end{pmatrix} + \begin{pmatrix} L_{0,0}^T \\ L_{0,1}^T \\ -1 \end{pmatrix} Q_{1,1,0}^T \\ &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} (-1) = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \equiv L_{1,0}^T, \end{aligned} \quad (\text{A15})$$

$$V_{1,1,1}^T = \begin{pmatrix} V_{1,0,2}^T \\ V_{1,0,3}^T \\ 0 \end{pmatrix} + \begin{pmatrix} L_{0,2}^T \\ L_{0,3}^T \\ -1 \end{pmatrix} Q_{1,1,1}^T$$

$$= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} (-1) = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \equiv L_{1,1}^T, \quad (\text{A16})$$

and Eq. (32),

$$\begin{aligned} S_{1,0} &= C_{1,0} - B_{1,0} L_{1,0}^T - B_{1,1} L_{1,1}^T \\ &= 1 - (0 \ 1 \ 0) \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} - (1 \ 0 \ 0) \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \\ &= 1. \end{aligned} \quad (\text{A17})$$

Finally updating the precursors $X_{1,0}^{-1}$ and $X_{1,1}^{-1}$ of the inverse of the matrix X using Eqs. (38) and (39) yields the inverse of X as follows:

$$\begin{aligned} X_{2,0}^{-1} &= \begin{pmatrix} X_{1,0}^{-1} & * & * \\ * & X_{1,1}^{-1} & * \\ * & * & 0 \end{pmatrix} + \begin{pmatrix} Y_{1,0}^T L_{1,0} & * & -Y_{1,0}^T \\ * & Y_{1,1}^T L_{1,1} & -Y_{1,1}^T \\ -Y_{1,0} & -Y_{1,1} & S_{1,0}^{-1} \end{pmatrix} \\ &= \begin{pmatrix} 0 & * & 1 & * & * & * & * \\ * & 0 & 1 & * & * & * & 0 \\ 1 & 1 & -1 & * & * & * & * \\ * & * & * & 0 & * & 1 & 0 \\ * & * & * & * & 0 & 1 & * \\ * & * & * & 1 & 1 & -1 & * \\ * & 0 & * & 0 & * & * & 0 \end{pmatrix} \\ &+ \begin{pmatrix} 1 & * & -1 & * & * & * & * \\ * & 0 & 0 & * & * & * & 0 \\ -1 & 0 & 1 & * & * & * & * \\ * & * & * & 0 & * & 0 & 0 \\ * & * & * & * & 1 & -1 & * \\ * & * & * & 0 & -1 & 1 & * \\ * & 0 & * & 0 & * & * & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & * & 0 & * & * & * & * \\ * & 0 & 1 & * & * & * & 0 \\ 0 & 1 & 0 & * & * & * & * \\ * & * & * & 0 & * & 1 & 0 \\ * & * & * & * & 1 & 0 & * \\ * & * & * & 1 & 0 & 0 & * \\ * & 0 & * & 0 & * & * & 1 \end{pmatrix} \equiv X^{-1}. \end{aligned} \quad (\text{A18})$$

The calculated elements in the inverse X^{-1} are found to be consistent with those by conventional methods such as the LU method. It is also noted that one can easily obtain the corresponding elements in the inverse of the original matrix using a table function generated in the nested dissection which converts the row or column index of the structured matrix to the original one.

APPENDIX B: CALCULATION OF SELECTED EIGENSTATES

In this appendix, it is shown that a few eigenstates around a selected energy ξ can be obtained by a similar way with the same computational complexity as in the calculation for the density matrix, though the proposed method directly computes the density matrix without explicitly calculating the eigenvectors.

We compute the few eigenstates around ξ using a block shift-invert iterative method in which the generalized eigenvalue problem of Eq. (2) is transformed as

$$(H - \xi S)^{-1} S c_\nu = \frac{1}{\varepsilon_\nu - \xi} c_\nu. \quad (\text{B1})$$

Then, the following iterative procedure yields a set of eigenstates around ξ as the convergent result:

$$\mathbf{b}_l = (H - \xi S)^{-1} S \mathbf{c}_l, \quad (\text{B2})$$

$$\langle \mathbf{b}_l | \hat{H} | \mathbf{b}_l \rangle \mathbf{c}_{l+1} = \langle \mathbf{b}_l | \hat{S} | \mathbf{b}_l \rangle \mathbf{c}_{l+1} \varepsilon_{l+1}, \quad (\text{B3})$$

where l is the iterative step, ε is a square matrix consisting of diagonal elements, and \mathbf{b} and \mathbf{c} are a set of vectors, where the number of vectors is that of the selected states. The matrix multiplication in Eq. (B2) and the solution of the generalized eigenvalue problem for Eq. (B3) are repeated until convergence, and the convergent \mathbf{c} and the diagonal elements of ε correspond to the eigenstates around ξ . If the number of selected eigenstates is independent of the size of system, the computational cost required for Eq. (B3) is $O(N)$, which arises from the matrix multiplications of $\langle \mathbf{b}_l | \hat{H} | \mathbf{b}_l \rangle$ and $\langle \mathbf{b}_l | \hat{S} | \mathbf{b}_l \rangle$. Therefore, the computational cost of the iterative calculation is governed by the matrix multiplication of $(H - \xi S)^{-1} \mathbf{y}_l^T$ in Eq. (B2), where $\mathbf{y}_l^T = S \mathbf{c}_l$.

Here we show that the matrix multiplication of $(H - \xi S)^{-1} \mathbf{y}_l^T$ can be performed by a similar way with the same computational complexity as in the calculation for the density matrix. As an example of $(H - \xi S)$, let us consider the matrix X given by Eq. (22). After the recurrence calculation of Eqs. (28)–(32), it turns out that the matrix X is factorized as

$$X = L_1 L_0 D L_0^T L_1^T \quad (\text{B4})$$

with matrices defined by

$$D = \begin{pmatrix} A_{0,0} & & & & & & \\ & A_{0,1} & & & & & \\ & & S_{0,0} & & & & \\ & & & A_{0,2} & & & \\ & & & & A_{0,3} & & \\ & & & & & S_{0,1} & \\ & & & & & & S_{1,0} \end{pmatrix},$$

that of the calculation of the inverse matrix. Thus, if the number of selected eigenstates and the number of iterations for convergence are independent of the size of system, the computational effort of calculation of the selected eigenstates

is governed by the recurrence calculation of Eqs. (28)–(32) even for the calculation of selected eigenstates. The scheme may be useful for calculation of eigenstates near the Fermi level.

- ¹R. Car and M. Parrinello, *Phys. Rev. Lett.* **55**, 2471 (1985).
- ²M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, *Rev. Mod. Phys.* **64**, 1045 (1992).
- ³E. R. Davidson, in *Methods in Computational Molecular Physics*, NATO Advanced Study Institute, Series C: Mathematical and Physical Sciences Vol. 113, edited by G. H. F. Diercksen and S. Wilson (Plenum, New York, 1983), p. 95.
- ⁴P. Pulay, *Chem. Phys. Lett.* **73**, 393 (1980).
- ⁵D. M. Wood and A. Zunger, *J. Phys. A* **18**, 1343 (1985).
- ⁶M. P. Teter, M. C. Payne, and D. C. Allan, *Phys. Rev. B* **40**, 12255 (1989).
- ⁷I. Štich, R. Car, M. Parrinello, and S. Baroni, *Phys. Rev. B* **39**, 4997 (1989).
- ⁸G. Kresse and J. Furthmüller, *Phys. Rev. B* **54**, 11169 (1996).
- ⁹A. S. Householder, *J. ACM* **5**, 339 (1958).
- ¹⁰G. H. Golub and C. F. van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore and London, 1996).
- ¹¹S. Goedecker, *Rev. Mod. Phys.* **71**, 1085 (1999), and references therein.
- ¹²S. Goedecker and G. E. Scuseria, *Comput. Sci. Eng.* **5**, 14 (2003).
- ¹³W. Yang, *Phys. Rev. Lett.* **66**, 1438 (1991).
- ¹⁴D. G. Pettifor, *Phys. Rev. Lett.* **63**, 2480 (1989).
- ¹⁵M. Aoki, *Phys. Rev. Lett.* **71**, 3842 (1993).
- ¹⁶T. Ozaki and K. Terakura, *Phys. Rev. B* **64**, 195126 (2001).
- ¹⁷T. Ozaki, *Phys. Rev. B* **74**, 245101 (2006).
- ¹⁸P. Ordejón, D. A. Drabold, M. P. Grumbach, and R. M. Martin, *Phys. Rev. B* **48**, 14646 (1993); P. Ordejón, D. A. Drabold, R. M. Martin, and M. P. Grumbach, *ibid.* **51**, 1456 (1995).
- ¹⁹X.-P. Li, R. W. Nunes, and D. Vanderbilt, *Phys. Rev. B* **47**, 10891 (1993).
- ²⁰D. R. Bowler and T. Miyazaki, *J. Phys.: Condens. Matter* **22**, 074207 (2010).
- ²¹N. D. M. Hine, P. D. Haynes, A. A. Mostofi, C.-K. Skylaris, and M. C. Payne, *Comput. Phys. Commun.* **180**, 1041 (2009).
- ²²K. Varga, *Phys. Rev. B* **81**, 045109 (2010).
- ²³E. Tsuchida, *J. Phys. Soc. Jpn.* **76**, 034708 (2007).
- ²⁴F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta, *Phys. Rev. B* **77**, 085103 (2008).
- ²⁵R. Takayama, T. Hoshi, T. Sogabe, S.-L. Zhang, and T. Fujiwara, *Phys. Rev. B* **73**, 165108 (2006).
- ²⁶M. Ogura and H. Akai, *J. Comput. Theor. Nanosci.* **6**, 2483 (2009).
- ²⁷R. Baer and M. Head-Gordon, *Phys. Rev. B* **58**, 15296 (1998).
- ²⁸A. D. Daniels and G. E. Scuseria, *J. Chem. Phys.* **110**, 1321 (1999).
- ²⁹K. Kitaura, E. Ikeo, T. Asada, T. Nakano, and M. Uebayasi, *Chem. Phys. Lett.* **313**, 701 (1999).
- ³⁰P. Hohenberg and W. Kohn, *Phys. Rev.* **136**, B864 (1964).
- ³¹W. Kohn and L. J. Sham, *Phys. Rev.* **140**, A1133 (1965).
- ³²F. L. Gervasio, P. Carloni, and M. Parrinello, *Phys. Rev. Lett.* **89**, 108102 (2002).
- ³³T. Miyazaki, D. R. Bowler, R. Choudhury, and M. J. Gillan, *Phys. Rev. B* **76**, 115327 (2007); T. Miyazaki, D. R. Bowler, M. J. Gillan, and T. Ohno, *J. Phys. Soc. Jpn.* **77**, 123706 (2008).
- ³⁴K. Nishio, T. Ozaki, T. Morishita, and M. Mikami, *Phys. Rev. B* **77**, 201401(R) (2008); K. Nishio, T. Ozaki, T. Morishita, W. Shinoda, and M. Mikami, *ibid.* **77**, 075431 (2008).
- ³⁵N. Zonias, P. Lagoudakis, and C.-K. Skylaris, *J. Phys.: Condens. Matter* **22**, 025303 (2010).
- ³⁶J. I. Iwata, K. Shiraishi, and A. Oshiyama, *Phys. Rev. B* **77**, 115208 (2008).
- ³⁷Y.-K. Choe, E. Tsuchida, T. Ikeshoji, S. Yamakawa, and S. Hyodo, *Phys. Chem. Chem. Phys.* **11**, 3892 (2009).
- ³⁸D. M. Ceperley and B. J. Alder, *Phys. Rev. Lett.* **45**, 566 (1980); J. P. Perdew and A. Zunger, *Phys. Rev. B* **23**, 5048 (1981).
- ³⁹J. P. Perdew, K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.* **77**, 3865 (1996).
- ⁴⁰L. Lin, J. Lu, L. Ying, R. Car, and Weinan E, *Commun. Math. Sci.* **7**, 755 (2009).
- ⁴¹S. Li, A. Ahmed, G. Klimeck, and E. Darve, *J. Comput. Phys.* **227**, 9408 (2008).
- ⁴²D. E. Petersen, S. Li, K. Stokbro, H. H. B. Sorensen, P. C. Hansen, S. Skelboe, and E. Darve, *J. Comput. Phys.* **228**, 5020 (2009).
- ⁴³K. Takahashi, J. Fagan, and M.-S. Chin, Proceedings of the Eighth PICA Conference, Minneapolis, Minnesota, 1973 (unpublished), pp. 63–69.
- ⁴⁴A. M. Erisman and W. F. Tinney, *Commun. ACM* **18**, 177 (1975).
- ⁴⁵Recently, Varga developed a similar method for quasi-one-dimensional systems in which the density matrix is computed based on a contour integration and the evaluation of the Green's function by making use of a LDL^T factorization (Ref. 22). Although he suggested a possible way to extend his method to higher dimensional systems, the feasibility and the required computational cost remain unclear.
- ⁴⁶T. Ozaki, *Phys. Rev. B* **67**, 155108 (2003); T. Ozaki and H. Kino, *ibid.* **69**, 195113 (2004).
- ⁴⁷E. Tsuchida and M. Tsukada, *Phys. Rev. B* **54**, 7602 (1996).
- ⁴⁸L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S. A. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, and R. Schneider, *J. Chem. Phys.* **129**, 014109 (2009).
- ⁴⁹T. Ozaki, *Phys. Rev. B* **75**, 035123 (2007).
- ⁵⁰A. George, *SIAM J. Numer. Anal.* **10**, 345 (1973).
- ⁵¹When $z_0 = iR_c^2$ and $z_1 = iR_c$, it can be shown that in Eq. (15) $\text{Re}[\lambda G(iR_c^2)] \sim O(\kappa)$ and $\text{Re}[\gamma G(iR_c)] \sim O(\sum_\nu \epsilon_\nu)$, implying that they are not a large number, which is effective to avoid the numerical round-off error in the calculation of Eq. (15).
- ⁵²In a precise sense, we use domain to mean a system that we are now trying to bisect into two *subdomains*. In the recursive dis-

section, we obtain two subdomains after the dissection of the domain. Once we move to each subdomain to perform the next dissection, then the subdomain is called domain in the precise sense. In the text both the terms are distinguished for cases which may cause confusion.

⁵³G. Karypis and V. Kumar, *SIAM J. Sci. Comput. (USA)* **20**, 359 (1999).

⁵⁴T. A. Davis, *Fundamentals of Algorithms*, SIAM Book Series (SIAM, Philadelphia, 2006).

⁵⁵D. E. Muller, *Math. Tables Aids Comput.* **10**, 208 (1956).

⁵⁶X. Wu, *Appl. Math. Comput.* **166**, 299 (2005).

⁵⁷Although the coefficients a , b , and c can be analytically evaluated, the round-off error in the analytically evaluated solution is non-negligible as ΔN converges to zero. In order to avoid the numerical problem, we refine a , b , and c by minimizing a function $F \equiv \sum_{i=0}^2 (a\mu_i^2 + b\mu_i + c - \Delta N_i)^2$ with the analytically evaluated coefficients as initial values. We find that the coefficients refined by the minimization are much more accurate than the

analytic ones for serious cases, and that the refinement is quite effective to avoid the numerical instability.

⁵⁸It is noted that the recurrence formula derived by Lin *et al.* allows us to compute selected elements in $O(N^2)$ operations for 3D systems (Ref. 40), which is superior to our recurrence formulas. However, the size of separators in their way for the nested dissection can be larger than that of our separators especially for the case that basis functions overlap with a number of other basis functions like in the PAO basis functions, which leads to a large prefactor for the computational cost in spite of the lower scaling. Also, the comparison with the algorithm by Takahashi *et al.* (Ref. 43) and Erisman and Tinney (Ref. 44) will be in a future work.

⁵⁹The code, OPENMX, pseudoatomic basis functions, and pseudo-potentials are available on a web site, <http://www.openmx-square.org/>

⁶⁰N. Troullier and J. L. Martins, *Phys. Rev. B* **43**, 1993 (1991).