Complex band structure calculations based on the overbridging boundary matching method without using Green's functions

Shigeru Tsukamoto,[1,*] Tomoya Ono,[2,3,†] Shigeru Iwase,[2] and Stefan Blügel[1]

[1]*Peter Grünberg Institut & Institute for Advanced Simulation, Forschungszentrum Jülich and JARA, D-52425 Jülich, Germany*
[2]*Department of Physics, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan*
[3]*Center for Computational Sciences, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan*

A complex band structure describes the dispersion relation not only of propagating bulk states but also of evanescent ones, both of which are together referred to as generalized Bloch states and are important for understanding the electronic nature of solid surfaces and interfaces. On the basis of the real-space finite-difference formalism within the framework of the density functional theory, we formulate the Kohn-Sham equation for generalized Bloch wave functions as a generalized eigenvalue problem without using any Green's function matrix. By exploiting the sparseness of the coefficient matrices and using the Sakurai-Sugiura projection method, we efficiently solve the derived eigenvalue problem for the propagating and slowly decaying/growing evanescent waves, which are essential for describing the physics of surface/interface states. The accuracy of the generalized Bloch states and the computational efficiency of the present method in solving the eigenvalue problem obtained are compared with those by other methods using the Green's function matrix. In addition, we propose two computational techniques to be combined with the Sakurai-Sugiura projection method and achieve further improvement in the accuracy and efficiency. Complex band structures are calculated with the present method for single- and multiwall carbon nanotubes, and the interwall hybridization and branch points of evanescent electronic states observed in the imaginary parts of the band structures are also discussed.

I. INTRODUCTION

Generalized Bloch states are generalizations of the *propagating* Bloch states constituting the band structure of a crystalline material with translational symmetry and a periodic potential, and include not only the propagating states but also *evanescent* states that decay and grow in magnitude from one unit cell to the next one. Although a propagating Bloch state is characterized by a real wave number, an evanescent state is characterized by a complex wave number. A set of the evanescent states constitutes the complex band structure of a surface or an interface [1], which breaks the translational symmetry in the direction perpendicular to the surface or interface plane. More specifically, the evanescent Bloch states are needed for describing the faint tail of the wave function of a surface or an interface state, which is located within a few layers of the surface or interface and decays depth-wise in the bulk solid [2]. Indeed, the complex band structures and generalized Bloch states are extensively used in various fields of study, such as heterostructures [3], electron transport [4–6], as well as surface and interface [7].

Many aspects have been made to determine the generalized Bloch states on the basis of the density functional theory [8]. In particular, within the frameworks of the tight-binding formalism [9] and real-space finite-difference formalism [10], it can be assumed that the interaction from second-nearest-neighboring unit cells is negligibly small in comparison with that from neighboring ones. Therefore a generalized Bloch wave function ψ with an energy ε satisfies the following three-term one-particle Schrödinger equation [11]:

$$-\mathbf{H}^{(i-1,i)}\psi^{(i-1)} + (\varepsilon - \mathbf{H}^{(i)})\psi^{(i)} - \mathbf{H}^{(i,i+1)}\psi^{(i+1)} = 0, \tag{1}$$

where $\mathbf{H}^{(i)}$ denotes the Hamiltonian of the ith unit cell (see Fig. 1). $\mathbf{H}^{(i-1,i)}$ and $\mathbf{H}^{(i,i+1)}$ are the Hamiltonian representing the interaction between the nearest-neighboring unit cells and are Hermitian conjugate to each other. The matrix elements of $\mathbf{H}^{(i)}$, $\mathbf{H}^{(i-1,i)}$, and $\mathbf{H}^{(i,i+1)}$ are given by $-\frac{1}{2}\frac{d^2}{dr^2} + V$, where the first and second terms denote the kinetic and potential energies, respectively. According to the generalized Bloch theorem [1], the generalized Bloch wave functions in the ith and jth unit cells, $\psi^{(i)}$ and $\psi^{(j)}$, are related by a complex Bloch phase factor λ, as

$$\psi^{(j)} = \lambda^{j-i}\psi^{(i)}. \tag{2}$$

λ is defined as $\lambda = \exp(ikL)$, where k and L are the *complex wave number* of the Bloch state and the length of the unit cell, respectively. In the case of $|\lambda| = 1$, i.e., k has zero imaginary part and is a real number, the generalized Bloch state is a propagating one. On the other hand, if $|\lambda| \neq 1$, i.e., k has a nonzero imaginary part and is a pure imaginary or complex number, the generalized Bloch state is an evanescent one. Note that the decay and growth of the evanescent Bloch states become very sharp when $|\lambda| \ll 1$ or $1 \ll |\lambda|$. The rapid decay and growth
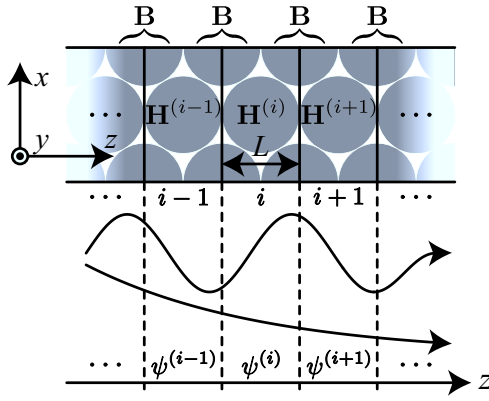
*s.tsukamoto@fz-juelich.de
†ono@ccs.tsukuba.ac.jp

FIG. 1. Schematic representation of crystalline solid with translational symmetry. The unit cells are indexed as $\dots, i-1, i, i+1, \dots$. The matrix $\mathbf{H}^{(i)}$ is the Hamiltonian block corresponding to the ith unit cell, and the matrix $\mathbf{B}$ represent the interaction between the neighboring unit cells. The curves with arrows represent a propagating Bloch wave and decaying evanescent Bloch wave. The wave-function vector ψ is also partitioned by the unit cells.

of evanescent wave functions are known to frequently cause intolerable numerical instability in computations [12].

Stable, accurate, and efficient methods for solving the Schrödinger equation (1) for ψ under the generalized Bloch condition (2) have been extensively studied [12–17]. In 1991, Ando combined Eqs. (1) and (2) and formulated a standard eigenvalue problem for ψ within the framework of the tight-binding formalism [13]. In 2004, Khomyakov and Brocks rewrote the Ando's formulation based on the real-space finite-difference formalism [10] within the framework of the density functional theory [8] and proposed solving a generalized eigenvalue problem for ψ [14]. The eigenvalue problem represented by a small number of basis functions, such as the tight-binding formalism, can be directly solved within a reasonable time. On the other hand, the eigenvalue problem based on the real-space finite-difference formalism has a large degree of freedom; hence, it is difficult to solve directly as it is. Khomyakov and Brocks exploited the sparseness of the interaction matrices $\mathbf{H}^{(i-1,i)}$ and $\mathbf{H}^{(i,i+1)}$, and succeeded in reducing the dimension of the generalized eigenvalue problem. Fujimoto and Hirose also reduced the dimension of the generalized eigenvalue problem in a different way, and proposed the overbridging boundary matching (OBM) method, which first determines the boundary conditions of a generalized Bloch wave function by solving the reduced generalized eigenvalue problem, and then solves the Schrödinger-like Kohn-Sham equation for the generalized Bloch state under the boundary conditions obtained (for more details, see Refs. [12,15]). Both Khomyakov and Brocks and Fujimoto and Hirose achieved large reductions in the computational cost for calculating ψ. In 2014, we improved upon the OBM method to further reduce the dimension of the generalized eigenvalue problem to a requisite minimum using singular-value decomposition of the interaction matrices $\mathbf{H}^{(i-1,i)}$ and $\mathbf{H}^{(i,i+1)}$ [16]. These methods are advantageous in terms of the dimensions of the eigenvalue problems to be solved; however, none avoid calculating the inverse of the Hamiltonian matrix $\mathbf{H}^{(i)}$, i.e.,

the Green's function matrix. This increase the computational cost in constructing the coefficient matrices of the generalized eigenvalue problems to be solved, despite reducing the computational cost in solving the eigenvalue problems.

The solutions of the eigenvalue problems, i.e., ψ, are known to contain rapidly decaying/growing evanescent waves, and the steep evanescent waves can be discarded because some are nonphysical and the others have only negligibly small contribution to the tails of the surface/interface states and scattering wave functions. In other words, only propagating waves and *slowly* decaying/growing evanescent waves are essential in practical calculations, e.g., the calculation of surface and interface states, and the construction of the self-energy of semi-infinite systems.[1] From such a point of view, based on the tight-binding formalism, Sørensen *et al.* derived a small-dimensional eigenvalue problem only for the necessary generalized Bloch states by using the Arnoldi procedure, which is a Krylov subspace method [18]. More specifically, the quadratic eigenvalue problem with respect to λ,

$$-\mathbf{H}^{(i-1,i)}\psi^{(i-1)} + \lambda(\varepsilon - \mathbf{H}^{(i)})\psi^{(i-1)} - \lambda^2\mathbf{H}^{(i,i+1)}\psi^{(i-1)}=0,$$
(3)

which is straightforwardly derived by substituting the Eq. (2) into Eq. (1), is transformed into a smaller quadratic eigenvalue problem with the dimension equal to the number of necessary generalized Bloch wave functions. Recently, we proved that only from propagating waves and *slowly* decaying/growing evanescent waves one can obtain the correct self-energy matrices of semi-infinite systems [19].

We have also proposed the improvement upon the OBM method for reducing the dimension of the generalized eigenvalue problem to the number of necessary generalized Bloch wave functions [16] by using the Sakurai-Sugiura projection method [20]. The Sakurai-Sugiura projection method reduces the dimension of an eigenvalue problem to the number of the eigenvalues located in a specific domain of the complex plane. Recently, we applied it to a quadratic eigenvalue problem (3) represented in the real-space finite-difference formalism by making use of the advantage of the Sakurai-Sugiura projection method on the contour integral for sparse matrices [17,21,22]. The method for solving the quadratic eigenvalue problem is clearly advantageous over the conventional OBM method in terms of avoiding expensive computation of the Green's function matrix and treating sparse coefficient matrices. On the other hand, the conventional OBM method has an advantage over the method for solving the quadratic eigenvalue problem (3), because a linear eigenvalue problem is easier than a nonlinear one. Moreover, more computational libraries

---

[1]The phrase "self-energy" is used not only in electron transport theory but also in many body theory, for instance, in the GW approximation. In the work incorporating the GW approximation with electron transport calculations [6], the phrase "self-energy" is used to refer to "coupling/electrode self-energy" and "GW self-energy." For detailed description on how the phrase "self-energy" used in this paper, see pp. 145–148 in Ref. [4].

are available for linear eigenvalue problems than for nonlinear ones[2] [23].

In this paper, on the basis of the conventional OBM method proposed in 2003 [12] and 2014 [16], we derive another generalized eigenvalue problem for $\boldsymbol{\psi}$, in which no Green's function matrix is used in constructing the coefficient matrices; therefore, the coefficient matrices are kept as sparse as the Hamiltonian matrix. The derived generalized eigenvalue problem is numerically solved by using the Rayleigh-Ritz variation of the Sakurai-Sugiura projection method [24], which is known to be more accurate and robust [25] than the Hankel-matrix variation of the Sakurai-Sugiura projection method [20]. To solve the generalized eigenvalue problem using the Sakurai-Sugiura Rayleigh-Ritz method more accurately and efficiently, we propose two computational techniques, which are combined with the Sakurai-Sugiura method but have almost no impact to the computational time. The computational accuracy and efficiency of the present method are also evaluated in comparison with the method that solves a quadratic eigenvalue problem for generalized Bloch states [17]. The real parts of the complex band structures of single- and multiwall carbon nanotubes obtained with the present method are confirmed through comparison with those obtained with conventional electronic structure calculations. Moreover, the penetration length of the evanescent Bloch states of carbon nanotube is discussed, and the interwall hybridization of the evanescent Bloch states observed in the imaginary part of the complex band structures of multiwall nanotubes is investigated and discussed in terms of the spatial distribution of the evanescent states at the branch point.

II. EIGENVALUE PROBLEM FOR GENERALIZED BLOCH STATES

In this section, we first derive a generalized eigenvalue problem for generalized Bloch states without using the Green's function. Then, we discuss the Sakurai-Sugiura projection method as a suitable eigensolver of the generalized eigenvalue problem, and evaluate the computational cost for solving the generalized eigenvalue problem with the Sakurai-Sugiura method and those with the previously proposed methods. To overcome the problem on computational accuracy originating the Sakurai-Sugiura projection method, we propose two techniques compensating the weakness, which are referred to as the dividing-and-scaling technique and the interchange technique throughout this paper. The computational time of the present method is measured under several conditions, and compared with those obtained by the previously proposed methods. Finally, to confirm the validity of the present method, we calculate the complex band structures of carbon nanotubes, and compare the real parts of the band structures with those obtained with other electronic structure calculations based on the density functional theory.

---

[2]For example, the FEAST eigensolver is currently not capable of treating nonlinear problems. See http://www.feast-solver.org/ for more detail.

A. Derivation of generalized eigenvalue problem

The derivation of the generalized eigenvalue problem starts with the Schrödinger equaiton (1). Since we treat a periodic system with translational symmetry in the z direction, as shown in Fig. 1, the Hamiltonian matrices of the respective unit cells, $\varepsilon\mathbf{I} - \mathbf{H}^{(i)}$ for $i = -\infty, \ldots, +\infty$, are identical, i.e., $\varepsilon\mathbf{I} - \mathbf{H}^{(i)} = \mathbf{A}$ for all i. The interaction matrices between the nearest neighboring unit cells, $\mathbf{H}^{(i-1,i)}$ and $\mathbf{H}^{(i,i+1)}$, are also independent of the index i, and are Hermitian conjugate to each other, i.e., $\mathbf{H}^{(i,i-1)} = [\mathbf{H}^{(i-1,i)}]^\dagger$ for all i. Note that the superscript $\dagger$ denotes the conjugate transpose of a matrix. Within the framework of the real-space finite-difference formalism, the interaction matrix $\mathbf{H}^{(i-1,i)}$ is semilocal; hence, it is expressed as a sparse matrix, which has nonzero entries only at the bottom-left corner:

$$-\mathbf{H}^{(i-1,i)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} \end{bmatrix}. \tag{4}$$

One can more easily see the sparseness of Eq. (1) by rewriting it matrixwise:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}^\dagger & \mathbf{A} & & \mathbf{0} \\ \mathbf{0} & & & \mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}^\dagger & \mathbf{A} \end{bmatrix} \begin{bmatrix} \vdots \\ \boldsymbol{\psi}_{M_\mathbf{B}}^{(i-1)} \\ \boldsymbol{\psi}_{N_\mathbf{B}}^{(i)} \\ \boldsymbol{\psi}_{\mathrm{mid}}^{(i)} \\ \boldsymbol{\psi}_{M_\mathbf{B}}^{(i)} \\ \boldsymbol{\psi}_{N_\mathbf{B}}^{(i+1)} \\ \vdots \end{bmatrix} = \mathbf{0}. \tag{5}$$

Since nonzero entries of the submatrix $\mathbf{B}$ are localized around the diagonal of the matrix on the left-hand side, the dimensions of $\mathbf{B}$ are finite and are defined as $M_\mathbf{B} \times N_\mathbf{B}$. We assume that the spaces under the influence of $\mathbf{B}$ from the $(i-1)$th and $(i+1)$th unit cells do not overlap each other in the ith unit cell, as schematically represented in Fig. 1; i.e., $M_\mathbf{B} + N_\mathbf{B}$ is smaller than the dimension of the square submatrix $\mathbf{A}$, which is referred to as $N_\mathbf{A}$ hereafter. Therefore the generalized Bloch wave-function vector in the ith unit cell, $\boldsymbol{\psi}^{(i)}$, can be divided into three subvectors: the first $N_\mathbf{B}$ entries, the last $M_\mathbf{B}$ entries, and the rest in the middle, as $\boldsymbol{\psi}_{N_\mathbf{B}}^{(i)}$, $\boldsymbol{\psi}_{M_\mathbf{B}}^{(i)}$, and $\boldsymbol{\psi}_{\mathrm{mid}}^{(i)}$, respectively.

Extracting the row block corresponding to the ith unit cell from Eq. (5) being infinite in dimension, we obtain the following finite-dimensional equation for a generalized Bloch wave function:

$$\begin{bmatrix} \mathbf{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}_{N_\mathbf{B}}^{(i)} \\ \boldsymbol{\psi}_{\mathrm{mid}}^{(i)} \\ \boldsymbol{\psi}_{M_\mathbf{B}}^{(i)} \end{bmatrix} = \begin{bmatrix} -\mathbf{B}^\dagger \boldsymbol{\psi}_{M_\mathbf{B}}^{(i-1)} \\ \mathbf{0} \\ -\mathbf{B} \boldsymbol{\psi}_{N_\mathbf{B}}^{(i+1)} \end{bmatrix}, \tag{6}$$

where the terms containing **B** are moved to the right-hand side. According to the generalized Bloch theorem with respect to the neighboring unit cells, $\psi^{(i+1)} = \lambda \psi^{(i)}$, the wavefunction subvectors of the $(i-1)$th and $(i+1)$th unit cells, $\psi_{M_{\mathbf{B}}}^{(i-1)}$ and $\psi_{N_{\mathbf{B}}}^{(i+1)}$, on the right-hand side, can be expressed with λ and corresponding wave function subvectors of the ith unit cell, $\psi_{M_{\mathbf{B}}}^{(i)}$ and $\psi_{N_{\mathbf{B}}}^{(i)}$, respectively. By the application of the generalized Bloch theorem, Eq. (6) can be transformed into an equation with respect to the wave-function vector only of the ith unit cell:

$$[\mathbf{A}] \begin{bmatrix} \psi_{N_{\mathbf{B}}}^{(i)} \\ \psi_{\mathrm{mid}}^{(i)} \\ \psi_{M_{\mathbf{B}}}^{(i)} \end{bmatrix} = \begin{bmatrix} -\mathbf{B}^{\dagger} \frac{1}{\lambda} \psi_{M_{\mathbf{B}}}^{(i)} \\ \mathbf{0} \\ -\mathbf{B}\lambda \psi_{N_{\mathbf{B}}}^{(i)} \end{bmatrix}. \quad (7)$$

Multiplying the both sides of the first $N_{\mathbf{B}}$ rows of the equation above by λ and interchanging the left- and right-hand sides of the first $N_{\mathbf{B}}$ rows, we obtain the following generalized eigenvalue problem for λ and $\psi^{(i)}$:

$$\underbrace{\begin{bmatrix} \mathbf{0} & & \mathbf{B}^{\dagger} \\ & \mathbf{A_2} & \\ & \mathbf{A_3} & \end{bmatrix}}_{\mathbf{\Pi_A}} \underbrace{\begin{bmatrix} \psi_{N_{\mathbf{B}}}^{(i)} \\ \psi_{\mathrm{mid}}^{(i)} \\ \psi_{M_{\mathbf{B}}}^{(i)} \end{bmatrix}}_{\psi} = \lambda \underbrace{\begin{bmatrix} & -\mathbf{A_1} & \\ -\mathbf{B} & & \mathbf{0} \end{bmatrix}}_{\mathbf{\Pi_B}} \underbrace{\begin{bmatrix} \psi_{N_{\mathbf{B}}}^{(i)} \\ \psi_{\mathrm{mid}}^{(i)} \\ \psi_{M_{\mathbf{B}}}^{(i)} \end{bmatrix}}_{\psi},$$

(8)

where the submatrix $\mathbf{A_1}$ is composed of the first $N_{\mathbf{B}}$ rows of $\mathbf{A}$, and the rest of $\mathbf{A}$ is further divided into the submatrices $\mathbf{A_2}$ and $\mathbf{A_3}$ such that $\mathbf{A_3}$ is composed of the last $M_{\mathbf{B}}$ rows of $\mathbf{A}$. Hereafter, the coefficient matrices on the left- and right-hand sides are referred to as $\mathbf{\Pi_A}$ and $\mathbf{\Pi_B}$, respectively. Note that the generalized eigenvalue problem (8) can be further reduced in dimension by using a Green's function submatrix; the formulation is described in Appendix A. The Hamiltonian matrices $\mathbf{H}^{(i)}$ and $\mathbf{H}^{(i-1,i)}$ and the coefficient matrices $\mathbf{\Pi_A}$ and $\mathbf{\Pi_B}$ for a simple case are exemplified in Appendix B.

B. Sakurai-Sugiura Rayleigh-Ritz method

As mentioned in Introduction, we generally do not need all the eigenpair of the generalized eigenvalue problem (8) but only some of the eigenpairs, whose eigenvalues lie in a specific domain of the complex λ plane. For instance, in electron transport calculations, in addition to propagating Bloch waves, one needs evanescent Bloch waves decaying slowly depth-wise in the semi-infinite electrodes, because it is proved that the self-energy matrices of semi-infinite electrodes are correctly calculated only from the propagating waves and slowly decaying evanescent waves [19]. In other words, we need only the eigenvalues just around the unit circle on the complex λ plane, i.e., $|\lambda| \sim 1$, and the eigenvectors belonging to the eigenvalues. For such purpose, the Sakurai-Sugiura projection method [20,24] is highly suitable as an eigensolver. This method projects a function derived from an original eigenvalue problem onto a subspace associated with the eigenvalues located in a given domain of the complex plane by performing the contour integral numerically along

the domain edge. According to the residue theorem, unnecessary eigenpairs are excluded from the newly generated eigenvalue problem. The new eigenvalue problem is small in dimension, and therefore, quickly solvable for the necessary eigenpairs.

In the previously proposed OBM method [16], we have adopted a variation of the Sakurai-Sugiura projection method, which uses a couple of the Hankel matrices composed of the moments of the function derived from an original eigenvalue problem. It is known that the Sakurai-Sugiura Hankel (SS-Hankel) method can be numerically unstable because of explicit use of higher-order moments, and the Hankel matrices become ill-conditioned if eigenvalues in a given domain are close to each other [24,25]. To overcome these drawbacks, another variation of the Sakurai-Sugiura projection method using the Rayleigh-Ritz method was developed. The Sakurai-Sugiura Rayleigh-Ritz (SS-RR) method requires half the order of moment as the SS-Hankel method, and is numerically stable even if eigenvalues are degenerate or close to each other. For more details about the SS-RR and SS-Hankel methods, see Appendix C. In general, the phase factors of generalized Bloch states exist densely around the origin of the complex plane. Therefore the SS-RR method is preferable to the SS-Hankel method. Then, we examine the accuracy of the eigenpairs obtained by solving Eq. (8) with the two variations of the Sakurai-Sugiura projection method. Figures 2(a) and 2(b) show the computational errors as a function of λ for (6,6) and (8,0) carbon nanotubes, respectively. In these calculations, to find the generalized Bloch states with λ in the range of $10^{-2} \leqslant |\lambda| \leqslant 10^{2}$, we set an annular domain with the inner-circle radius $\lambda_{\min} = 10^{-2}$ and the outer-circle radius $\lambda_{\max} = 10^{2}$ on the complex plane. The parameters used in the error evaluation are optimized to keep the computational cost equal. Comparing the computational errors indicated with the open and solid circles, one can easily see that the SS-RR method improves the accuracy of the eigenpairs by three to four orders of magnitude. It should be noted that the computational errors are still large when the absolute value of the phase factor, $|\lambda|$, is small. We address this problem in Sec. II D.

The computational errors indicated by the crosses in Fig. 2, which are obtained by solving the nonlinear eigenvalue problem (3) with the SS-Hankel method, are at the same order as or more than those indicated with the open circles obtained by solving the generalized eigenvalue problem (8) with the SS-Hankel method. One may expect that the computational errors indicated by the crosses can be substantially reduced by switching from the SS-Hankel method to the SS-RR method. Nevertheless, the reduced eigenvalue problem obtained from an original quadratic eigenvalue problem in the SS-RR method is also quadratic. The nonlinear eigenvalue problem is generally more expensive to solve than the generalized one even if the dimension is small; furthermore, there are a few numerical computing libraries for solving quadratic eigenvalue problems. In the light of the purpose of this paper, we do not further discuss solving the quadratic eigenvalue problem (3).

The most time-consuming part of the Sakurai-Sugiura projection method is to calculate the moment vectors $\mathbf{s}_l$ for $l = 0, 1, \ldots, N_{\mathrm{mom}} - 1$, where N_{mom} denotes the number of

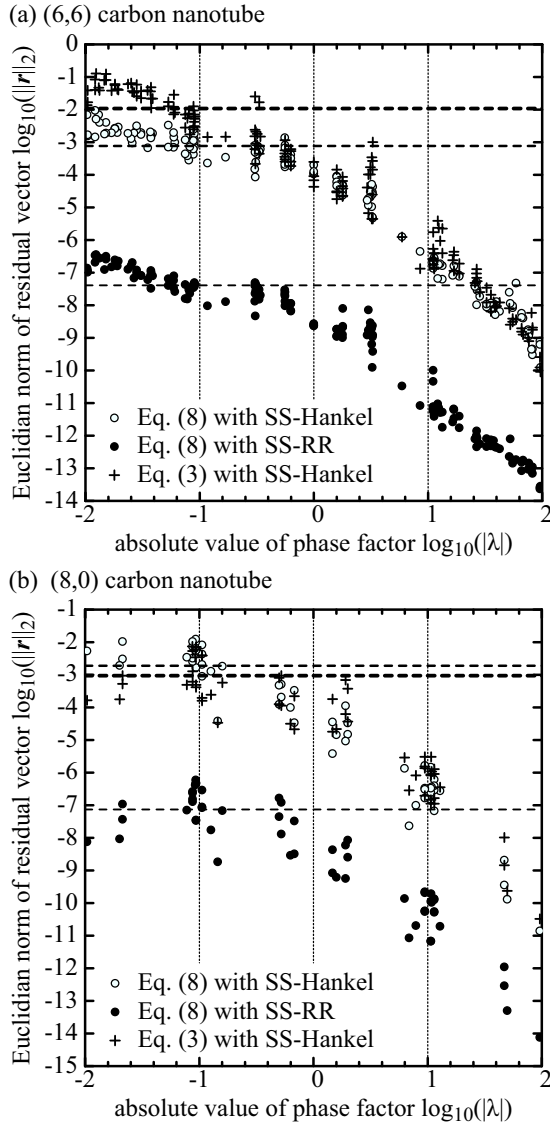## (a) (6,6) carbon nanotube

(b) (8,0) carbon nanotube

FIG. 2. Computational errors in solving eigenvalue problems for generalized Bloch states of carbon nanotubes. Computational errors are evaluated using the Euclidean norms of the residual vectors r, which are obtained by substituting the computed eigenpair into Eqs. (3) and (8). Eigenpairs are calculated using the SS-Hankel and SS-RR methods. The number of sampling points for the numerical contour integral and maximum order of moment used in the SS-RR method, N_{int} and N_{mom}, are set to 32 and 16, respectively. The number of right-hand-side vectors, N_{rhs}, is set to 128 for (6,6) and 56 for (8,0) carbon nanotubes. The thick, normal, and thin dashed horizontal lines represent the averages of the computational errors over the all eigenpairs indicated by the crosses, solid circles, and open circles, respectively.

the moments considered:

$$s_l = \frac{1}{2\pi i} \oint_C z^l (z\Pi_B - \Pi_A)^{-1} u \, dz \qquad (9a)$$

$$\approx \frac{1}{N_{int}} \sum_i^{N_{int}} z_i^{l+1} (z_i\Pi_B - \Pi_A)^{-1} u. \qquad (9b)$$

In practical calculations, the contour integral in Eq. (9a) is discretized and numerically performed according to Eq. (9b). The integrand is evaluated at the discrete sampling points z_i ($i = 1, 2, \ldots, N_{int}$) on the integral path through solving the linear equation:

$$(z_i \Pi_B - \Pi_A)x_i = u. \qquad (10)$$

Here, the constant-term vector u can be chosen arbitrarily. Note that within the scope of the present method, the sampling points $\{z_i\}$ lie on the inner and outer circles of an annular domain, i.e., either $|z_i| = \lambda_{min}$ or $|z_i| = \lambda_{max}$ is hold.

Now, let us consider to apply a shifted conjugate gradient method [26] to solving Eq. (10) for further reduction of the computational cost. Multiplying only the first N_B rows of the linear equations (10) by $1/z_i$, one obtains the following linear equations:

$$\underbrace{\left(-\mathbf{A} - \begin{bmatrix} & & \frac{1}{z_i}\mathbf{B}^\dagger \\ & \mathbf{0} & \\ z_i\mathbf{B} & & \end{bmatrix} \right)}_{\Pi(z_i)} x_i = \underbrace{\begin{bmatrix} \frac{1}{z_i}u_{N_B} \\ u_{mid} \\ u_{M_B} \end{bmatrix}}_{u(z_i)}. \tag{11}$$

where the subvectors u_{N_B} and u_{M_B} on the right-hand side denote the first N_B and last M_B elements of u, respectively. The rest of the elements of u compose the subvector u_{mid}. Since $\mathbf{A} = \varepsilon\mathbf{I} - \mathbf{H}^{(i)}$ as defined in Sec. II A, ε appears only in the diagonal elements of $\Pi(z_i)$. It is known that a Krylov subspace spanned by the products of a matrix $\mathbf{M}$ and a vector v, i.e., $\mathcal{K}(\mathbf{M}, v) \equiv \text{span}\{v, \mathbf{M}v, \mathbf{M}^2v, \ldots\}$, is invariant with respect to the addition of a scalar matrix $\sigma\mathbf{I}$ to $\mathbf{M}$, i.e., $\mathcal{K}(\mathbf{M} + \sigma\mathbf{I}, v) = \mathcal{K}(\mathbf{M}, v)$. This means that once a Krylov subspace is obtained by solving Eq. (11) at a single energy ε using an iterating solver, it can be reused for solving Eq. (11) at different energies $\varepsilon' \neq \varepsilon$. As a consequence of exploiting the general property of the Krylov subspace, we can omit expensive operations of matrix–vector products in iterative solvers for second and subsequent reference energies, and greatly reduce the computational cost for executing the Sakurai-Sugiura projection method [21,27]. This allows us to draw band structures denser in energy.

Within the framework of the real-space finite-difference formalism, $\Pi(z_i)$ is sparse but non-Hermitian; therefore, we choose the shifted biconjugated gradient (BiCG) method [28,29] as the iterative linear-equation solver. It is known that a BiCG solver solves not only the linear equation $\mathbf{Q}x_1 = b_1$ but also $\mathbf{Q}^\dagger x_2 = b_2$ by default for any matrix $\mathbf{Q}$ [28]. From this fact and the property of $\Pi(z_i)$, i.e., $[\Pi(z_i)]^\dagger = \Pi(1/z_i^*)$, it is seen that a BiCG solver solves Eq. (11) for two sampling points z_a and $z_b = 1/z_a^*$ at once [21]. Note that the superscript $*$ denotes the complex conjugate of a scalar. When the inner and outer radii of an annular domain satisfy $\lambda_{max} = 1/\lambda_{min}$, one can simultaneously obtain two solution vectors x_i^{in} and x_i^{out}, the former is for a sampling point on the inner circle z_i^{in} and the latter for the corresponding sampling point on

the outer circle $z_i^{out} = 1/(z_i^{in})^*$. Consequently, one only has to explicitly solve Eq. (11) for the sampling points either z_i^{in} or z_i^{out}. In the case that the computational framework is implemented within the tight-binding and/or localized-basis formalism, $\mathbf{\Pi}(z_i)$ is rather dense. Therefore a direct diagonalization method, such as the LU decomposition, may be advantageous in solving Eq. (11).

C. Computational cost

As mentioned in Sec. II A, the present formulation does not refer to any Green's function matrix, which is the inverse of the matrix in Eq. (5) and is generally expensive to calculate. This is apparently advantageous in computing generalized Bloch wave functions in comparison with other methods that require a Green's function matrix, e.g., previously proposed OBM method [12,16] and the methods proposed by Khomyakov and Brocks [14] and Sørensen *et al.* [18]. Although some of these methods derive and solve a reduced eigenvalue problem instead of Eq. (8), the coefficient matrices treated in the previously proposed methods are dense. On the other hand, Eq. (8) derived in Sec. II A is advantageous in that coefficient matrices $\mathbf{\Pi_A}$ and $\mathbf{\Pi_B}$ are both sparse because of adopting the real-space finite-difference formalism.

In this subsection, we estimate the computational cost for solving Eq. (8), and compare it with those of the OBM methods proposed in Refs. [12,16]. In solving Eq. (8) with the present method and the corresponding eigenvalue problems with the previously proposed methods, we use the Sakurai-Sugiura projection method [20,24]. Since the most time-consuming part of the Sakurai-Sugiura projection method is to solve linear equations at all sampling points on a domain edge, we examine the computational cost for solving the linear equations. As mentioned in Sec. II B, $\mathbf{\Pi}(z_i)$ in Eq. (11) is sparse; hence, we use a BiCG method for solving Eq. (11). On the other hand, the generalized eigenvalue problems to be solved with the previously proposed methods, i.e., Eq. (19) in Ref. [12] and Eq. (32) in Ref. [16], have a couple of dense coefficient matrices, and the coefficient matrix of the linear equations to be solved in the Sakurai-Sugiura projection method is also dense. Therefore we use the LU decomposition for solving the linear equations for the previously proposed methods. In practical computation, the block version of the Sakurai-Sugiura projection method [24] is frequently used, and it solves instead of Eq. (11) the linear equations with multiple right-hand side vectors:

$$\mathbf{\Pi}(z_i)\mathbf{X}_i = \mathbf{U}(z_i), \qquad (12)$$

where $\mathbf{X}_i = [\boldsymbol{x}_{i,1}, \dots, \boldsymbol{x}_{i,N_{rhs}}]$, $\mathbf{U}(z_i) = [\boldsymbol{u}_{i,1}, \dots, \boldsymbol{u}_{i,N_{rhs}}]$ and N_{rhs} is the number of the column vectors. The computational cost for solving Eq. (12) with an iterative method is proportional not only to N_A but also the number of iterations needed until convergency, N_{iter}, and N_{rhs}. On the other hand, the computational cost for solving Eq. (12) with the LU decomposition is proportional only to the cube of the dimension of the linear equation. In Table I, the computational costs of the most time-consuming part in the Sakurai-Sugiura projection method for the present and previously proposed methods are

TABLE I. Estimated computational cost in solving linear equations derived from Eq. (8) of the present work, the nonlinear eigenvalue problem in Ref. [21], Eq. (19) of Ref. [12], and Eq. (32) of Ref. [16]. N_A represents the dimension of $\mathbf{A}$. M_B and N_B represent the numbers of rows and columns of $\mathbf{B}$, respectively. N_{iter} is the averaged number of iterations needed until the solution vectors $\boldsymbol{x}_i$ of Eq. (12) converge with a CG method, and N_{rhs} is the number of right-hand side vectors of Eq. (12).

	linear system solver	dimension of linear system	computational cost
present work Ref. [21]	BiCG	N_A	$\mathcal{O}(N_{iter}N_A N_{rhs})$
Ref. [12]	LU decomp.	$M_B + N_B$	$\mathcal{O}((M_B + N_B)^3)$
Ref. [16]	LU decomp.	rank $\mathbf{B}$	$\mathcal{O}((\text{rank } \mathbf{B})^3)$

summarized. In general, N_{iter} and N_{rhs} are smaller than N_A,[3] and become negligibly small when N_A increases. Therefore the computational cost for the present work can be regarded as $\mathcal{O}(N_A)$.

Now, let us discuss the case in which a system is extended either in the x or y direction (see Fig. 1). When a system size becomes n times larger in one of the directions, the dimension of the linear equations to be solved in each method, N_A, $M_B + N_B$, or rank $\mathbf{B}$, becomes n times larger. According to Table I, the computational cost for the present method becomes only n times larger, while those for the methods in Refs. [12,16] become n^3 times larger. This suggests that the present method has an advantage over the previously proposed methods when the unit cell of a system is extended in the x and y directions.

In the case of a unit cell with large L (see Fig. 1), the present method looks not to be advantageous to the previously proposed methods any more, because according to Table I, the computational cost of the present method depends on the number of the real-space grid points in a unit cell, N_A; however, those of the other methods are independent of the number of the real-space grid points in the z direction. In such a case, the computational cost for calculating the Green's function matrix becomes dominant in the previously proposed methods, and is $\mathcal{O}(N_{iter}N_A(M_B + N_B))$ for the method in Ref. [12] and $\mathcal{O}(N_{iter}N_A \text{ rank } \mathbf{B})$ for the method in Ref. [16]. In general, $M_B + N_B$ and rank $\mathbf{B}$ are much larger than N_{rhs} and N_{int}, for instance, in the calculation of the (6,6) carbon nanotube to be discussed in Sec. II D, $M_B = N_B \sim 26\,000$, rank $\mathbf{B} \sim 21000$, $N_{rhs} = 32$, and $N_{int} = 32$. Therefore we can conclude that the present method is still advantageous to the previously proposed method in terms of the computational cost. If the Green's function of a whole unit cell is constructed from those of the partitioned domains of the unit cell, as single-site Green's functions in the Korringa-Kohn-Rostoker Green function method [30], the computational cost for calculating

---

[3]The number of iterations until convergence is proportional to the square-root of the condition number of a coefficient matrix, and the number of column vectors on the right-hand side considered in the Sakurai-Sugiura eigensolver is usually of the order of ten.

each single-site Green's function can be smaller than that for calculating generalized Bloch waves with the present method.

D. Dividing-and-scaling technique

As mentioned in Sec. II B, we usually set an annular domain on the complex plane to define a contour integral path used in the Sakurai-Sugiura projection method, because we need only propagating and slowly decaying/growing evanescent Bloch waves with λ in a certain range, i.e., $\lambda_{min} < |\lambda| < \lambda_{max}$, where $\lambda_{min} < 1$ and $1 < \lambda_{max}$ are assumed. The computational cost of the Sakurai-Sugiura projection method used in the present method is proportional to N_{rhs}, as discussed in Sec. II C. Besides, since the contour integral in Eq. (9a) is performed numerically according to Eq. (9b), the computational cost is proportional to N_{int} too. It is also clearly seen from Eq. (9b) that N_{int} affects the computational accuracy of s_l (for more detail, see Ref. [20]). The computational accuracy is also affected by the difference in the order of magnitude of s_l, which manifests itself through the calculation of z_i^{l+1} in Eq. (9b). More specifically, numerical errors such as loss of tailoring digits readily occur, when processing a matrix generated from $\{s_l | l = 0, 1, \ldots, N_{mom} - 1\}$ in the Sakurai-Sugiura projection method (for more details, see Appendix C).

The numerical errors deteriorate the computational accuracy. Therefore $\lambda_{min(max)} \ll 1$ or $1 \ll \lambda_{min(max)}$ is not favorable in solving Eq. (8) with the Sakurai-Sugiura projection method accurately. In addition to the deterioration in the accuracy, using an annular domain with $\lambda_{min(max)} \ll 1$ or $1 \ll \lambda_{min(max)}$ causes increase in computational cost. In Fig. 3, the average numbers of BiCG iterations N_{iter} required for solving Eq. (12) for two different carbon nanotubes are plotted as a function of the absolute value of z, which is on the inner or outer circle of a given domains. It is found that the computational cost for solving Eq. (12) increases biquadratically as a function of
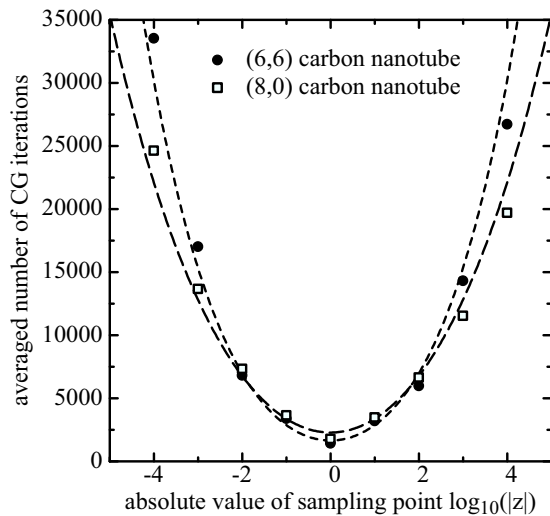

FIG. 3. Average numbers of BiCG iterations until convergency. Equation (12) is solved for two different carbon nanotubes. Data points represented with solid symbols are plotted as a logarithmic function of domain radius $|z|$, and each data set is well fitted by a biquadratic function, as shown in dashed curve.

$\log_{10}(|z|)$. This drastic increase in computational cost through N_{iter} is clearly unfavorable for the present method in comparison to the previously proposed methods [12,16], as one can see from Table I.

To overcome the difficulty in the computational cost and accuracy, we employ the following strategy. The reduction of the computational cost is promoted by subdividing a single annular domain, and the improvement of the computational accuracy by scaling the subdivisions. The set of the procedures is referred to as dividing-and-scaling technique hereafter. First, we divide a single annular domain on the complex plane into multiple annular ones with thinner widths. As an example, let us divide the single annular domain with the inner radius $\lambda_{min} = 10^{-2}$ and the outer radius $\lambda_{max} = \lambda_{min}^{-1} = 10^2$ into four thinner annular subdomains, which have the inner and outer radii

$$\{(\lambda_{min}, \lambda_{max})_m | m = 1, 2, 3, 4\}$$
$$= \{(10^{-2}, 10^{-1}), (10^{-1}, 10^0), (10^0, 10^1), (10^1, 10^2)\}$$
$$(13)$$

like the schematic representation in Fig. 4(a). From the fact that the computational cost for solving Eq. (12) is biquadratically proportional to $\log_{10}(|z|)$ as shown in Fig. 3, N_{iter} for solving Eq. (12) for the subdomains of $m = 2$ and 3 are very small and those for the subdomains of $m = 1$ and 4 are not so much as N_{iter} for the original domain. Therefore the total of N_{iter} for all four subdomains is expected to be much less than four times of N_{iter} for the original domain. The other parameter affecting the computational cost, i.e., N_{rhs}, is able to be smaller when dividing a single domain into smaller subdomains; therefore the computational cost for solving Eq. (12) can be reduced as seen in Table I. More specifically, N_{rhs} can be reduced in inverse proportion to the number of subdomains, e.g., in this case it can be set at one-quarter of N_{rhs} used for the original single domain. This reduction of N_{rhs} is reasonable, because N_{rhs} relates to the dimension of the subspace associated with the eigenvalues located in a given domain, and the number of the eigenvalues in a domain decreases when reducing the domain size. When the distribution of eigenvalues on the complex plane is not uniform, we can choose proper N_{rhs} for each of the subdomains. Consequently, the sum of the product of N_{iter} and N_{rhs} required for solving Eq. (12) for all subdomains is expected to be smaller than that required for solving Eq. (12) for the original single domain, resulting in that the reduction of the total computational cost is achieved when dividing a single annular domain into multiple thinner annular subdomains.

Next, let us discuss scaling each of the annular subdomains for improving the computational accuracy. As mentioned above, the deterioration of the computational accuracy originates from z_i^{l+1} in Eq. (9b), and it becomes significant when $\lambda_{min(max)} \ll 1$ or $1 \ll \lambda_{min(max)}$. Here, we propose to scale each annular subdomains as $\hat{\lambda}_{min(max)} = a\lambda_{min(max)}$, where the scaling factor a is determined so as to satisfy $\hat{\lambda}_{max} = 1/\hat{\lambda}_{min}$. This procedure brings both inner and outer circles to the unit circle on the complex plane as close as possible. Note that the scaling changes Eq. (8) into

$$\mathbf{\Pi_A}\psi = \hat{\lambda}\hat{\mathbf{\Pi}}_\mathbf{B}\psi \qquad (14)$$
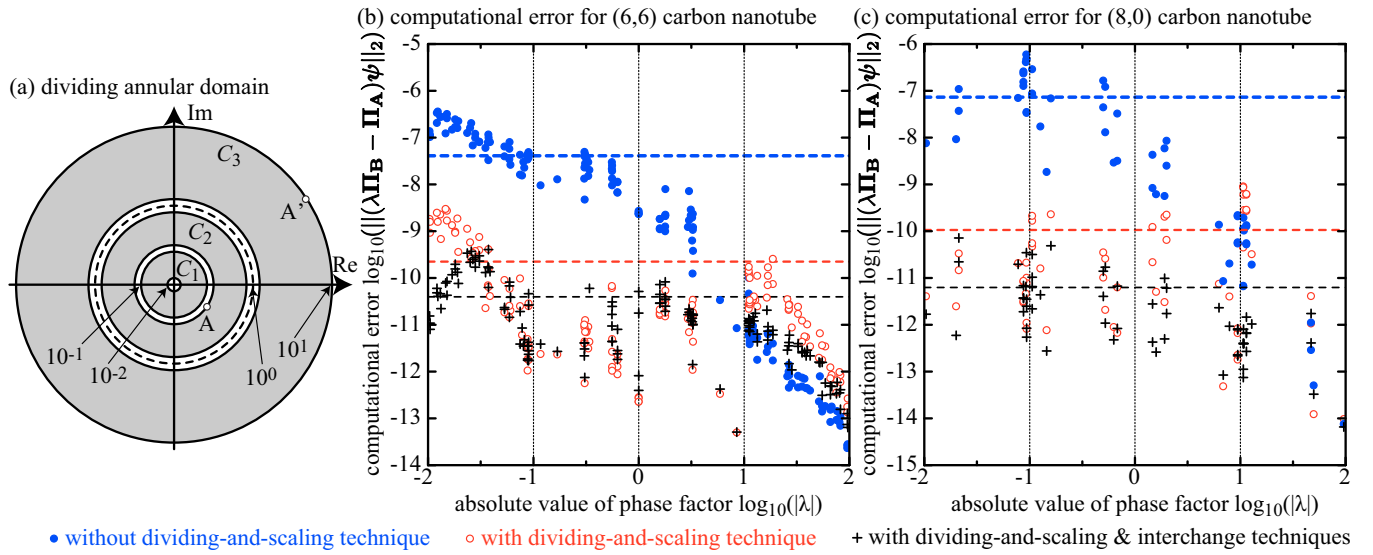
FIG. 4. (a) Schematic representation of dividing an annular domain, and (b) [(c)] computational error in λ and ψ for a (6,6) [(8,0)] carbon nanotube. The open circles in (a) represent sampling points on the integral paths C_2 and C_3, which are $z_{A'} = 1/z_A^*$. The subdomains are represented by the gray areas, and the dashed circle the unit circle on the complex plane. The computational errors are evaluated by the Euclidean norms of the residual vectors of Eq. (8), i.e., $\|(\lambda\Pi_B - \Pi_A)\psi\|_2$. The parameters for the Sakurai-Sugiura projection method are set as $N_{int} = 32$ and $N_{mom} = 16$. N_{rhs} is 128 and 56 for (6,6) and (8,0) carbon nanotubes, respectively. In the case with the dividing-and-scaling technique, N_{rhs} is reduced to 32 and 14 for (6,6) and (8,0) carbon nanotubes, respectively. The thick, normal, and thin dashed horizontal lines represent the averages of the computational errors over the all eigenpairs for the cases without the dividing-and-scaling technique, with the dividing-and-scaling technique, and with the dividing-and-scaling and interchange techniques, respectively.

for each subdomain, where $\hat{\lambda} = a\lambda$ and $\hat{\Pi}_B = \Pi_B/a$. Therefore Eq. (9) also changes to

$$\hat{s}_l = \frac{1}{2\pi i} \oint_{\hat{C}} \hat{z}^l (\hat{z}\hat{\Pi}_B - \Pi_A)^{-1} u \, d\hat{z} \tag{15a}$$

$$\approx \frac{1}{N_{int}} \sum_i^{N_{int}} \hat{z}_i^{l+1} (\hat{z}_i\hat{\Pi}_B - \Pi_A)^{-1} u. \tag{15b}$$

Since the integration variable $\hat{z}$ runs on the inner and outer circle of the scaled subdomain, the difference in the order of magnitude of $\hat{s}_l$ is suppressed in comparison with that of s_l. Therefore processing a matrix generated from $\hat{s}_l$ for $l = 0, 1, \ldots, N_{mom} - 1$ in the Sakurai-Sugiura projection method less suffers from numerical errors and it becomes stable. It should be noted that according to the definitions of $\hat{\lambda}$ and $\hat{\Pi}_B$, the matrix pencil in Eq. (15) is identical to that of Eq. (9), i.e., $\hat{z}_i\hat{\Pi}_B - \Pi_A = z_i\Pi_B - \Pi_A$. Therefore the numerical calculation of $\hat{s}_l$ can be carried out using x_i obtained by solving Eq. (11). In the case of using the block version of the Sakurai-Sugiura projection method, X_i obtained by solving Eq. (12) is used instead of x_i.

As mentioned concerning Eq. (11), it is very advantageous in reducing computational cost to solve Eq. (11) [(12)] with a BiCG method. To exploit the property, an original annular domain and divided annular subdomains have to fulfill the folowings conditions: the inner and outer radii of the original domain satisfy $\lambda_{max} = 1/\lambda_{min}$, the original domain is divided into an even number N_{div} of subdomains, and the inner and outer radii of the mth and $(N_{div} - m + 1)$th subdomains satisfy $(\lambda_{min}, \lambda_{max})_{N_{div}-m+1} = (1/\lambda_{max}, 1/\lambda_{min})_m$. For example, the subdivision shown in Eq. (13) fulfills the conditions. By

employing such a set of annular subdomains, one can reduce the computational time for solving a set of Eq. (11) [(12)] by half, because when solving Eq. (11) [(12)] at the point A in Fig. 4(a), one can also obtain the solution of Eq. (11) [(12)] at the point A' simultaneously, i.e., one only has to solve Eq. (11) [(12)] for $N_{div}/2$ subdomains. To further reduce the computational costs, it should be noted that the integral along an outer (inner) radius of an annular domain can be reused when calculating the integral along the inner (outer) radius of the next annular domain. Thus we can successively reduce the total number of the BiCG iterations thanks to the convergence property of the BiCG method and the reuse of the integral although the product $N_{rhs}N_{div}$ is larger than N_{rhs} of the single domain on k-plane [21] in some cases.

Now let us compare the accuracy of λ and ψ obtained by solving Eq. (8) with and without the dividing-and-scaling technique. The accuracy is evaluated based on the computational errors, which are given as the Euclidean norms of the residual vectors of Eq. (8), i.e., $\|(\lambda\Pi_B - \Pi_A)\psi\|_2$. Bloch waves are described by $\psi(r) = \exp(ik \cdot r)\phi(r)$ with $\phi(r)$ having the same periodicity as the atomic structure of the bulk. We practically compute and normalize $\phi(r)$ so that $\|\phi(r)\|_2 = 1$. In Figs. 4(b) and 4(c), the computational errors for the respective eigenpairs (λ, ψ) are plotted as a function of $|\lambda|$. One can clearly see that the large computational errors observed at the smaller eigenvalues $|\lambda| < 1$ for the case without the dividing-and-scaling technique are suppressed by two orders of magnitude when using the dividing-and-scaling technique. As the consequence, the computational errors obtained with the present method with the dividing-and-scaling technique are below 10^{-8} and 10^{-9} for the (6,6) and (8,0) carbon nanotubes, respectively. The averages of the

computational errors over the all eigenpairs are improved by more than two orders of magnitude for the (6,6) carbon nanotube, and by nearly three orders of magnitude for the (8,0) carbon nanotube. Note that to compare the computational errors with and without the dividing-and-scaling technique under an equivalent computational condition, the number of right-hand-side vectors N_{rhs} used in the block version of the Sakurai-Sugiura projection method with the dividing-and-scaling technique is set at one-quarter of that without the dividing-and-scaling technique, because the single annular domain is divided into four smaller annular ones.

E. Interchange technique

In addition to the propose of the dividing-and-scaling technique discussed in Sec. II D, we also propose another technique to improve the accuracy of $(\lambda, \boldsymbol{\psi})$ in Eq. (8), which is solved within the framework of the Sakurai-Sugiura projection method. The technique is referred to as the interchange technique hereafter. Firstly, we introduce the following auxiliary eigenvalue problem:

$$\boldsymbol{\Pi_B}\boldsymbol{\psi}' = \lambda'\boldsymbol{\Pi_A}\boldsymbol{\psi}'. \tag{16}$$

Comparing the eigenvalue problem with Eq. (8), one can easily see that the coefficient matrices in the left- and right-hand sides, $\boldsymbol{\Pi_A}$ and $\boldsymbol{\Pi_B}$, are interchanged. Therefore $\lambda = 1/\lambda'$ and $\boldsymbol{\psi} = \boldsymbol{\psi}'$. According to the discussion in Sec. II B, in the framework of the Sakurai-Sugiura projection method, Eq. (16) is solved through calculating the moment vectors

$$\boldsymbol{s}'_l = \frac{1}{2\pi i} \oint_{C'} (z')^l (z'\boldsymbol{\Pi_A} - \boldsymbol{\Pi_B})^{-1}\boldsymbol{u}\,dz' \tag{17a}$$

$$\approx \frac{1}{N_{int}} \sum_i^{N_{int}} (z'_i)^l (z'_i\boldsymbol{\Pi_A} - \boldsymbol{\Pi_B})^{-1}\boldsymbol{u} \tag{17b}$$

for $l = 0, 1, \ldots, N_{mom} - 1$, which corresponds to Eq. (9). Note that $z' = 1/z$ because $\lambda = 1/\lambda'$. The contour integral in Eq. (17a) is numerically computed according to Eq. (17b) by solving the following linear equations for each of discrete sampling points z'_i:

$$(z'_i\boldsymbol{\Pi_A} - \boldsymbol{\Pi_B})\boldsymbol{x}'_i = \boldsymbol{u}. \tag{18}$$

Using $z'_i = 1/z_i$, Eq. (18) can be rewritten as

$$(z_i\boldsymbol{\Pi_B} - \boldsymbol{\Pi_A})\left(-\frac{1}{z_i}\boldsymbol{x}'_i\right) = \boldsymbol{u}. \tag{19}$$

This linear equation is equivalent to Eq. (10) if $\boldsymbol{x}'_i = -z_i\boldsymbol{x}_i$, meaning that if one already knows $\boldsymbol{x}_i$, one can easily obtain $\boldsymbol{x}'_i$ without solving Eq. (18).

$(\lambda, \boldsymbol{\psi})$ of Eq. (8) and $(\lambda', \boldsymbol{\psi}')$ of Eq. (16) are obtained with the Sakurai-Sugiura projection method using $\{\boldsymbol{x}_i\}$ and $\{\boldsymbol{x}'_i\}$, respectively. Since $\lambda = 1/\lambda'$ and $\boldsymbol{\psi} = \boldsymbol{\psi}'$, one can easily convert the eigenpairs of Eq. (16) into those of Eq. (8) as $(\lambda', \boldsymbol{\psi}') \leftarrow (1/\lambda', \boldsymbol{\psi}')$. Although the two sets of the eigenpairs, $(\lambda, \boldsymbol{\psi})$ and $(\lambda', \boldsymbol{\psi}')$, are essentially identical to each other, in practice they are not identical because of the numerical errors caused during solving Eqs. (8) and (16) by the Sakurai-Sugiura projection method. Especially, when N_{mom} is large, the numerical error is expected to be large based on the discussion in Sec. II D. Now,

by comparing the Euclidean norms of the residual vectors of Eq. (8), i.e., $\|(\lambda\boldsymbol{\Pi_B} - \boldsymbol{\Pi_A})\boldsymbol{\psi}\|_2$ and $\|(\lambda'\boldsymbol{\Pi_B} - \boldsymbol{\Pi_A})\boldsymbol{\psi}'\|_2$, we can choose either of $(\lambda, \boldsymbol{\psi})$ or $(\lambda', \boldsymbol{\psi}')$ as the solution of Eq. (8), which has less computational error. In this technique, we avoid explicit calculations for solving Eq. (18); thus, the increase in computational time due to the interchange technique is negligible. This technique is universally applicable for any generalized eigenvalue problems to be solved with the Sakurai-Sugiura projection method, because the discussion above does not depend on the properties of the coefficient matrices $\boldsymbol{\Pi_A}$ and $\boldsymbol{\Pi_B}$. Note that the number of the eigenvalues of Eq. (8) in the domain C and that of Eq. (16) in the domain C' are rightly equal. The computer algorithm of the present method based on the SS-RR method incorporating the two computational techniques is described in Appendix D.

In Figs. 4(b) and 4(c), the computational errors obtained for adopting the interchange technique in addition to the dividing-and-scaling technique are plotted. It is clearly seen that when the interchange technique is included, the relatively large computational errors for the case with the dividing-and-scaling technique are suppressed, though the smaller ones do not change so much. Consequently, we achieve to improve the computational errors by roughly one order of magnitude on the average for both the (6,6) and (8,0) carbon nanotubes only with the negligible increase in the computational time. Note that the increase in the computational error observed for some eigenpairs when including the interchange technique seems to occur due to the unreproducible loss of trailing digits during performing the Sakurai-Sugiura projection method.

F. Computational time

In this section, we compare the computational time for solving the eigenvalue problems for generalized Bloch states in an annular domain with the present method with/without the two computational technique as well as with the methods proposed in Refs. [16,21]. The coefficient matrices of Eq. (8) is prepared[4] for a (6,6) carbon nanotube based on the real-space finite-difference formalism [10] within the framework of the density functional theory [8], and Eq. (8) is solved to measure the computational time. The measurement is carried out on a single computer node with two Intel Xeon E5-2667v2 processors with eight cores/processor. Table II shows the computational time averaged per a single energy point as well as the parameters used in the measurement. The parameters listed in Table II, i.e., N_{int}, N_{rhs}, and N_{mom}, are optimized so that the Euclid norm of the residual vector $\|(\lambda\boldsymbol{\Pi_B} - \boldsymbol{\Pi_A})\boldsymbol{\psi}\|_2$ for each eigenpair reaches the tolerance smaller than 10^{-8} and the computational time becomes as small as possible.[5]

---

[4]The effective potential and parameters necessary for constructing $\boldsymbol{\Pi_A}$ and $\boldsymbol{\Pi_B}$ are predetermined by the electronic structure calculations using RSPACE [15,32], which is also based on the real-space finite-difference formalism. The exchange-correlation interaction is treated using the local density approximation [38] within the framework of the density functional theory, and the interaction between valence electrons and nuclei is treated using the norm-conserving pseudopotentials [39].

[5]N_{rhs} is increased by 8, 16, 128, and 1024 for $N_{rhs} \leqslant 32$, $32 < N_{rhs} \leqslant 128$, $128 < N_{rhs} \leqslant 1024$, and $1024 < N_{rhs}$, respectively.

TABLE II. Computational time for calculating generalized Bloch states in an annular domain. The computational time is averaged over the number of the energy points N_ε considered in the shifted CG/BiCG method. $\lambda_{\min(\max)}$ corresponds to the radius of the inner (outer) domain edge. N_{int}, N_{rhs}, and N_{mom} denote the numbers of sampling points on the integral path, of the column vectors in the right-hand side of Eq. (12), and of momenta l considered in Eq. (9), respectively. When the dividing-and-scaling technique is used, an annular domain is divided into two (four) subdomains for $\lambda_{\min} = 0.1$ (0.01). For the method of Ref. [16], the computational time for the LU decomposition is shown in the parentheses, and the rest is almost for the calculations of the Green's function matrix $\mathbf{A}^{-1}$, which are performed using the shifted CG method.

(a) $\lambda_{\min} = 0.1$ and $\lambda_{\max} = 10$

	Sakurai-Sugiura projection method	eigenvalue	DS[a]	interchange	N_{int}	N_{rhs}	N_{mom}	computational time / N_ε (sec.)			
								$N_\varepsilon = 1$	11	51	101
	SS-Hankel + shifted BiCG	λ	no	no	64	128	8	10198	3819	2102	1996
present work			no	no	32	48	16	1912	541	387	390
	SS-RR + shifted BiCG	λ	yes	no	32	2×16	16	939	263	188	182
			yes	yes	32	2×12	16	702	207	146	141
Ref. [16]	SS-RR + LU decomp.	λ	no	no	8	512	4	17423 (3938)	7802 (3938)	6544 (3938)	6956 (3938)
Ref. [21]	SS-Hankel + shifted BiCG	k	no	no	96	32	16	2553	653	498	481

(b) $\lambda_{\min} = 0.01$ and $\lambda_{\max} = 100$

	Sakurai-Sugiura projection method	eigenvalue	DS[a]	interchange	N_{int}	N_{rhs}	N_{mom}	computational time / N_ε (sec.)			
								$N_\varepsilon = 1$	11	51	101
	SS-Hankel + shifted BiCG	λ	no	no	128	2048	4	$\sim$7d.[b]	$\sim$2d.[b]	$\sim$1.5d.[b]	$\sim$1.5d.[b]
present work			no	no	32	384	4	29346	8086	6340	6474
	SS-RR + shifted BiCG	λ	yes	no	32	4×32	16	4370	1223	886	890
			yes	yes	32	4×20	16	2764	787	592	565
Ref. [16]	SS-RR + LU decomp.	λ	no	no	8	1024	4	18637 (4193)	8746 (4193)	7487 (4193)	7423 (4193)
Ref. [21]	SS-Hankel + shifted BiCG	k	no	no	96	128	16	16938	4524	3444	3182

[a]The dividing-and-scaling technique.
[b]Values are estimated from the data at one row below according to the computational cost in Table I.

When the dividing-and-scaling technique is included, an annular domain is divided into two and four thinner annular domains for $\lambda_{\min} = 0.1$ and 0.01, respectively. The BiCG linear-equations solver is parallelized on four MPI processes and four OpenMP threads/process, and the LU decomposition is executed on 16 OpenMP threads. Note that the shifted BiCG solver is implemented in the linear equation solver of the Sakurai-Sugiura projection method for the present method and the method of Ref. [21], and the shifted CG method [27] is implemented in the calculation of the Green's function matrix $\mathbf{A}^{-1}$ for the method of Ref. [16]. The computational time for the LU decomposition is measured only for an single energy point, because it depends only on the matrix dimension and the matrix dimension does not change for any energy points.

Comparing with the methods in Refs. [16,21], the present method improves the computational time for $\lambda_{\min} = 0.1$. However, for $\lambda_{\min} = 0.01$, the present method without the dividing-and-scaling technique is not faster than the method of Ref. [21], even is slower than or comparable to the method of Ref. [16]. More specifically, the computational time of the present method without the dividing-and-scaling technique for $\lambda_{\min} = 0.01$ is approximately 16 times of that for $\lambda_{\min} = 0.1$. In the rest of this paragraph, we discuss the computational time of the present method without the dividing-and-scaling technique. The large increase in the computational time can be explained by the fact that the computational cost for solving the linear equation is proportional to N_{iter} and N_{rhs}, as shown in Table I. By changing $\lambda_{\min}$ from 0.1 to 0.01, not only N_{rhs} increases to eight times, but also N_{iter} increases due to the increase in the distance between a sampling point z on the integral path and the unit circle on the complex λ plane, as seen in Fig. 3. In the case of the method of Ref. [16], the computational time does not strongly depend on $\lambda_{\min}$, because the most of the computational time is spend for calculating the Green's function matrix $\mathbf{A}^{-1}$, and is independent of $\lambda_{\min}$. The dimension of the matrix treated in the LU decomposition is the same for both $\lambda_{\min}$, and it does not affect the computational time, as seen in Table I. Only the difference in N_{rhs} slightly affects the computational time for solving the sets of linear equations with the LU decomposition. In the case of the method of Ref. [21], the computational cost for solving each set of the linear equations derived from a nonlinear eigenvalue problem is proportional to N_{iter}, $N_{\mathbf{A}}$, and N_{rhs}, the same to the present work as shown in Table I. When changing $\lambda_{\min}$ from 0.1 to 0.01, N_{rhs} increases to four times, and N_{iter} is expected to increase as well as the case of the present method. However, the increase in N_{rhs} is not so much as that of the present method, for which N_{rhs} increases to eight times. Because of the relatively small increase in N_{rhs}, the computational time for the method of Ref. [21] does not increase as much as that

for the present method. The less increase in the computational time for the method of Ref. [21] can be understood from the fact that one can keep N_{mom} relatively large and the increase in N_{rhs} small even when λ_{min} is changed, because the contour integral (9) is carried out not on the λ plane but on the k plane, where $k = -i \ln(\lambda)/L$.

Now let us discuss the effect of the dividing-and-scaling technique on the computational time. Comparing the computational time of the present method with and without the dividing-and-scaling technique, one can see that the computational time decreases to half and one-seventh for $\lambda_{min} = 0.1$ and 0.01, respectively. The decrease in the computational time is brought by the decrease in N_{rhs} and N_{iter}. The decrease in N_{rhs} is achieved by dividing an annular domain into subdomains, because the number of the eigenvalues in a single domain decreases. The decrease in N_{iter} is also achieved by scaling the annular subdomains, i.e., by setting the radii of the outer and inner edges of each subdomain as close to the unit circle on the complex λ plane as possible, one can decreases N_{iter}, as seen from Fig. 3. Besides, one can increase N_{mom}, which has no impact on the computational time. The effect on the computational time become drastic when λ_{min} (λ_{max}) is small (large), as seen in Table II.

Table II also shows the computational time of the present method when the interchange technique is used. One can clearly see that the computational time with the interchange technique is close to 75% and 63% of that without the interchange technique for $\lambda_{min} = 0.1$ and 0.01, respectively. Each of the ratios corresponds to the ratio of N_{rhs} of the cases with and without the interchange technique. From the fact we can see that the decrease in the computational time is brought by the decrease in N_{rhs}, and the decrease in N_{rhs} is enabled by the improvement in accuracy brought by the interchange technique

From Table II, one can also see that in all the cases the shifted BiCG linear equation solver works very well. The averaged computational time per single energy point decreases to 20–40% when the number of energy points N_ε is increased from 1 to 101 for both λ_{min}.

G. Complex band structure

Here, we confirm that the present method is able to reproduce the band structures obtained from conventional electronic structure calculations, which shows the energies of propagating electronic states as a function of *real* wave number $\Re(k)$ (referred to as the real part of a complex band structure hereafter). We also show the dispersion relations between the energies of evanescent Bloch states and the *imaginary* wave number $\Im(k)$ (referred to as the imaginary part of a complex band structure hereafter). The complex wave number k is determined through the definition of the phase factor $\lambda = \exp(ikL)$, and λ is obtained by solving Eq. (8), as discussed in the previous subsections. Figure 5 shows the complex band structures of two different single-wall carbon nanotubes as well as the corresponding band structures obtained from conventional electronic structure calculations. The complex band structures are calculated [31] based on the real-space finite-difference formalism [10] within the framework of the density functional theory [8]. The reference band structures are obtained using the electronic structure calculation code
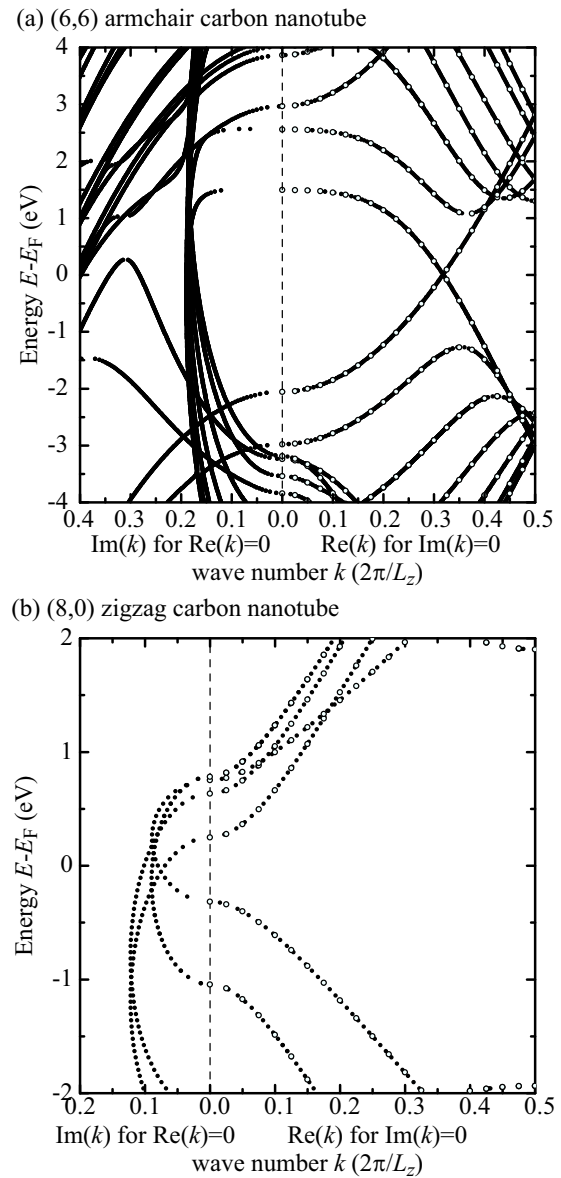

FIG. 5. Complex band structures of (a) (6,6) armchair carbon nanotube and (b) (8,0) zigzag carbon nanotube. Complex band structures composed of black dots are obtained with the present method, and band structures composed of open circles are obtained from conventional electronic structure calculations. E_F denotes the Fermi energy.

RSPACE [15,32]. In each panel of Fig. 5, the real part of the complex band structure is in good agreement with the reference band structure. Moreover, in comparison to the carbon nanotube band structures calculated elsewhere within the framework of the density functional theory [33], the real parts are qualitatively consistent with them. These facts mean that the present method can reproduce band structures properly. The electronic band in the real part smoothly continues into the imaginary part through the Γ point, where the dispersion of the electronic band is zero, as Heine has proposed as a rule for the complex band structure [34]. These bands in the imaginary parts are categorized into the imaginary bands of the first kind, according to Chang's classification [35].

Now let us compare the complex band structures of the (6,6) armchair and (8,0) zigzag carbon nanotubes, and discuss the difference in the behavior of evanescent Bloch states of these nanotubes. An armchair carbon nanotube, such as the (6,6) carbon nanotube, has the Dirac point at the real wave number $\Re(k) = 1/3$, and a large energy gap between the valence and conduction bands at the Γ point in the real part ($\Re(k) = 0$), as shown in Fig. 5(a). Therefore, around the Γ point in the imaginary parts ($\Im(k) = 0$), a large energy gap is present and slowly decaying evanescent Bloch states with small imaginary wave number are forbidden. In contrast, a semiconducting zigzag carbon nanotube, such as the (8,0) carbon nanotube, has a relatively small valence-conduction band gap at the Γ point in the real part, and propagating Bloch states are observed at close to the Fermi energy E_F, as shown in Fig. 5(b). Therefore, in the imaginary part, the electronic bands connecting the valence and conduction bands observed in the real part are found around E_F. The evanescent Bloch states composing the electronic bands in the imaginary part have small imaginary wave number; hence they decay very slowly. Although such evanescent Bloch states are not considered when translational symmetry along the nanotube axis exists, they play a significant role when the translational symmetry is broken, for instance, by cutting a carbon nanotube to form an open end and attaching a cluster to form an interface. If the cluster has an electronic state around E_F, it may hybridize with a slowly decaying evanescent Bloch state of the zigzag carbon nanotube. The hybridized state is principally localized at the cluster, and the wave function tail would deeply penetrate the carbon nanotube. On the other hand, when a cluster is attached at an open end of an armchair carbon nanotube, it is speculated that the hybridized state between a cluster state and an evanescent Bloch state around E_F is more localized around the cluster than in the case of a zigzag carbon nanotube. As a consequence, the physical properties of the localized hybridized states can be affected by the decay length of the evanescent Bloch states of host materials.

III. APPLICATION

In this section, we use the present method to calculate the complex band structures of more complex systems, i.e., a multiwall carbon nanotube composed of (8,0), (17,0), and (26,0) zigzag single-wall carbon nanotubes. In the same way to the practical calculations in the previous section, the complex band structure calculations [31] in this section are based on the real-space finite-difference formalism [10] within the framework of the density functional theory [8], and the reference band structures are obtained using the RSPACE code [15,32]. The complex band structures of the individual (8,0), (17,0), and (26,0) single-wall carbon nanotubes are illustrated in Figs. 5(b), 6(a), and 6(b), respectively. Analogous to the complex band structure of the (8,0) carbon nanotube, each of the complex band structures of the (17,0) and (26,0) carbon nanotubes has a small energy gap between the valence and conduction bands at the Γ point in the real parts, and the imaginary bands of the first kind are observed. The valence and conduction bands are also connected by an imaginary band, which has a small imaginary wave number $\Im(k)$ and is composed of slowly decaying evanescent Bloch states.
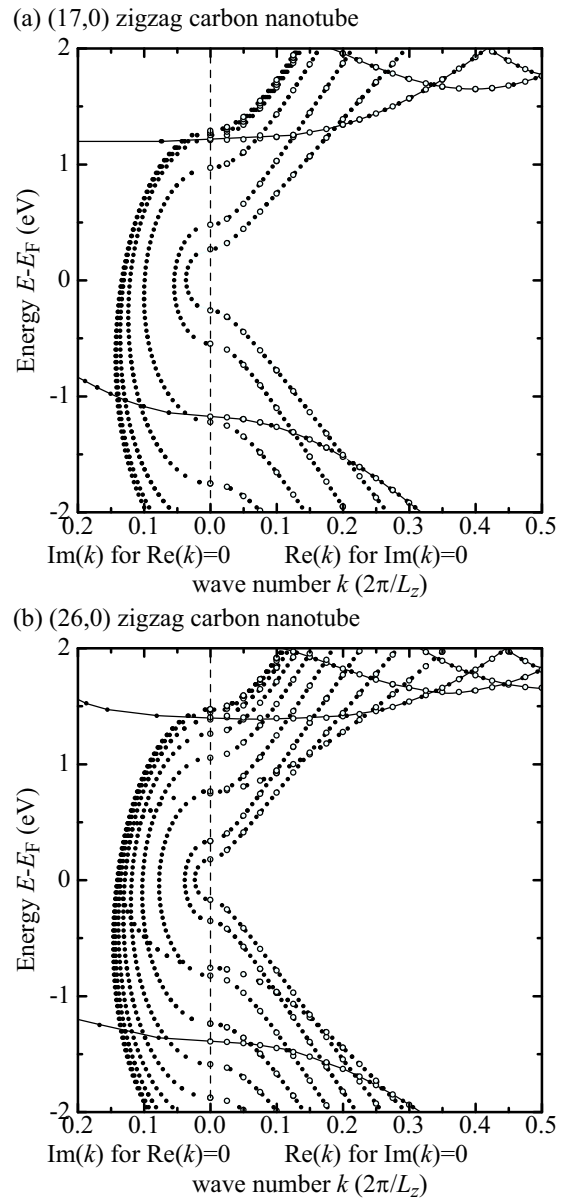

FIG. 6. Complex band structures of single-wall zigzag carbon nanotubes. Solid dots are obtained with the present method, and open circles from conventional electronic structure calculations. (a) and (b) are the complex band structures of (17,0) and (26,0) carbon nanotubes, respectively.

Figure 7(a) shows the complex band structure of the multiwall carbon nanotube composed of the three single-wall carbon nanotubes. In the real part, the band structure looks just like the superimposition of the three band structures of the individual zigzag carbon nanotubes. This is reasonable because the separations between the nanotube walls in the multiwall carbon nanotube are more than 3 Å, which is too large to form hybridization between the π orbitals of different carbon nanotubes. Between the different nanotube walls, the van der Waals interaction is dominant. The band structure in the imaginary part, however, can not be explained straightforwardly only by superimposing the three band structures. In the complex band structure around E_F and the Γ point, which is shown in Fig. 7(b), one can see that the electronic band looks
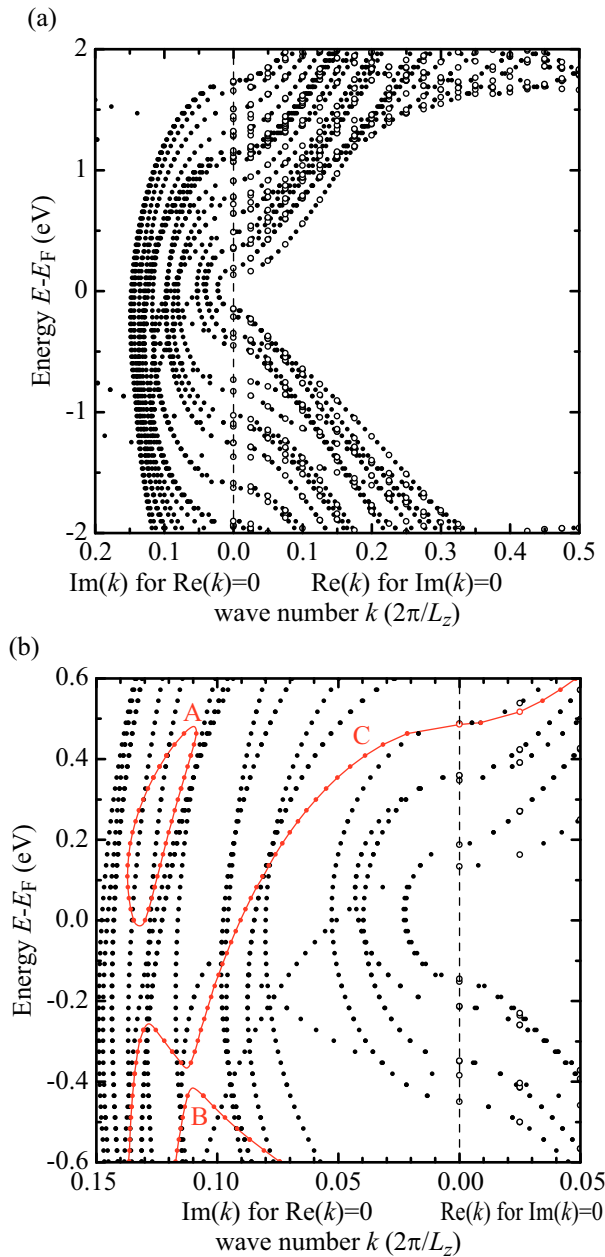
FIG. 7. Complex band structures of multiwall carbon nanotube. Multiwall nanotube is composed of (8,0), (17,0), and (26,0) carbon nanotubes. (b) is an enlarged view around the origin of (a). Solid dots are obtained with the present method, and open circles from conventional electronic structure calculations. Notable data points are in red and are connected with lines for visibility.

significantly different from those of the individual carbon nanotubes depicted in Figs. 5(b), 6(a), and 6(b). More specifically, the electronic band indicated by A in Fig. 7(b) forms a closed loop in the imaginary part, and does not connect any pair of valence and conduction bands in the real part. The electronic band indicated as B in Fig. 7(b) appears to connect two valence bands in the real part. The electronic band indicated as C in Fig. 7(b) has sharp bends looking like disconnections. Consequently, there exist band gaps in the imaginary part, which are not observed in the band structures of the individual carbon nanotubes. These unique behaviors of the electronic

bands and the resultant band gaps in the imaginary part are clearly brought about by the formation of the multiwall carbon nanotube from the individual carbon nanotubes.

To investigate the band gaps in more detail, we use a multiwall heteronanotube composed of (8,0) and (17,0) carbon nanotubes and a (26,0) boron-nitride nanotube. Boron-nitride nanotubes are known to have a relatively large band gap [36], therefore, the electronic bands originating from the (26,0) carbon nanotube, which are present in Fig. 7, are expected to be removed from the energy range considered in this study. Figure 8(a) shows the complex band structure of the heteronanotube. The complex band structure looks simpler than that of the multiwall carbon nanotube, shown in Fig. 7. Compared with the complex band structures of the (8,0) and (17,0) single-wall carbon nanotubes shown in Fig. 8(b), one can see that the complex band structure of the multiwall hetero nanotube is composed of the electronic bands originating only from the two individual carbon nanotubes, and the electronic bands originating from the boron-nitride nanotube are excluded from the energy range shown in Fig. 8(a). The band gap in the imaginary part, which we have pointed out in Fig. 7(b), is also successfully reproduced as indicated by solid and dotted curves in Fig. 8(a), and measures ∼0.6 eV. Because of the opening of the energy gap in the imaginary part, one can see two electronic bands characteristic to the multiwall nanotube: one is the band connecting two conduction bands in the real part, and the other seems to connect two valence bands in the real part, as indicated by the solid and dotted curves in Fig. 8(a). Comparing the complex band structures of the multiwall heteronanotube and each individual single-wall nanotube, we can see that each characteristic band in the imaginary part connects two electronic bands in the real part: one is originating from the (8,0) carbon nanotube and the other is from the (17,0) carbon nanotube. This implies the hybridization of electronic states between different carbon nanotubes.

Let us investigate the hybridization of the electronic states in the imaginary parts in detail. Figure 9 shows the spatial distributions of the wave functions on an xy cross-section, which are indicated as A, B, and C in Fig. 8(a). It should be noted that the wave functions in Fig. 9(b) are multiplied by a phase, $\exp(i\theta)$, so that the wave-function values become real numbers. Electronic states are doubly degenerate at each of A, B, and C on the E-k plane; hence, two different views are depicted for each point, as shown in Figs. 9(a)–9(c). From the spatial distributions in Figs. 9(a) and 9(c), one can see that the generalized Bloch states at A and C are localized at the (8,0) and (17,0) carbon nanotube walls, respectively. Moreover, these states have the same symmetry, i.e., the wave functions in Fig. 9(a) have the three-fold rotational symmetry with respect to the nanotube axis, and the envelop functions of the wave functions in Fig. 9(c) also have the same symmetry, which can be observed in the region between the two nanotube walls. These facts imply that these states can hybridize with each other over the different nanotube walls. Indeed, the intrawall hybridization occurs in the imaginary part of the complex band structure, as indicated by the solid curve in Fig. 8(a), though the two electronic bands do not cross each other in the real part. On the electronic band indicated by the solid curve, one can find the branch point being the most distant from the real axis of the complex band

(a) multi-wall hetero nanotube
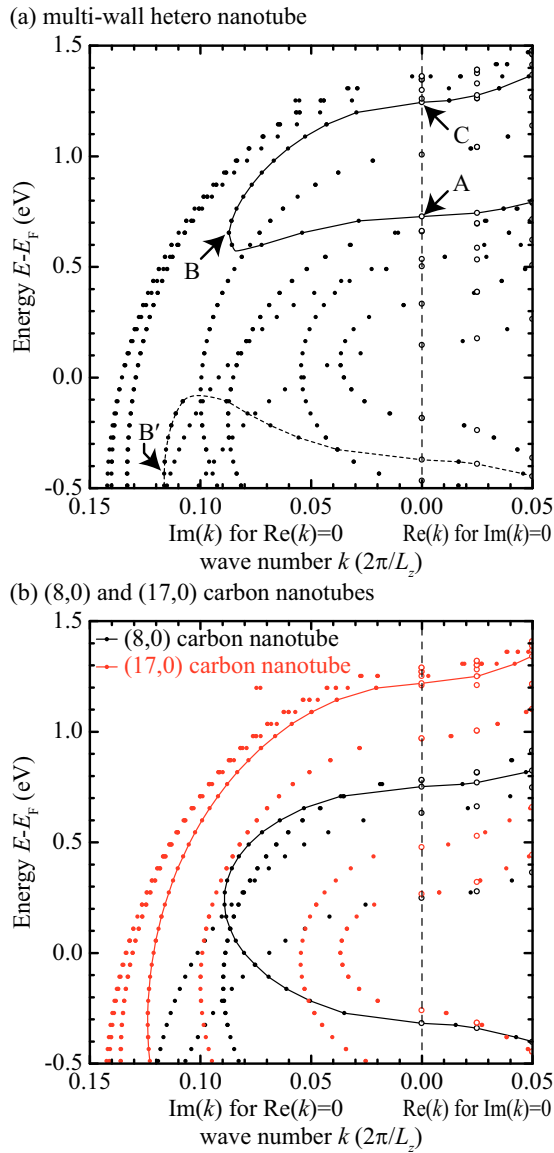

(b) (8,0) and (17,0) carbon nanotubes

FIG. 8. Complex band structures of multiwall heteronanotube and two single-wall carbon nanotubes. The multiwall nanotube is composed of (8,0) and (17,0) carbon nanotubes and (26,0) boron-nitride nanotube. Solid dots are obtained with the present method, and open circles from conventional electronic structure calculations. Some data points are connected with solid and dotted curves for visibility. B and B′ in (a) indicate the branch points of the bands. The spatial distributions of the electronic states at A, B, and C are depicted in Fig. 9.

(a) State A1 & A2

(b) State B1 & B2
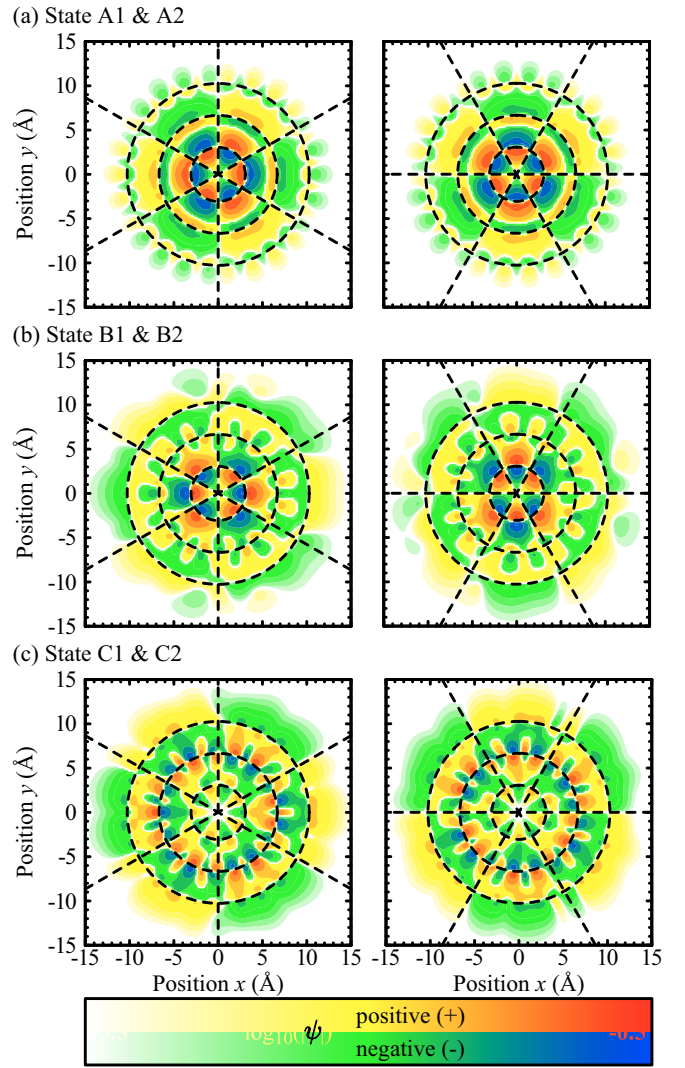
(c) State C1 & C2

FIG. 9. Spatial distributions of generalized Bloch states of multiwall hetero nanotube. In each panel, the wave function values on an xy plane are plotted in logarithmic scale. Note that the wave functions depicted in the panel (b) are multiplied by a phase, $\exp(i\theta)$, so as to become real numbers. (a), (b), and (c) correspond to A, B, and C indicated in Fig. 8(a), respectively. Since the generalized Bloch states in each of (a), (b), and (c) are doubly degenerate, the two wave functions are depicted in each panel. Concentric dashed circles represent (8,0) and (17,0) carbon, and (26,0) boron-nitride nanotube walls. Dashed lines are the guides to see the symmetry of the wave functions.

structure, which is indicated by B in Fig. 8(a). Note that B′ on the dashed curve in Fig. 8(a) indicates another branch point caused by the band-gap opening due to the intrawall hybridization. In the spatial distributions of the generalized Bloch states at the branch point B, shown in Fig. 9(b), one can clearly see the hybridization of the generalized Bloch states shown in Figs. 9(a) and 9(c): the number of the nodes in the circumferential direction of the (8,0) carbon nanotube wall in Fig. 9(b) is the same to that in in Fig. 9(a), and the number of the nodes along the circle of the (17,0) carbon nanotube wall in Fig. 9(b) is the same to that in Fig. 9(c). Comparing

the line profiles of the generalized Bloch states across the multiwall nanotubes, which is drawn in Fig. 10, the feature of the generalized Bloch states at the branch point is more clearly seen. The features observed around the (8,0) nanotube wall in Fig. 10(b) are also visible in Fig. 10(a), and those observed around the (17,0) nanotube wall in Fig. 10(b) are in Fig. 10(c). From this observation, the electronic band in the energy ranges above and below the branch point B is seen to be characterized by the electronic states of (17,0) and (8,0) carbon nanotubes, respectively. This implies that the branch point B is the boundary between the two states, and this is consistent with the explanation of the branch point in Ref. [1].
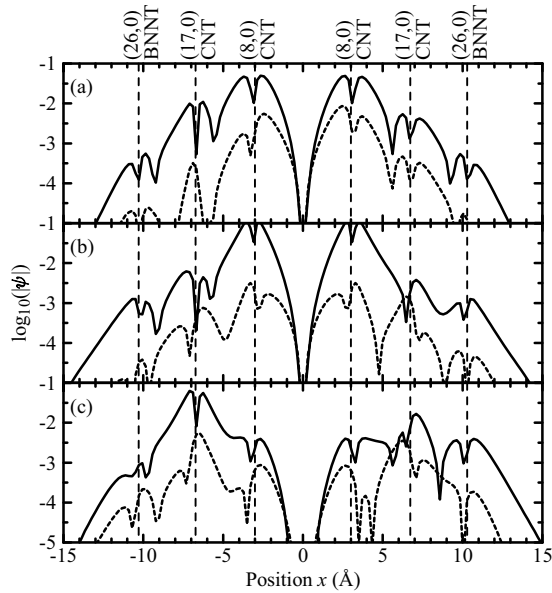
FIG. 10. Line profiles of generalized Bloch states across the multiwall heteronanotube. In each panel, the amplitude of the wave function on the line $y = 0$ in Fig. 9 is plotted in logarithmic scale as a function of the position x. The solid and dotted curves in (a), (b), and (c) represent the line profiles of the left and right panels in Figs. 9(a)–9(c), respectively. The positions of the respective nanotube walls are indicated by the vertical dashed lines.

It should be noted that the branch point at -0.44 eV indicated by B′ in Fig. 8(a) locates within the energy width of the electronic band in the real part drawn by the dotted curves. This is contrast to the point (VII.2) in Ref. [1], which states that branch points appear only in band gaps for one-dimensional materials. This inconsistency can be understood from the fact that the hybridization of the evanescent Bloch states occurs in the plane perpendicular to the nanotube axis.

As a consequence of the band-gap opening, the slowly decaying evanescent Bloch state with the three-fold rotational symmetry is forbidden in the energy gap. More specifically, the interface states with the three-fold rotational symmetry do not appear around E_F, when an end of the multiwall nanotube is attached to a solid surface to form a heterojunction. This is in contrast to the case of the (8,0) and (17,0) individual carbon nanotubes; such evanescent Bloch states are allowed to exist in the energy gap, as shown in Fig. 8(b). At the interface between an end of the single-wall nanotube and a solid surface, the interface states with the threefold rotational symmetry can exist around E_F, and affect the electronic properties of the interface. Consequently, we conclude that when a solid surface is attached to an end of a multiwall nanotube, the electronic structure at the interface is not the same to the superimposition of the electronic structures of the interfaces between the solid surface and the individual single-wall nanotubes.

IV. CONCLUSION

On the basis of the OBM method within the framework of the density-functional real-space finite-difference formal-

ism, we derived a generalized eigenvalue problem for generalized Bloch states, which does not include any Green's function matrix in the coefficient matrices; hence the coefficient matrices are sparse. The Rayleigh-Ritz variation of the Sakurai-Sugiura projection method, one of the more accurate and stable variations, was applied to solving the generalized eigenvalue problem for finding only propagating states and slowly decaying/growing evanescent states, which are essential for describing the physics of surface/interface states and constructing the self-energy of semi-infinite systems. To overcome the disadvantages in computational accuracy and cost in comparison to the previously proposed methods, we also proposed the dividing-and-scaling technique and the interchange technique. Consequently, the derived generalized eigenvalue problem was proven to be accurately and efficiently solved only for necessary generalized Bloch states by means of the Rayleigh-Ritz variation of the Sakurai-Sugiura projection method incorporating the two computational techniques. Throughout the comparison of the computational time measured with the present method to those with the method proposed previously, we showed that the present method with the two computational techniques is advantageous to the conventional methods in terms of the computational time. The real parts of the complex band structures calculated with the present method are in good agreement with the band structures calculated from conventional electronic structure calculations. The imaginary part of the complex band structures obtained with the present method is also consistent with previous work on the general properties of complex band structures. Applying the present method to more practical calculations of the complex band structure of multiwall carbon/heteronanotubes, we investigated the change in the band structure from individual single-wall carbon nanotubes. Although the interaction between the different nanotube walls is dominated by the van der Waals interaction and no significant change is observed in the real part of the complex band structure, the imaginary bands are significantly different in comparison to those of the individual single-wall carbon nanotubes. This is caused by the hybridization of evanescent Bloch states in the imaginary part of the complex band structure. Consequently, the application calculations reveal that multiwall carbon/hetero nanotubes have specific evanescent Bloch states and complex band structures, which are different from those of the individual single-wall nanotubes composing the multiwall nanotubes. The material-specific evanescent Bloch states are obviously important for designing interface/surface electronic structures in the field of band engineering, and the present method can contribute to understand interface/surface states of large-scale systems.

carried out using the system B of the Institute for Solid State Physics at the University of Tokyo, the COMA of the Center for Computational Sciences at University of Tsukuba, and the K computer provided by the RIKEN Advanced Institute for Computational Science through the HPCI System Research project (Project ID: hp160228).

S.T. and T.O. equally contributed to this work.

APPENDIX A: REDUCING DIMENSION OF EQ. (8)

Although we presented in the main text the formulation to obtain the generalized Bloch wave functions without calculating the Green's function, in this section we introduce a variation of the formulation to reduce the dimension of the generalized eigenvalue problem (8) using of the inverse of a small block of the matrix in Eq. (5). From the second row block of Eq. (8), one can see that the wave-function subvector $\boldsymbol{\psi}_{\text{mid}}^{(i)}$ is expressed as

$$\boldsymbol{\psi}_{\text{mid}}^{(i)} = \mathbf{A}_{22}^{-1}\left(\mathbf{A}_{21}\boldsymbol{\psi}_{N_{\mathbf{B}}}^{(i)} + \mathbf{A}_{23}\boldsymbol{\psi}_{M_{\mathbf{B}}}^{(i)}\right) \tag{A1}$$

without the phase factor λ, because the right-hand side of the second row block of Eq. (8) is zero. The matrices $\mathbf{A}_{21}$ and $\mathbf{A}_{23}$ are composed of the first $N_{\mathbf{B}}$ and last $M_{\mathbf{B}}$ column vectors of $\mathbf{A}_2$, respectively. The matrix $\mathbf{A}_{22}$ is composed of the rest of the column vectors of $\mathbf{A}_2$, and is, therefore, square. $\mathbf{A}_{22}$ is invertible because it is a full-rank. The inverse of $\mathbf{A}_{22}$ can be efficiently calculated by using the shifted conjugate gradient method [26] and/or constructed from the inverses of the diagonal block of $\mathbf{A}_{22}$ [30].

Substituting Eq. (A1) into the first and third row blocks of Eq. (8), we obtain the following generalized eigenvalue problem only for subvectors $\boldsymbol{\psi}_{N_{\mathbf{B}}}^{(i)}$ and $\boldsymbol{\psi}_{M_{\mathbf{B}}}^{(i)}$:

$$\begin{bmatrix} \mathbf{0} & -\mathbf{B}^{\dagger} \\ \boldsymbol{\Pi}_{31} & \boldsymbol{\Pi}_{33} \end{bmatrix}\begin{bmatrix} \boldsymbol{\psi}_{N_{\mathbf{B}}}^{(i)} \\ \boldsymbol{\psi}_{M_{\mathbf{B}}}^{(i)} \end{bmatrix} = \lambda\begin{bmatrix} \boldsymbol{\Pi}_{11} & \boldsymbol{\Pi}_{13} \\ -\mathbf{B} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\psi}_{N_{\mathbf{B}}}^{(i)} \\ \boldsymbol{\psi}_{M_{\mathbf{B}}}^{(i)} \end{bmatrix}, \tag{A2}$$

where $\boldsymbol{\Pi}_{ij} = \mathbf{A}_{ij} - \mathbf{A}_{i2}\mathbf{A}_{22}^{-1}\mathbf{A}_{2j}$ for $i, j = 1, 3$. The dimension of the generalized eigenvalue problem is reduced to $M_{\mathbf{B}} + N_{\mathbf{B}}$. The matrices $\mathbf{A}_{11}$ and $\mathbf{A}_{13}$ ($\mathbf{A}_{31}$ and $\mathbf{A}_{33}$) are composed of the first $N_{\mathbf{B}}$ and last $M_{\mathbf{B}}$ column vectors of the matrix $\mathbf{A}_1$ ($\mathbf{A}_3$), respectively. The matrix $\mathbf{A}_{12}$ ($\mathbf{A}_{32}$) is composed of the rest of the column vectors of $\mathbf{A}_1$ ($\mathbf{A}_3$). It should be noted that the dimension of Eq. (A2) is reduced to $2 \times \text{rank }\mathbf{B}$ when the singular-value decomposition of the interaction matrix $\mathbf{B}$ is introduced, as discussed in Ref. [16].

APPENDIX B: EXAMPLE OF HAMILTONIAN MATRICES

In this section, we exemplify the Hamiltonian matrices $\mathbf{H}^{(i)}$ and $\mathbf{H}^{(i-1,i)}$ for a one-dimensional system based on the real-space finite-difference formalism within the framework of the density functional theory, and derive a corresponding generalized eigenvalue problem to be solved. Adopting the central difference formula for the kinetic energy operator[6] and assuming a local effective potential $v(z)$, $\mathbf{H}^{(i)}$ is expressed as

---

[6]This corresponds to $N = 1$ in Ref. [10].

a tridiagonal matrix and $\mathbf{H}^{(i-1,i)}$ has only one nonzero element at the bottom-left corner:

$$\mathbf{H}^{(i)} = \begin{bmatrix} \alpha_1 & \beta & & 0 \\ \beta & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta \\ 0 & & \beta & \alpha_{N_{\mathrm{A}}} \end{bmatrix} \tag{B1}$$

and

$$\mathbf{H}^{(i-1,i)} = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & & & \vdots \\ \beta & 0 & \cdots & 0 \end{bmatrix} \tag{B2}$$

with $\alpha_i = \frac{1}{d^2} + v_i$ and $\beta = -\frac{1}{2d^2}$, where d is the grid spacing and v_i denotes the effective potential at $z = z_i$. From Eq. (B2), one can see that $M_{\mathbf{B}} = N_{\mathbf{B}} = 1$. According to the discussion in Sec. II A, the coefficient matrices of the generalized eigenvalue problem (8) are obtained as

$$\boldsymbol{\Pi}_{\mathbf{A}} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & | & \beta \\ \beta & \alpha_2 & \beta & & & 0 \\ & \ddots & \ddots & \ddots & & \\ 0 & & \beta & \alpha_{N_{\mathrm{A}}-1} & \beta \\ 0 & \cdots & 0 & \beta & \alpha_{N_{\mathrm{A}}} \end{bmatrix} \tag{B3}$$

and

$$\boldsymbol{\Pi}_{\mathbf{B}} = \begin{bmatrix} -\alpha_1 & -\beta & 0 & \cdots & 0 \\ \hline 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & & & \vdots \\ \hline -\beta & 0 & \cdots & \cdots & 0 \end{bmatrix}. \tag{B4}$$

Note that the partitions in the matrices correspond to those in Eq. (8). The Hamiltonian matrices $\mathbf{H}^{(i)}$ and $\mathbf{H}^{(i-1,i)}$ for three-dimensional systems are exemplified in Eqs. (13) and (A6) of Ref. [12], respectively. The Hamiltonian matrices with nonlocal potentials are also discussed in Sec. II A of Ref. [16].

APPENDIX C: THE SAKURAI-SUGIURA PROJECTION METHOD

In this section, we briefly introduce the Sakurai-Sugiura projection method that is employed for solving Eq. (8). The Sakurai-Sugiura projection method is classified as an eigensolver based on a contour integral [20,23–25,37] that finds eigenvalues only in a given domain on the complex plane and the eigenvectors belonging to the eigenvalues. In general, the contour-integral-based eigensolvers filter out the information of unnecessary eigenpairs by performing contour integrals along the edge of the given domain, and generate a subspace associated with the eigenvectors belonging to the necessary eigenvalues. Among various contour-integral-based eigensolvers, the Sakurai-Sugiura projection method is characterized by generation of the subspace, i.e., the subspace is spanned by a set of the moment vectors $\boldsymbol{s}_l$ defined by Eq. (9),

$\mathbf{S} = [\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_{\mathrm{mom}}-1}]$, which is generated from a single arbitrary vector $\boldsymbol{u}$.

In the Rayleigh-Ritz variation of the Sakurai-Sugiura projection (SS-RR) method [24], $\mathbf{S}$ is processed by the singular-value decomposition,

$$\mathbf{S} = \mathbf{U_S} \boldsymbol{\Sigma_S} \mathbf{V_S^\dagger}, \tag{C1}$$

to find rank $\mathbf{S}$, which corresponds to the number of the eigenvalues in the given domain, i.e, $N_{\mathrm{eig}} = \mathrm{rank}\,\mathbf{S}$. This is easily understood by the fact that according to the residue theorem the singular values corresponding to the eigenpairs outside of the given domain are suppressed by preforming the contour integral of Eq. (9) [20]. $\boldsymbol{\Sigma_S}$ is the N_{eig}-dimensional diagonal matrix composed of the singular values, and $\mathbf{U_S}$ ($\mathbf{V_S}$) is composed of N_{eig} orthonormalized column vectors spanning the column (row) space of $\mathbf{S}$. Therefore $\mathbf{U_S}$ can be used as the subspace that is necessary and sufficient to express the eigenpairs one wants. By projecting the generalized eigenvalue problem (8) onto $\mathbf{U_S}$, the reduced eigenvalue problem is obtained as

$$\widetilde{\boldsymbol{\Pi}}_\mathbf{A} \widetilde{\boldsymbol{\psi}}_{\mathrm{RR}} = \lambda \widetilde{\boldsymbol{\Pi}}_\mathbf{B} \widetilde{\boldsymbol{\psi}}_{\mathrm{RR}}, \tag{C2}$$

where $\widetilde{\boldsymbol{\Pi}}_\mathbf{A} = \mathbf{U_S^\dagger} \boldsymbol{\Pi}_\mathbf{A} \mathbf{U_S}$ and $\widetilde{\boldsymbol{\Pi}}_\mathbf{B} = \mathbf{U_S^\dagger} \boldsymbol{\Pi}_\mathbf{B} \mathbf{U_S}$. The eigenvector of Eq. (8) is expressed by the linear combination of the column vectors of $\mathbf{U_S}$, i.e., $\boldsymbol{\psi} = \mathbf{U_S} \widetilde{\boldsymbol{\psi}}_{\mathrm{RR}}$.

On the other hand, the Hankel-matrix variation of the Sakurai-Sugiura (SS-Hankel) method [20] uses the following pair of Hankel matrices:

$$\mathbf{H}_{\mathrm{Hankel}} = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{N_{\mathrm{mom}}-1} \\ \mu_1 & \mu_2 & & \mu_{N_{\mathrm{mom}}} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{N_{\mathrm{mom}}-1} & \mu_{N_{\mathrm{mom}}} & \ddots & \mu_{2N_{\mathrm{mom}}-2} \end{bmatrix} \tag{C3}$$

and

$$\mathbf{H}_{\mathrm{Hankel}}^< = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_{N_{\mathrm{mom}}} \\ \mu_2 & \mu_3 & & \mu_{N_{\mathrm{mom}}+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{N_{\mathrm{mom}}} & \mu_{N_{\mathrm{mom}}+1} & \ddots & \mu_{2N_{\mathrm{mom}}-1} \end{bmatrix}, \tag{C4}$$

where μ_l is the lth-order moment of the scalar-valued function $\boldsymbol{u}^\dagger (z\boldsymbol{\Pi}_\mathbf{B} - \boldsymbol{\Pi}_\mathbf{A})^{-1} \boldsymbol{u}$, i.e., $\mu_l = \boldsymbol{u}^\dagger \boldsymbol{s}_l$. It is known that the generalized eigenvalue problem composed of $\mathbf{H}_{\mathrm{Hankel}}$ and $\mathbf{H}_{\mathrm{Hankel}}^<$,

$$\mathbf{H}_{\mathrm{Hankel}}^< \boldsymbol{\psi}_{\mathrm{Hankel}} = \lambda \mathbf{H}_{\mathrm{Hankel}} \boldsymbol{\psi}_{\mathrm{Hankel}}, \tag{C5}$$

also give the eigenvalues in the given domain on the complex plane, which are identical to those of Eq. (8); therefore $N_{\mathrm{eig}} = \mathrm{rank}\,\mathbf{H}_{\mathrm{Hankel}} = \mathrm{rank}\,\mathbf{S}$. Note that $\mathbf{H}_{\mathrm{Hankel}}$ and $\mathbf{H}_{\mathrm{Hankel}}^<$ can be rank-deficient, i.e., $N_{\mathrm{eig}} < N_{\mathrm{mom}}$, because we are not able to know the number of the eigenvalue in the given domain in advance. Now, let us consider reducing the dimension of Eq. (C5) to N_{eig}. Analogous to the case of the SS-RR method, using the singular-value decompostion $\mathbf{H}_{\mathrm{Hankel}}$ is expressed as a product of three matrices:

$$\mathbf{H}_{\mathrm{Hankel}} = \mathbf{U_H} \boldsymbol{\Sigma_H} \mathbf{V_H^\dagger}. \tag{C6}$$

Here, $\boldsymbol{\Sigma_H}$ is the N_{eig}-dimensional diagonal matrix composed of the singular values, and $\mathbf{U_H}$ ($\mathbf{V_H}$) is composed of N_{eig} orthonormalized column vectors spanning the column (row) space of $\mathbf{H}_{\mathrm{Hankel}}$. Substituting Eq. (C6) for $\mathbf{H}_{\mathrm{Hankel}}$ in Eq. (C5), and operating matrices so that the generalized eigenvalue

problem (C5) changes to a standard eigenvalue problem, one obtains

$$\underbrace{\boldsymbol{\Sigma}_\mathbf{H}^{-1/2} \mathbf{U_H^\dagger} \mathbf{H}_{\mathrm{Hankel}}^< \mathbf{V_H} \boldsymbol{\Sigma}_\mathbf{H}^{-1/2}}_{\widetilde{\mathbf{H}}_{\mathrm{Hankel}}} \widetilde{\boldsymbol{\psi}}_{\mathrm{Hankel}} = \lambda \widetilde{\boldsymbol{\psi}}_{\mathrm{Hankel}}, \tag{C7}$$

where $\boldsymbol{\psi}_{\mathrm{Hankel}} = \mathbf{V_H} \boldsymbol{\Sigma}_\mathbf{H}^{-1/2} \widetilde{\boldsymbol{\psi}}_{\mathrm{Hankel}}$. Now, Eq. (C5) is reduced to the N_{eig}-dimensional standard eigenvalue problem with respect to $\widetilde{\mathbf{H}}_{\mathrm{Hankel}}$. Consequently, the eigenvector of Eq. (8) is given as $\boldsymbol{\psi} = \mathbf{S} \boldsymbol{\psi}_{\mathrm{Hankel}}$. From Eqs. (C3) and (C4), it is clearly seen that to construct $\mathbf{H}_{\mathrm{Hankel}}$ and $\mathbf{H}_{\mathrm{Hankel}}^<$ large enough to contain the information of all eigenpairs in the given domain, one has to calculate the moment vectors $\boldsymbol{s}_l$ up to $l = 2N_{\mathrm{mom}} - 1$. The maximum order of $\boldsymbol{s}_l$ processed in the SS-Hankel method is larger than that in the SS-RR method; therefore the deterioration of computational accuracy may occur easily, as discussed in Secs. II B and II D.

Algorithm 1

1: **input** $\boldsymbol{\Pi}_\mathbf{A}$, $\boldsymbol{\Pi}_\mathbf{B}$, $\lambda_{\min}$, N_{int}, N_{mom}, and N_{div}

2: assume $\lambda_{\min} < 1$ and $N_{\mathrm{div}} = 2n$ for $n \in \mathbb{N}$

3: $\lambda_{\max} \leftarrow 1/\lambda_{\min}$, $N_{\mathrm{eig}} \leftarrow 0$, and $\{(\lambda_i, \boldsymbol{\psi}_i)\} \leftarrow \emptyset$

4: **for** $j = 1, \ldots, N_{\mathrm{div}}/2$ **do**

5:     $\lambda_{\min}^{\mathrm{in}} \leftarrow 10^{\frac{\log_{10}(\lambda_{\max}) - \log_{10}(\lambda_{\min})}{N_{\mathrm{div}}}(j-1) + \log_{10}(\lambda_{\min})}$

6:     $\lambda_{\max}^{\mathrm{in}} \leftarrow 10^{\frac{\log_{10}(\lambda_{\max}) - \log_{10}(\lambda_{\min})}{N_{\mathrm{div}}}j + \log_{10}(\lambda_{\min})}$

7:     $\lambda_{\min}^{\mathrm{out}} \leftarrow 1/\lambda_{\max}^{\mathrm{in}}$ and $\lambda_{\max}^{\mathrm{out}} \leftarrow 1/\lambda_{\min}^{\mathrm{in}}$

8:     solve Eq. (11) for $\boldsymbol{x}_i^{\mathrm{in}}$ and $\boldsymbol{x}_i^{\mathrm{out}}$ ($i = 1, \ldots, N_{\mathrm{int}}$)

9:     **for** X=in,out **do**

10:         set a so that $a\lambda_{\max}^{\mathrm{X}} = (a\lambda_{\min}^{\mathrm{X}})^{-1}$

11:         $\boldsymbol{x}_i' \leftarrow -z_i^{\mathrm{X}} \boldsymbol{x}_i^{\mathrm{X}}$

12:         **for** $l = 0, \ldots, N_{\mathrm{mom}} - 1$ **do**

13:             set $\hat{\boldsymbol{s}}_l$ from $\{\boldsymbol{x}_i^{\mathrm{X}}\}$ by Eq. (15b)

14:             set $\hat{\boldsymbol{s}}_l'$ from $\{\boldsymbol{x}_i'\}$ by Eq. (D1b)

15:         **end for**

16:         $\mathbf{S} \leftarrow [\hat{\boldsymbol{s}}_0, \ldots, \hat{\boldsymbol{s}}_{N_{\mathrm{mom}}-1}]$

17:         $\mathbf{S}' \leftarrow [\hat{\boldsymbol{s}}_0', \ldots, \hat{\boldsymbol{s}}_{N_{\mathrm{mom}}-1}']$

18:         $N_{\mathrm{eig}}' \leftarrow \mathrm{rank}\,\mathbf{S}$

19:         $\mathbf{U_S} \leftarrow$ column-space matrix of $\mathbf{S}$

20:         $\mathbf{U_S'} \leftarrow$ column-space matrix of $\mathbf{S}'$

21:         $\widetilde{\boldsymbol{\Pi}}_\mathbf{A} \leftarrow [\mathbf{U_S}]^\dagger \boldsymbol{\Pi}_\mathbf{A} \mathbf{U_S}$ and $\widetilde{\boldsymbol{\Pi}}_\mathbf{B} \leftarrow [\mathbf{U_S}]^\dagger \boldsymbol{\Pi}_\mathbf{B} \mathbf{U_S}$

22:         $\widetilde{\boldsymbol{\Pi}}_\mathbf{A}' \leftarrow [\mathbf{U_S'}]^\dagger \boldsymbol{\Pi}_\mathbf{A} \mathbf{U_S'}$ and $\widetilde{\boldsymbol{\Pi}}_\mathbf{B}' \leftarrow [\mathbf{U_S'}]^\dagger \boldsymbol{\Pi}_\mathbf{B} \mathbf{U_S'}$

23:         solve $\widetilde{\boldsymbol{\Pi}}_\mathbf{A} \check{\boldsymbol{\psi}} = \tilde{\lambda} \widetilde{\boldsymbol{\Pi}}_\mathbf{B} \check{\boldsymbol{\psi}}$ for $\{(\tilde{\lambda}_k, \check{\boldsymbol{\psi}}_k)\}$

24:         solve $\widetilde{\boldsymbol{\Pi}}_\mathbf{A}' \check{\boldsymbol{\psi}}' = \tilde{\lambda}' \widetilde{\boldsymbol{\Pi}}_\mathbf{B}' \check{\boldsymbol{\psi}}'$ for $\{(\tilde{\lambda}_k', \check{\boldsymbol{\psi}}_k')\}$

25:         $\{(\tilde{\lambda}_k, \check{\boldsymbol{\psi}}_k)\} \leftarrow \{(\tilde{\lambda}_k/a, \mathbf{U_S}\check{\boldsymbol{\psi}}_k)\}$

26:         $\{(\tilde{\lambda}_k', \check{\boldsymbol{\psi}}_k')\} \leftarrow \{(1/a\tilde{\lambda}_k', \mathbf{U_S'}\check{\boldsymbol{\psi}}_k')\}$

27:         **for** $k = 1, \ldots, N_{\mathrm{eig}}'$ **do**

28:             $\|\boldsymbol{r}\|_2 \leftarrow \|(\tilde{\lambda}_k \boldsymbol{\Pi}_\mathbf{B} - \boldsymbol{\Pi}_\mathbf{A})\check{\boldsymbol{\psi}}_k\|_2$

29:             $\|\boldsymbol{r}'\|_2 \leftarrow \|(\tilde{\lambda}_k' \boldsymbol{\Pi}_\mathbf{B} - \boldsymbol{\Pi}_\mathbf{A})\check{\boldsymbol{\psi}}_k'\|_2$

30:             **if** $\|\boldsymbol{r}'\|_2 < \|\boldsymbol{r}\|_2$ **then**

31:                 $(\tilde{\lambda}_k, \check{\boldsymbol{\psi}}_k) \leftarrow (\tilde{\lambda}_k', \check{\boldsymbol{\psi}}_k')$

32:             **end if**

33:         **end for**

34:         append $\{(\tilde{\lambda}_k, \check{\boldsymbol{\psi}}_k)\}$ to $\{(\lambda_i, \boldsymbol{\psi}_i)\}$

35:         $N_{\mathrm{eig}} \leftarrow N_{\mathrm{eig}} + N_{\mathrm{eig}}'$

36:     **end for**

37: **end for**

38: **output** N_{eig}, $\{(\lambda_i, \boldsymbol{\psi}_i) | i = 1, \ldots, N_{\mathrm{eig}}\}$

APPENDIX D: ALGORITHM FOR PRACTICAL CALCULATIONS

In practical calculations, the dividing-and-scaling technique and the interchange technique are used together. Taking into account the scaling factor a introduced in Sec. II D, Eq. (17) changes to

$$\hat{\boldsymbol{s}}'_l = \frac{1}{2\pi \mathrm{i}} \oint_{\hat{C}'} (\hat{z}')^l (\hat{z}' \hat{\boldsymbol{\Pi}}_{\mathbf{A}} - \boldsymbol{\Pi}_{\mathbf{B}})^{-1} \boldsymbol{u} d\hat{z}' \tag{D1a}$$

$$\approx \frac{1}{N_{\mathrm{int}}} \sum_{i}^{N_{\mathrm{int}}} (\hat{z}'_i)^l \underbrace{(\hat{z}'_i \hat{\boldsymbol{\Pi}}_{\mathbf{A}} - \boldsymbol{\Pi}_{\mathbf{B}})^{-1} \boldsymbol{u}}_{\hat{\boldsymbol{x}}'_i}, \tag{D1b}$$

where $\hat{z}'_i = z'_i / a$ and $\hat{\boldsymbol{\Pi}}_{\mathbf{A}} = a \boldsymbol{\Pi}_{\mathbf{A}}$. As is the discussion in Sec. II D, $\hat{\boldsymbol{x}}'_i$ is identical to $\boldsymbol{x}'_i$ of Eq. (18).

The algorithm for solving Eq. (8) in the present method incorporating the two computational techniques is shown in Algorithm 1. Note that the algorithm is described using the nonblock version of the SS-RR method, i.e., $N_{\mathrm{rhs}} = 1$. The algorithm using the block version of the SS-RR method can be straightforwardly derived from the algorithm.

[1] M. G. Reuter, J. Phys.: Condens. Matter **29**, 053001 (2017).

[2] J. A. Appelbaum and D. R. Hamann, Rev. Mod. Phys. **48**, 479 (1976); D. H. Lee and J. D. Joannopoulos, Phys. Rev. B **23**, 4997 (1981); J. Tersoff, Phys. Rev. Lett. **52**, 465 (1984); G. Wachutka, Phys. Rev. B **34**, 8512 (1986); P. H. Dederichs, Ph. Mavropoulos, O. Wunnicke, N. Papanikolaou, V. Bellini, R. Zeller, V. Drchal, and J. Kudrnovský, J. Magn. Magn. Mater. **240**, 108 (2002).

[3] J. Tersoff, Phys. Rev. B **30**, 4874 (1984); T. B. Boykin, *ibid.* **54**, 7670 (1996); R. T. Tung and L. Kronik, *ibid.* **94**, 075310 (2016); W. Mönch, Mater. Sci. Semicond. Proc. **28**, 2 (2014).

[4] S. Datta, *Electronic Transport in Mesoscopic Systems*, edited by H. Ahme, M. Pepper, and A. Broers, Cambridge Studies in Semiconductor Physics and Microelectronic Engineering Vol. 3 (Cambridge University Press, Cambridge, 1995).

[5] H. J. Choi and J. Ihm, Phys. Rev. B **59**, 2267 (1999); D. Wortmann, H. Ishida, and S. Blügel, *ibid.* **65**, 165103 (2002); P. S. Krstić, X.-G. Zhang, and W. H. Butler, *ibid.* **66**, 205319 (2002); I. Rungger and S. Sanvito, *ibid.* **78**, 035407 (2008); M. Luisier, A. Schenk, W. Fichtner, and G. Klimeck, *ibid.* **74**, 205323 (2006); S. Brück, M. Calderara, M. H. Bani-Hashemian, J. VandeVondele, and M. Luisier, J. Chem. Phys. **147**, 074116 (2017).

[6] K. S. Thygesen and A. Rubio, Phys. Rev. B **77**, 115333 (2008); M. Strange, C. Rostgaard, H. Häkkinen, and K. S. Thygesen, *ibid.* **83**, 115108 (2011).

[7] R. O. Jones, Proc. Phys. Soc. **89**, 443 (1966); Ph. Mavropoulos, N. Papanikolaou, and P. H. Dederichs, Phys. Rev. Lett. **85**, 1088 (2000); M. G. Reuter, T. Seideman, and M. A. Ratner, Phys. Rev. B **83**, 085412 (2011); J. Betancourt, S. Li, X. Dang, J. D. Burton, E. Y. Tsymbal, and J. P. Velev, J. Phys.: Condens. Matter **28**, 395501 (2016).

[8] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).

[9] J. C. Slater and G. F. Koster, Phys. Rev. **94**, 1498 (1954); C. M. Goringe, D. R. Bowler, and E. Hernández, Rep. Prog. Phys. **60**, 1447 (1997).

[10] J. R. Chelikowsky, N. Troullier, K. Wu, and Y. Saad, Phys. Rev. B **50**, 11355 (1994); J. R. Chelikowsky, N. Troullier, and Y. Saad, Phys. Rev. Lett. **72**, 1240 (1994).

[11] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965); W. Kohn, Rev. Mod. Phys. **71**, 1253 (1999).

[12] Y. Fujimoto and K. Hirose, Phys. Rev. B **67**, 195315 (2003).

[13] T. Ando, Phys. Rev. B **44**, 8017 (1991).

[14] P. A. Khomyakov and G. Brocks, Phys. Rev. B **70**, 195402 (2004); P. A. Khomyakov, G. Brocks, V. Karpan, M. Zwierzycki, and P. J. Kelly, *ibid.* **72**, 035450 (2005).

[15] K. Hirose, T. Ono, Y. Fujimoto, and S. Tsukamoto, *First-Principles Calculations in Real-Space Formalism* (Imperial College Press, London, 2005).

[16] S. Tsukamoto, K. Hirose, and S. Blügel, Phys. Rev. E **90**, 013306 (2014).

[17] S. Iwase, Y. Futamura, A. Imakura, T. Sakurai, and T. Ono, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '17* (ACM, New York, NY, USA, 2017), pp. 40:1–40:12.

[18] H. H. B. Sørensen, P. C. Hansen, D. E. Petersen, S. Skelboe, and K. Stokbro, Phys. Rev. B **77**, 155301 (2008).

[19] S. Tsukamoto, T. Ono, and S. Blügel, Phys. Rev. B **97**, 115450 (2018).

[20] T. Sakurai and H. Sugiura, J. Comput. Appl. Math. **159**, 119 (2003); T. Ikegami, T. Sakurai, and U. Nagashima, *ibid.* **233**, 1927 (2010); J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura, JSIAM Lett. **1**, 52 (2009).

[21] S. Iwase, Y. Futamura, A. Imakura, T. Sakurai, S. Tsukamoto, and T. Ono, Phys. Rev. B **97**, 195449 (2018).

[22] Y. Futamura, T. Sakurai, S. Furuya, and J.-I. Iwata, in *High Performance Computing for Computational Science - VECPAR 2012*, edited by M. Daydé, O. Marques, and K. Nakajima (Springer, Berlin, Heidelberg, 2013), pp. 226–235.

[23] S. E. Laux, Phys. Rev. B **86**, 075103 (2012); E. Polizzi, *ibid.* **79**, 115112 (2009).

[24] T. Sakurai and H. Tadano, Hokkaido Math. J. **36**, 745 (2007); T. Ikegami and T. Sakurai, Taiwan. J. Math. **14**, 825 (2010); A. Imakura and T. Sakurai, Numer. Alg. **75**, 413 (2017); Y. Nagai, Y. Shinohara, Y. Futamura, Y. Ota, and T. Sakurai, J. Phys. Soc. Jpn. **82**, 094701 (2013).

[25] A. Imakura, L. Du, and T. Sakurai, Numer. Alg. **71**, 103 (2016).

[26] J. van den Eshof and G. L. G. Sleijpen, Appl. Numer. Math. **49**, 17 (2004).

[27] S. Iwase, T. Hoshi, and T. Ono, Phys. Rev. E **91**, 063305 (2015).

[28] R. Fletcher, in *Numerical Analysis*, edited by G. A. Watson, Lecture Notes in Mathematics Vol. 506 (Springer, Berlin, Heidelberg, 1976), pp. 73–89.

[29] A. Frommer, Computing **70**, 87 (2003); R. Takayama, T. Hoshi, T. Sogabe, S.-L. Zhang, and T. Fujiwara, Phys. Rev. B **73**, 165108 (2006).

[30] H. Ebert, D. Ködderitzsch, and J. Minár, Rep. Prog. Phys. **74**, 096501 (2011); N. Papanikolaou, R. Zeller, and P. H. Dederichs, J. Phys.: Condens. Matter **14**, 2799 (2002).

[31] The effective potential and parameters necessary for constructing Π_A and Π_B are predetermined by the electronic structure calculations using RSPACE [15,32], which is also based on the real-space finite-difference formalism. The exchange-correlation interaction is treated using the local density approximation [38] within the framework of the density functional theory, and the interaction between valence electrons and nuclei is treated using the norm-conserving pseudopotentials [39].

[32] T. Ono, M. Heide, N. Atodiresei, P. Baumeister, S. Tsukamoto, and S. Blügel, Phys. Rev. B **82**, 205115 (2010); T. Ono and K. Hirose, *ibid.* **72**, 085115 (2005); Phys. Rev. Lett. **82**, 5016 (1999).

[33] Y. Matsuda, J. Tahir-Kheli, and W. A. Goddard, III, J. Phys. Chem. Lett. **1**, 2946 (2010).

[34] V. Heine, Proc. Phys. Soc. **81**, 300 (1963).

[35] Y.-C. Chang, Phys. Rev. B **25**, 605 (1982).

[36] A. Rubio, J. L. Corkill, and M. L. Cohen, Phys. Rev. B **49**, 5081 (1994); X. Blase, A. Rubio, S. G. Louie, and M. L. Cohen, Europhys. Lett. **28**, 335 (1994); N. G. Chopra, R. J. Luyken, K. Cherrey, V. H. Crespi, M. L. Cohen, S. G. Louie, and A. Zettl, Science **269**, 966 (1995).

[37] W.-J. Beyn, Lin. Alg. Appl. **436**, 3839 (2012).

[38] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981); D. M. Ceperley and B. J. Alder, Phys. Rev. Lett. **45**, 566 (1980).

[39] N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991); K. Kobayashi, Comput. Mater. Sci. **14**, 72 (1999).