

Solution of the k -th eigenvalue problem in large-scale electronic structure calculations

Dongjin Lee^{a,*}, Takeo Hoshi^b, Tomohiro Sogabe^a, Yuto Miyatake^a, Shao-Liang Zhang^a

^a Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

^b Department of Applied Mathematics and Physics, Tottori University, 4-101 Koyama-Minami, Tottori 680-8550, Japan

ARTICLE INFO

Article history:

Received 12 October 2017

Received in revised form 5 April 2018

Accepted 1 June 2018

Available online 6 June 2018

Keywords:

Generalized eigenvalue problem

Electronic structure calculations

Spectral bisection

Lanczos method

Sparse direct linear solver

ABSTRACT

We consider computing the k -th eigenvalue and its corresponding eigenvector of a generalized Hermitian eigenvalue problem of $n \times n$ large sparse matrices. In electronic structure calculations, several properties of materials, such as those of optoelectronic device materials, are governed by the eigenpair with a material-specific index k . We present a three-stage algorithm for computing the k -th eigenpair with validation of its index. In the first stage of the algorithm, we propose an efficient way of finding an interval containing the k -th eigenvalue ($1 \ll k \ll n$) with a non-standard application of the Lanczos method. In the second stage, spectral bisection for large-scale problems is realized using a sparse direct linear solver to narrow down the interval of the k -th eigenvalue. In the third stage, we switch to a modified shift-and-invert Lanczos method to reduce bisection iterations and compute the k -th eigenpair with validation. Numerical results with problem sizes up to 1.5 million are reported, and the results demonstrate the accuracy and efficiency of the three-stage algorithm.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

In this paper, we consider the solution of the k -th eigenvalue problem with $n \times n$ large sparse Hermitian A, B and positive definite B :

$$A\mathbf{x}_k = \lambda_k B\mathbf{x}_k, \quad \mathbf{x}_k \neq \mathbf{0}, \quad (1)$$

where $\lambda_1 \leq \dots \leq \lambda_k \leq \dots \leq \lambda_n$. Here, we assume that the problem-specific target index k satisfies $1 \ll k \ll n$ such that λ_k is not at either end of $[\lambda_1, \lambda_n]$.

The k -th eigenvalue problem (1) differs from other problems to compute part of the spectrum and the corresponding eigenvectors. Some eigenvalues at the ends of $[\lambda_1, \lambda_n]$ and their corresponding eigenvectors can be computed by the Lanczos [1] and LOBPCG [2] methods, and some eigenvalues near a given target point and their eigenvectors are computed by the shift-and-invert Lanczos (SI Lanczos) [3] and Jacobi–Davidson [4] methods. In addition, eigenvalues in a given target interval and their eigenvectors are computed by the Sakurai–Sugiura method [5], FEAST method [6], and filtering methods [7]. However, none of these methods aim at computing the eigenpair of a given target index k with $1 \ll k \ll n$.

* Corresponding author.

E-mail addresses: dongjin-i@na.nuap.nagoya-u.ac.jp (D. Lee), hoshi@damp.tottori-u.ac.jp (T. Hoshi), sogabe@na.nuap.nagoya-u.ac.jp (T. Sogabe), miyatake@na.nuap.nagoya-u.ac.jp (Y. Miyatake), zhang@na.nuap.nagoya-u.ac.jp (S.-L. Zhang).

<https://doi.org/10.1016/j.jcp.2018.06.002>

0021-9991/© 2018 Elsevier Inc. All rights reserved.

The k -th eigenvalue problem (1) arises from large-scale electronic structure calculations [8], where eigenvalues correspond to the energy of an electron and eigenvectors represent an electronic wave function. Here, λ_k and \mathbf{x}_k are referred to as the highest occupied (HO) energy and state, respectively. The target index k is a material-specific value and is approximately 10–50% of the matrix size [9], thereby satisfying $1 \ll k \ll n$. The index of the eigenpair must be validated because several of the physical properties of materials, such as those of optoelectronic device materials, are governed by the eigenpair with the material-specific target index k . Detailed explanation of physical origin and background of the k -th eigenvalue problem can be found in Appendices A and B.

Typical electronic structure calculations require the computation of many eigenpairs. Eigenpairs of practical interest are generally $(\lambda_1, \mathbf{x}_1)$ through $(\lambda_k, \mathbf{x}_k)$. A standard approach is to utilize a dense eigensolver in a massively parallel environment [9–11]. Recently, a one-million-dimensional generalized eigenvalue problem was solved by a dense eigensolver [12] using the full K computer. The elapsed time was 5516 seconds to compute all eigenpairs [13], indicating the practical size limit of eigenpair computation by a dense eigensolver. Therefore, a strong need for methodologies that can be applied to larger materials and matrices has become apparent.

A promising approach to large-scale electronic structure calculations is to construct purpose-specific methods with which one can bypass computation of many eigenpairs to obtain several physical quantities of practical interest. Several methods have been proposed to calculate the total energy of materials and the force on nuclei in order to realize quantum molecular dynamics simulations without computation of each individual eigenpair [14]. Such methods are referred to as linear-scaling methods.

Our previous paper [15] presented the k -th eigenvalue problem (1) as a purpose-specific methodology for large-scale calculations that can be considered complementary to linear-scaling methods. In our previous paper, as a preliminary study of the problem, spectral bisection and its variants were applied to computing the k -th eigenvalue and validating its index. As explained in Section 2.1, the main idea behind bisection is to locate λ_k based on the number of eigenvalues that are less than a given real number σ , denoted $\nu_\sigma(A, B)$.

Using our previous paper as a foundation, this paper presents a three-stage algorithm for solving the k -th eigenvalue problem with the following features.

1. Efficient initial interval for bisection

Setting an initial interval containing λ_k is necessary to begin bisection. A standard approach is to utilize some Gershgorin-type theorem so that the interval includes the entire spectrum and thus contains λ_k . In the first stage of the algorithm, we propose an efficient way of finding a narrow initial interval. The proposed approach iteratively generates a sequence of disjoint intervals until an interval validated as containing λ_k is obtained. For our problem with the target index $1 \ll k \ll n$, the numerical results show that information conventionally considered useless in the Lanczos method, i.e., Ritz values of the first few steps of the method, is of paramount importance to generate a narrow initial interval.

2. Computation of the k -th eigenvector with validation

Mistakenly computing another eigenvector, e.g., \mathbf{x}_{k-1} or \mathbf{x}_{k+1} , rather than \mathbf{x}_k leads to a completely unreliable result because eigenvectors are B -orthogonal to each other. In the third stage, as will be shown in Proposition 3, the index is validated by utilizing an eigenvalue error bound that can be evaluated from the residual vector at negligible cost.

3. Application to large-scale problems

Each bisection iteration requires computation of $\nu_\sigma(A, B)$, which is based on LDL^H factorization of a shifted matrix $A - \sigma B$ and thus requires $O(n^2)$ memory in general. In the second stage, $\nu_\sigma(A, B)$ is computed by utilizing a sparse direct linear solver to save memory to realize bisection for large-scale problems. Once a fill-reducing ordering and symbolic factorization are obtained for some $A - \sigma B$, they can be recycled for other shifted matrices because they depend on only the sparsity structure of a matrix.

The remainder of this paper is organized as follows. Section 2 explains the preliminaries of this paper. In Section 3, after explaining our approach to set an initial interval and compute the k -th eigenvector, we present the three-stage algorithm for the k -th eigenvalue problem. Numerical results of several real research problems and a comparison of the three-stage algorithm and dense eigensolvers are reported in Section 4. Concluding remarks are given in Section 5.

Throughout this paper, A^T and A^H denote the transpose and conjugate transpose of matrix A , respectively. I denotes the identity matrix. For Hermitian positive definite B , $\|\mathbf{x}\|_B$ denotes the B -norm of vector \mathbf{x} , defined as $\|\mathbf{x}\|_B = \sqrt{\mathbf{x}^H B \mathbf{x}}$.

2. Preliminaries

Section 2.1 explains spectral bisection for computing the k -th eigenvalue. Sections 2.2 and 2.3 provide the preliminaries of the Lanczos and SI Lanczos methods.

2.1. Spectral bisection for the k -th eigenvalue

In spectral bisection, $\nu_\sigma(A, B)$ is computed to determine whether λ_k is located to the left or right of σ . Specifically, $k \leq \nu_\sigma(A, B)$ implies that $\lambda_k < \sigma$, while $k > \nu_\sigma(A, B)$ implies that $\lambda_k \geq \sigma$. This idea dates back to the work of Givens [16].

Algorithm 1 shows the spectral bisection for the k -th eigenvalue, which was employed but not clearly presented in our previous paper [15]. When an interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$ containing λ_k is set, the midpoint σ of the interval is calculated in line 2, and then $\nu_\sigma(A, B)$ is computed by utilizing LDL^H factorization of the shifted matrix $A - \sigma B$ in lines 3–4. Here, $A - \sigma B$ is indefinite; thus, permutation P is necessary for numerically stable factorization [17], and L and D are a unit lower triangular matrix and a block diagonal matrix of block size one or two, respectively. Due to Sylvester's law of inertia, $\nu_0(D, I)$ equals $\nu_\sigma(A, B)$. Depending on $\nu_\sigma(A, B)$, either the left or right half-interval is selected as the next interval in line 5. $\lceil \log_2[(\sigma_{\text{upper}} - \sigma_{\text{lower}})/\tau] \rceil$ iterations are required until the interval becomes narrower than a given tolerance τ . Thus, a narrower initial interval will result in fewer required iterations to locate λ_k .

Algorithm 1: Spectral bisection for the k -th eigenvalue [15].

Input : matrices A, B of generalized eigenvalue problem (1), target index k ,
initial interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$ containing the k -th eigenvalue, tolerance τ .
Output: approximate eigenvalue $\hat{\lambda}_k := (\sigma_{\text{lower}} + \sigma_{\text{upper}})/2$, where $|\hat{\lambda}_k - \lambda_k| < \tau/2$.

```

1 repeat until  $\sigma_{\text{upper}} - \sigma_{\text{lower}} < \tau$ 
2    $\sigma := (\sigma_{\text{lower}} + \sigma_{\text{upper}})/2$ ,
3    $LDL^H \leftarrow P(A - \sigma B)P^T$ ,
4    $\nu := \nu_0(D, I)$ ,
5   if  $k \leq \nu$  then  $\sigma_{\text{upper}} := \sigma$  else  $\sigma_{\text{lower}} := \sigma$ .
```

▷ P : permutation for numerical stability
▷ $\nu_0(D, I)$: number of negative eigenvalues of block diagonal D

Rather than bisection, it is possible to apply other root-finding algorithms to line 2 of Algorithm 1 to achieve faster convergence. By considering eigenvalues as the roots of a characteristic polynomial $\det(A - \sigma B)$, several root-finding algorithms are applied to select point σ to compute $\nu_\sigma(A, B)$ [18]. Generally, such variants can be applied only after the interval is narrowed down sufficiently to contain only one eigenvalue [19, Chapter 3.5]. Our previous paper [15] took another perspective and considered $\nu_\sigma(A, B)$ as a function of σ , which is non-decreasing, integer-valued, and discontinuous at $\sigma = \lambda_i$ for $1 \leq i \leq n$. This perspective enabled the application of a certain type of root-finding algorithms to select σ before λ_k was isolated from the other eigenvalues by the interval. To date, a thorough comparison of bisection and its variants has not been made for the k -th eigenvalue problem, and, in this paper, we utilize bisection for the sake of stable performance.

2.2. Lanczos method

The Lanczos method [1] is a projection method in which approximate solutions are constructed within a Krylov subspace and are determined to be optimal in the sense of the Galerkin condition. The subspace and its orthonormal basis are generated by the Lanczos process, which can be expressed in the matrix form:

$$AV_j = BV_jT_j + B\mathbf{v}_{j+1}\beta_j\mathbf{e}_j^T. \quad (2)$$

Here, V_j is an $n \times j$ matrix whose columns span a Krylov subspace and are B -orthonormal. T_j is real symmetric tridiagonal and has non-zero off-diagonal elements (irreducible). \mathbf{v}_{j+1} is B -orthogonal to the columns of V_j and is normalized with respect to the B -norm by scale factor β_j . \mathbf{e}_j is the last column of the identity matrix of size j . In this paper, we refer to (2) as j -step Lanczos decomposition.

From the Galerkin condition, the standard eigenvalue problem of T_j is derived:

$$T_j \mathbf{y}_i^{(j)} = \theta_i^{(j)} \mathbf{y}_i^{(j)}, \quad \mathbf{y}_i^{(j)} \neq \mathbf{0}. \quad (3)$$

Since T_j is irreducible, eigenvalues $\theta_i^{(j)}$ are distinct from each other [20, Chapter 1.3] and can be indexed in increasing order, i.e., $\theta_1^{(j)} < \theta_2^{(j)} < \dots < \theta_j^{(j)}$. The Lanczos method can be considered a Rayleigh–Ritz procedure, and eigenvalues $\theta_i^{(j)}$ are referred to as Ritz values.

2.3. Shift-and-invert Lanczos method

A small number of eigenvalues near a target point (or in a target interval) and their associated eigenvectors can be computed by the SI Lanczos method [3]. The SI Lanczos method applied to the original problem (1) can be considered as applying the original Lanczos method (Section 2.2) to the SI problem:

$$(A - \sigma B)^{-1} \tilde{\mathbf{x}} = \tilde{\lambda} B^{-1} \tilde{\mathbf{x}}, \quad \tilde{\mathbf{x}} \neq \mathbf{0}. \quad (4)$$

Here, it is assumed that shift σ does not coincide with an eigenvalue of (1), i.e., $\sigma \neq \lambda_i$. The eigenvalues of the original and SI problems have the relationship $\tilde{\lambda} = (\lambda - \sigma)^{-1}$, while eigenvectors satisfy both $\tilde{\mathbf{x}} = B\mathbf{x}$ and $\tilde{\mathbf{x}} = (A - \sigma B)\mathbf{x}$ because $B\mathbf{x}$ and $(A - \sigma B)\mathbf{x}$ are collinear. Based on these relationships, approximate eigenpairs for the SI problem (4) are transformed to those for the original problem (1).

The matrix form of j -step SI Lanczos decomposition is given as follows:

$$(A - \sigma B)^{-1} \tilde{V}_j = B^{-1} \tilde{V}_j \tilde{T}_j + B^{-1} \tilde{\mathbf{v}}_{j+1} \tilde{\beta}_j \mathbf{e}_j^T. \quad (5)$$

Here, \tilde{V}_j is an $n \times j$ matrix whose columns span a Krylov subspace and are B^{-1} -orthonormal, and \tilde{T}_j is real symmetric tridiagonal and irreducible. $\tilde{\mathbf{v}}_{j+1}$ is B^{-1} -orthogonal to the columns of \tilde{V}_j and is normalized with respect to the B^{-1} -norm by scale factor $\tilde{\beta}_j$.

From the Galerkin condition, the standard eigenvalue problem of \tilde{T}_j is derived:

$$\tilde{T}_j \tilde{\mathbf{y}}_i^{(j)} = \tilde{\theta}_i^{(j)} \tilde{\mathbf{y}}_i^{(j)}, \quad \tilde{\mathbf{y}}_i^{(j)} \neq \mathbf{0}. \quad (6)$$

Here, eigenvalues $\tilde{\theta}_i^{(j)}$ are indexed in increasing order, and, in the remainder of this paper, eigenvectors $\tilde{\mathbf{y}}_i^{(j)}$ are assumed to be normalized with respect to the 2-norm. Using the i -th eigenpair $(\tilde{\theta}_i^{(j)}, \tilde{\mathbf{y}}_i^{(j)})$ of \tilde{T}_j , the approximate eigenvalues for (1) are expressed as follows:

$$\lambda_i^{(j)} = \sigma + 1/\tilde{\theta}_i^{(j)}. \quad (7)$$

Approximate eigenvectors can be expressed in two different ways.

1. Using the first relationship $\tilde{\mathbf{x}} = B\mathbf{x}$, approximate eigenvectors are given as:

$$\mathbf{x}_{i,1}^{(j)} = B^{-1} \tilde{V}_j \tilde{\mathbf{y}}_i^{(j)}. \quad (8)$$

2. From the second relationship $\tilde{\mathbf{x}} = (A - \sigma B)\mathbf{x}$, we have:

$$\mathbf{x}_{i,2}^{(j)} = (A - \sigma B)^{-1} \tilde{V}_j \tilde{\mathbf{y}}_i^{(j)}. \quad (9)$$

In this paper, we utilize $\mathbf{x}_{i,2}^{(j)}$ in (9) as an approximate eigenvector, although it is common to use $\mathbf{x}_{i,1}^{(j)}$ in (8). This is because, as will be shown in Proposition 3, $\mathbf{x}_{i,2}^{(j)}$ allows economical evaluation of an eigenvalue error bound that can be utilized to validate the index of approximate eigenpairs. Other perspectives on approximate eigenvectors and their further treatment can be found in the literature [3] and [21, Chapter 7.6.8], in which $\mathbf{x}_{i,2}^{(j)}$ is considered a modification of $\mathbf{x}_{i,1}^{(j)}$.

The remainder of this subsection provides theoretical results that support the utilization of $\mathbf{x}_{i,2}^{(j)}$ as an approximate eigenvector. In Proposition 1, we show that $\mathbf{x}_{i,2}^{(j)}$ converges to the same eigenvector of (1) at the same iteration of the SI Lanczos method as $\mathbf{x}_{i,1}^{(j)}$. Then, in Proposition 2, we show that $\mathbf{x}_{i,2}^{(j)}$ with $1 \leq i \leq j$ become B -orthogonal to each other as the SI Lanczos method proceeds.

Proposition 1. Approximate eigenvectors $\mathbf{x}_{i,1}^{(j)}$ in (8) and $\mathbf{x}_{i,2}^{(j)}$ in (9) are collinear if and only if $\tilde{\mathbf{v}}_{j+1} = \mathbf{0}$.

Proof. We first prove the sufficiency. By post-multiplying (5) by $\tilde{\mathbf{y}}_i^{(j)}$ and from the sufficient condition $\tilde{\mathbf{v}}_{j+1} = \mathbf{0}$, we have:

$$\mathbf{x}_{i,2}^{(j)} = (A - \sigma B)^{-1} \tilde{V}_j \tilde{\mathbf{y}}_i^{(j)} = B^{-1} \tilde{V}_j \tilde{T}_j \tilde{\mathbf{y}}_i^{(j)} = \tilde{\theta}_i^{(j)} \mathbf{x}_{i,1}^{(j)}.$$

The third equality follows from (6). This proves the sufficiency. Now, we prove the necessity. By post-multiplying (5) by $\tilde{\mathbf{y}}_i^{(j)}$, we have:

$$\mathbf{x}_{i,2}^{(j)} = \tilde{\theta}_i^{(j)} \mathbf{x}_{i,1}^{(j)} + B^{-1} \tilde{\mathbf{v}}_{j+1} (\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)}).$$

Here, $\mathbf{x}_{i,1}^{(j)}$ and $\mathbf{x}_{i,2}^{(j)}$ are collinear from the necessary condition; thus, the last term $B^{-1} \tilde{\mathbf{v}}_{j+1} (\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)})$ must be collinear with $\mathbf{x}_{i,1}^{(j)}$ and $\mathbf{x}_{i,2}^{(j)}$. In addition, the last term is B -orthogonal to $\mathbf{x}_{i,1}^{(j)}$ because $\tilde{\mathbf{v}}_{j+1}$ is B^{-1} -orthogonal to the columns of \tilde{V}_j :

$$(\mathbf{x}_{i,1}^{(j)})^H B \left[B^{-1} \tilde{\mathbf{v}}_{j+1} (\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)}) \right] = (\tilde{\mathbf{y}}_i^{(j)})^H \tilde{V}_j^H B^{-1} \tilde{\mathbf{v}}_{j+1} (\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)}) = 0.$$

Due to this B -orthogonality and the positive-definiteness of B , the last term can never be collinear with $\mathbf{x}_{i,1}^{(j)}$ unless it is the zero vector. Therefore, $\tilde{\mathbf{v}}_{j+1} = \mathbf{0}$ (thus, $\tilde{\beta}_j = 0$) or $\mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)} = 0$. However, $\mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)}$ is non-zero because it is the last element of an eigenvector of an irreducible tridiagonal matrix [19, Theorem 7.9.3]. This proves the necessity. \square

Note that $\mathbf{x}_{i,2}^{(j)}$ with $1 \leq i \leq j$ do not have exact B -orthogonality. Their B -orthogonality can be measured by (10) in Proposition 2, which is the cosine similarity of two approximate eigenvectors in the B -inner product.

Proposition 2. The following holds for $1 \leq l < m \leq j$:

$$\frac{|(\mathbf{x}_{l,2}^{(j)})^H B \mathbf{x}_{m,2}^{(j)}|}{\|\mathbf{x}_{l,2}^{(j)}\|_B \cdot \|\mathbf{x}_{m,2}^{(j)}\|_B} = \frac{|\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_l^{(j)} / \tilde{\theta}_l^{(j)}|}{\sqrt{1 + |\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_l^{(j)} / \tilde{\theta}_l^{(j)}|^2}} \cdot \frac{|\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_m^{(j)} / \tilde{\theta}_m^{(j)}|}{\sqrt{1 + |\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_m^{(j)} / \tilde{\theta}_m^{(j)}|^2}}. \quad (10)$$

Proof. For $1 \leq l < m \leq j$,

$$\begin{aligned} (\mathbf{x}_{l,2}^{(j)})^H B \mathbf{x}_{m,2}^{(j)} &= \left[(A - \sigma B)^{-1} \tilde{V}_j \tilde{\mathbf{y}}_l^{(j)} \right]^H B \left[(A - \sigma B)^{-1} \tilde{V}_j \tilde{\mathbf{y}}_m^{(j)} \right] \\ &= \left(\tilde{V}_j \tilde{T}_j \tilde{\mathbf{y}}_l^{(j)} + \tilde{\mathbf{v}}_{j+1} \tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_l^{(j)} \right)^H B^{-1} B B^{-1} \left(\tilde{V}_j \tilde{T}_j \tilde{\mathbf{y}}_m^{(j)} + \tilde{\mathbf{v}}_{j+1} \tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_m^{(j)} \right) \\ &= \left(\tilde{V}_j \tilde{T}_j \tilde{\mathbf{y}}_l^{(j)} \right)^H B^{-1} \left(\tilde{V}_j \tilde{T}_j \tilde{\mathbf{y}}_m^{(j)} \right) + \left(\tilde{\mathbf{v}}_{j+1} \tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_l^{(j)} \right)^H B^{-1} \left(\tilde{\mathbf{v}}_{j+1} \tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_m^{(j)} \right) \\ &= \tilde{\theta}_l^{(j)} \tilde{\theta}_m^{(j)} \left[\delta_{lm} + (\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_l^{(j)} / \tilde{\theta}_l^{(j)}) (\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_m^{(j)} / \tilde{\theta}_m^{(j)}) \right]. \end{aligned} \quad (11)$$

Here, δ_{lm} denotes the Kronecker delta. The second equality follows from (5), and the third and fourth are due to the B^{-1} -orthonormality of the columns of \tilde{V}_j and $\tilde{\mathbf{v}}_{j+1}$. (10) is an immediate result of (11) with $l \neq m$. \square

Scalars $|\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)} / \tilde{\theta}_i^{(j)}|$ in (10) are simply the B^{-1} -norm of the residual vectors:

$$\mathbf{r}_{i,2}^{(j)} \equiv (A - \lambda_i^{(j)} B) \mathbf{x}_{i,2}^{(j)} = \left[A - (\sigma + 1 / \tilde{\theta}_i^{(j)}) B \right] (A - \sigma B)^{-1} \tilde{V}_j \tilde{\mathbf{y}}_i^{(j)} = -\tilde{\mathbf{v}}_{j+1} (\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)} / \tilde{\theta}_i^{(j)}). \quad (12)$$

Since the residual norm $|\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)} / \tilde{\theta}_i^{(j)}| \ll 1$ as the SI Lanczos method proceeds, (10) converges to zero and approximate eigenvectors $\mathbf{x}_{i,2}^{(j)}$ become B -orthogonal to each other. Therefore, we can say that $\mathbf{x}_{i,2}^{(j)}$ have B -orthogonality in a practical sense.

3. A three-stage algorithm for the k -th eigenvalue problem

3.1. Efficient initial interval for bisection

To utilize spectral bisection, it is necessary to set an initial interval that contains the k -th eigenvalue. A common approach is to set an interval that includes the entire spectrum, which necessarily contains λ_k . For standard eigenvalue problems, one of the most economical ways to set such an interval is to use the Gershgorin circle theorem. Several Gershgorin-type theorems have been proposed for generalized eigenvalue problems [22–24]. These theorems provide an inclusion set of the spectrum that is guaranteed to be bounded only when at least one of A and B is diagonally dominant for each row [24]. Unfortunately, such diagonal dominance of the matrices is not always assumed to be the case for problems in electronic structure calculations. The remainder of this subsection presents a systematic way to set an initial interval based on an application of the Lanczos method.

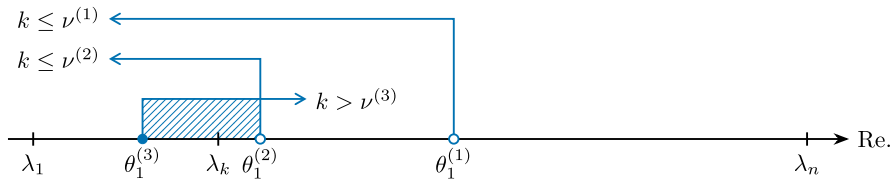
Ritz values by the Lanczos method become more accurate by expanding the subspace, exhibiting the interlacing property [20, Chapter 1.3]. By denoting the i -th Ritz value at the j -th iteration of the method as $\theta_i^{(j)}$ (Section 2.2), the interlacing property can be formally expressed as follows: $\theta_i^{(j+1)} < \theta_i^{(j)} < \theta_{i+1}^{(j+1)}$ for $i \leq j < n$. The monotonic convergence follows from this property, which states that the i -th smallest (resp. largest) Ritz value decreases (resp. increases) monotonically and converges to the i -th smallest (resp. largest) eigenvalue. We utilize this monotonicity of Ritz values to set an initial interval.

Algorithm 2 shows how we set an interval containing λ_k by utilizing Ritz values, and this process is illustrated in Fig. 1. Here, we utilize either the smallest $\theta_1^{(j)}$ or largest Ritz values $\theta_j^{(j)}$ with $j \geq 1$. We begin by computing $\theta_1^{(1)}$, which is equal to the generalized Rayleigh quotient $\mathbf{v}_1^H A \mathbf{v}_1 / \mathbf{v}_1^H B \mathbf{v}_1$ of a random starting vector \mathbf{v}_1 . The quotient is expected to be the average of the eigenvalues of (1). It is advantageous to begin from approximately the middle of the eigenvalue distribution when the target index satisfies $1 \ll k \ll n$. We then compute $\nu^{(1)} = \nu_{\theta_1^{(1)}}(A, B)$. If $k \leq \nu^{(1)}$, it follows that $\lambda_k < \theta_1^{(1)} < \theta_j^{(j)}$ for $j > 1$. Therefore, in the subsequent iterations, the smallest Ritz values $\theta_1^{(j)}$ are utilized as the points for setting an interval containing λ_k . On the other hand, if $k > \nu^{(1)}$, the largest Ritz values are selected as the endpoints of an interval. Accordingly, a sequence of disjoint intervals, i.e., either $[\theta_1^{(j)}, \theta_1^{(j-1)})$ or $[\theta_{j-1}^{(j-1)}, \theta_j^{(j)})$ with $j > 1$, is generated until an interval validated as containing λ_k is obtained.

An advantage of utilizing Ritz values is that an initial interval is necessarily included in and can be much narrower than $[\lambda_1, \lambda_n]$, which leads to a reduction of the number of bisection iterations. However, we must consider the cost of setting the interval, i.e., j iterations of the Lanczos method and j LDL^H factorizations. The interlacing property provides a theoretical upper bound of the iteration count required to set an interval. Here, since $\theta_1^{(n-k+1)} < \lambda_k < \theta_k^{(k)}$, at most $n - k + 1$ iterations

Algorithm 2: Setting an interval containing the k -th eigenvalue.**Input** : matrices A, B of generalized eigenvalue problem (1), target index k .**Output** : interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}}]$ containing the k -th eigenvalue, $\nu_{\text{lower}} = \nu_{\sigma_{\text{lower}}}(A, B), \nu_{\text{upper}} = \nu_{\sigma_{\text{upper}}}(A, B).$ 1 set a random starting vector $\mathbf{v}_1, \mathbf{v}_1 := \mathbf{v}_1 / \|\mathbf{v}_1\|_B, \nu^{(0)} := 0,$ 2 **for** $j = 1, 2, \dots$ **do**3 compute j -step Lanczos decomposition (2),

4 solve standard eigenvalue problem (3),

5 **if** $k \leq \nu^{(j-1)}$ **then** $\sigma^{(j)} := \theta_1^{(j)}$ **else** $\sigma^{(j)} := \theta_j^{(j)},$ 6 $LDL^H \leftarrow P(A - \sigma^{(j)}B)P^T,$ $\triangleright P$: permutation for numerical stability7 $\nu^{(j)} := \nu_0(D, I),$ $\triangleright \nu_0(D, I)$: number of negative eigenvalues of block diagonal D 8 **if** $j \neq 1$ and $(\nu^{(j)} < k \leq \nu^{(j-1)} \text{ or } \nu^{(j-1)} < k \leq \nu^{(j)})$ **then break,**9 $\sigma_{\text{lower}} := \min\{\sigma^{(j-1)}, \sigma^{(j)}\}, \sigma_{\text{upper}} := \max\{\sigma^{(j-1)}, \sigma^{(j)}\},$ 10 $\nu_{\text{lower}} := \min\{\nu^{(j-1)}, \nu^{(j)}\}, \nu_{\text{upper}} := \max\{\nu^{(j-1)}, \nu^{(j)}\}.$ **Fig. 1.** Illustration of Algorithm 2. $[\theta_1^{(3)}, \theta_1^{(1)}]$ does not contain λ_k but $[\theta_1^{(3)}, \theta_1^{(2)}]$ does (hatched).

are required for the $k \leq \nu^{(1)}$ case, and k iterations are required for the $k > \nu^{(1)}$ case. In practice, j can be very small because eigenvalues at the ends of $[\lambda_1, \lambda_n]$ are rapidly approximated by the Lanczos method and the target index k is approximately 10–50% of the matrix size n , thereby satisfying $1 \ll k \ll n$. In the numerical results presented in Section 4.3.1, the iteration count j is two for all experiments.

3.2. Computation of the k -th eigenpair with validation

Once an initial interval is set for bisection (Section 3.1), it is narrowed down until it contains the k -th eigenvalue and some nearby eigenvalues. We then compute the k -th eigenpair along with the other eigenpairs of the interval while validating their index. The eigenpairs of the interval can be computed by a variety of methods (Section 1). We utilize a modified SI Lanczos method (Section 2.3), where approximate eigenvectors $\mathbf{x}_{i,2}^{(j)}$ are computed based on (9) and have B -orthogonality, as shown in Proposition 2.

In Proposition 3, we explain that an eigenvalue error bound for validating the index of the eigenpairs can be evaluated from the residual vector of the SI Lanczos method at negligible cost.

Proposition 3. With the same notation as (5)–(7) in Section 2.3, there is an eigenvalue $\lambda_{l_i^{(j)}}$ of problem (1) such that:

$$\lambda_{l_i^{(j)}} \in \Gamma_i^{(j)} \equiv [\lambda_i^{(j)} - \eta_i^{(j)}, \lambda_i^{(j)} + \eta_i^{(j)}], \quad \eta_i^{(j)} \equiv \frac{1}{|\tilde{\theta}_i^{(j)}|} \cdot \frac{|\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)} / \tilde{\theta}_i^{(j)}|}{\sqrt{1 + |\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)} / \tilde{\theta}_i^{(j)}|^2}}. \quad (13)$$

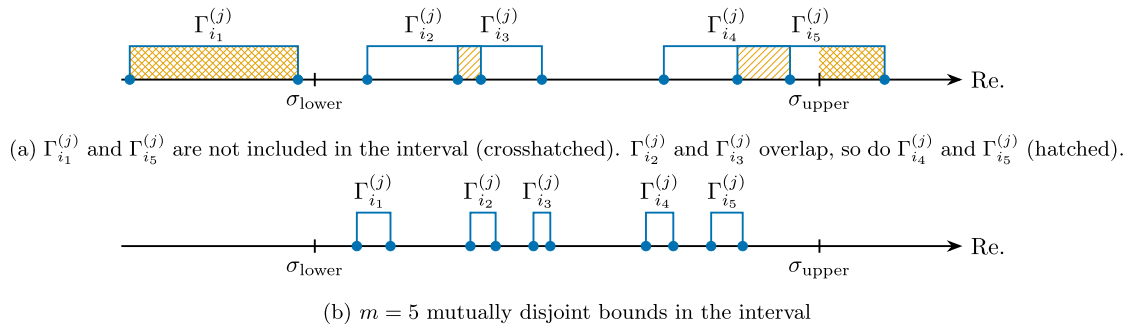
Proof. According to a classical result of the perturbation theory [19, Theorem 15.9.1], there is an eigenvalue λ_l of problem (1) for a scalar μ and non-zero vector \mathbf{u} such that:

$$|\lambda_l - \mu| \leq \frac{\|(A - \mu B)\mathbf{u}\|_{B^{-1}}}{\|B\mathbf{u}\|_{B^{-1}}}. \quad (14)$$

By substituting (μ, \mathbf{u}) in (14) with an approximate eigenpair $(\lambda_i^{(j)}, \mathbf{x}_{i,2}^{(j)})$ obtained by the SI Lanczos method, we have from (11) and residual (12) that:

$$|\lambda_{l_i^{(j)}} - \lambda_i^{(j)}| \leq \frac{\|(A - \lambda_i^{(j)}B)\mathbf{x}_{i,2}^{(j)}\|_{B^{-1}}}{\|B\mathbf{x}_{i,2}^{(j)}\|_{B^{-1}}} = \frac{\|\mathbf{r}_{i,2}^{(j)}\|_{B^{-1}}}{\|\mathbf{x}_{i,2}^{(j)}\|_B} = \frac{1}{|\tilde{\theta}_i^{(j)}|} \cdot \frac{|\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)} / \tilde{\theta}_i^{(j)}|}{\sqrt{1 + |\tilde{\beta}_j \mathbf{e}_j^T \tilde{\mathbf{y}}_i^{(j)} / \tilde{\theta}_i^{(j)}|^2}}. \quad (15)$$

By rewriting inequality (15), we have an eigenvalue error bound $\Gamma_i^{(j)}$ for $\lambda_{l_i^{(j)}}$ that includes an eigenvalue $\lambda_{l_i^{(j)}}$ of problem (1). \square

Fig. 2. Error bounds (13) for $m = 5$.

When bound (13) becomes sufficiently narrow, we can associate $\lambda_i^{(j)}$ with an eigenvalue of problem (1) and thus validate the index of the approximate eigenpairs. Here, assume that an interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$ contains m eigenvalues. If there are m error bounds in the interval that are mutually disjoint, as illustrated in Fig. 2(b), then each bound contains only one of the m eigenvalues of the interval. When the approximate eigenpairs have one-to-one correspondence with the eigenvalues of the interval, the index of each approximate eigenpair can be validated readily because we already know the index range of the eigenpairs of the interval from the bisection.

Algorithm 3 shows an implementation example, where the midpoint $\sigma = (\sigma_{\text{lower}} + \sigma_{\text{upper}})/2$ of the interval is selected as the shift for the SI Lanczos method. In line 8, we sort the approximate eigenpairs $(\lambda_i^{(j)}, \mathbf{x}_{i,2}^{(j)})$ of the method and re-index them for simplicity:

$$|\lambda_1^{(j)} - \sigma| \leq |\lambda_2^{(j)} - \sigma| \leq \dots \leq |\lambda_j^{(j)} - \sigma|, \text{ or equivalently } |\tilde{\theta}_1^{(j)}| \geq |\tilde{\theta}_2^{(j)}| \geq \dots \geq |\tilde{\theta}_j^{(j)}|. \quad (16)$$

Then, if the following two conditions hold, each $\lambda_i^{(j)}$ best approximates a distinct eigenvalue of the interval:

$$\text{inclusion: } \Gamma_i^{(j)} \subset [\sigma_{\text{lower}}, \sigma_{\text{upper}}) \text{ for } 1 \leq i \leq m. \quad (17)$$

$$\text{disjointness: } \Gamma_{i_1}^{(j)} \cap \Gamma_{i_2}^{(j)} = \emptyset \text{ for } 1 \leq i_1 < i_2 \leq m. \quad (18)$$

When (17) and (18) hold, we test for convergence in line 10. The algorithm is assumed to reach convergence when the relative residual 2-norm of each approximate eigenpair of the interval becomes less than a given tolerance τ_{res} . We also utilize the following criterion for the convergence test, which we refer to as the relative difference 2-norm between the approximate eigenvectors at the $(j-1)$ -th and j -th iterations:

$$\|\mathbf{x}_{i,2}^{(j)} - \mathbf{x}_{i,2}^{(j-1)}\|_2 / \|\mathbf{x}_{i,2}^{(j-1)}\|_2 < \tau_{\text{diff}} \text{ for } 1 \leq i \leq m. \quad (19)$$

Here, $\mathbf{x}_{i,2}^{(j-1)}$ and $\mathbf{x}_{i,2}^{(j)}$ are normalized to satisfy $\|\mathbf{x}_{i,2}^{(j-1)}\|_2 = \|\mathbf{x}_{i,2}^{(j)}\|_2$. As discussed in Section 4.3.3, criterion (19) is required because a small relative residual 2-norm does not necessarily imply that an approximate eigenvector is close to convergence. After the convergence test, we sort the approximate eigenpairs of the interval to the original order in line 11 to obtain the k -th eigenpair in line 12.

Algorithm 3: Computation of the k -th eigenpair.

Input : matrices A, B of generalized eigenvalue problem (1), target index k ,
interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$ containing the k -th eigenvalue,
 $\nu_{\text{lower}} = \nu_{\sigma_{\text{lower}}}(A, B)$, $\nu_{\text{upper}} = \nu_{\sigma_{\text{upper}}}(A, B)$,
tolerance τ_{res} for relative residual 2-norm, tolerance τ_{diff} for relative difference 2-norm.

Output: approximate eigenpair $(\hat{\lambda}_k, \hat{\mathbf{x}}_k)$, where $\|(A - \hat{\lambda}_k B)\hat{\mathbf{x}}_k\|_2 / \|\hat{\mathbf{x}}_k\|_2 < \tau_{\text{res}}$.

```

1  $l := k - \nu_{\text{lower}}$ ,  $m := \nu_{\text{upper}} - \nu_{\text{lower}}$ ,  $\sigma := (\sigma_{\text{lower}} + \sigma_{\text{upper}})/2$ ,
2 set a random starting vector  $\tilde{\mathbf{v}}_1$ ,  $\tilde{\mathbf{v}}_1 := \tilde{\mathbf{v}}_1 / \|\tilde{\mathbf{v}}_1\|_{B^{-1}}$ ,
3 for  $j = 1, 2, \dots$  do
4   compute  $j$ -step SI Lanczos decomposition (5) with reorthogonalization,
5   if  $j \geq m$  then
6     solve standard eigenvalue problem (6),
7     for  $i = 1$  to  $j$  do compute approximate eigenpairs  $(\lambda_i^{(j)}, \mathbf{x}_{i,2}^{(j)})$  as (7) and (9),
8     sort  $(\lambda_i^{(j)}, \mathbf{x}_{i,2}^{(j)})$  with  $1 \leq i \leq j$  to index as (16),
9     for  $i = 1$  to  $m$  do compute the  $i$ -th eigenvalue error bound (13),
10    if (17) and (18) and  $(\|(A - \lambda_i^{(j)} B)\mathbf{x}_{i,2}^{(j)}\|_2 / \|\mathbf{x}_{i,2}^{(j)}\|_2 < \tau_{\text{res}} \text{ for } 1 \leq i \leq m) \text{ and (19) then break,}$ 
11 sort  $(\lambda_i^{(j)}, \mathbf{x}_{i,2}^{(j)})$  with  $1 \leq i \leq m$  to index in increasing order  $\lambda_1^{(j)} < \lambda_2^{(j)} < \dots < \lambda_m^{(j)}$ ,
12  $(\hat{\lambda}_k, \hat{\mathbf{x}}_k) := (\lambda_l^{(j)}, \mathbf{x}_{l,2}^{(j)})$ .
```

3.3. Application to large-scale problems

We utilize a sparse direct linear solver for computing $\nu_\sigma(A, B)$ of large sparse A, B . Algorithm 1' shows a modification of Algorithm 1 to narrow down an initial interval. The main difference can be found in line 3, where a fill-reducing ordering Q is utilized to handle large sparse matrices, in addition to permutation P for numerical stability. In contrast to P , ordering Q is independent from shift σ because it depends on only the sparsity structure of $A - \sigma B$. Thus, once ordering and symbolic factorization are obtained for some shifted matrix, they can be recycled for other shifted matrices with varying shifts in the subsequent iterations. Computation of $\nu_\sigma(A, B)$ in Algorithm 2 can be performed in the same manner by utilizing a sparse direct linear solver. If shifted matrices are known to have a data-sparse (hierarchical low-rank) structure, their factorization can be done fast in a recursive manner [25–27].

Algorithm 1': Spectral bisection for the k -th eigenvalue problem.

Input : matrices A, B of generalized eigenvalue problem (1), target index k ,
initial interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$ containing the k -th eigenvalue,
 $\nu_{\text{lower}} = \nu_{\sigma_{\text{lower}}}(A, B)$, $\nu_{\text{upper}} = \nu_{\sigma_{\text{upper}}}(A, B)$, stopping criterion m_{max} .

Output : interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$, ν_{lower} , ν_{upper} .

```

1 repeat until  $\nu_{\text{upper}} - \nu_{\text{lower}} \leq m_{\text{max}}$                                 ▷  $\nu_{\text{upper}} - \nu_{\text{lower}}$ : number of eigenvalues in the interval
2    $\sigma := (\sigma_{\text{lower}} + \sigma_{\text{upper}})/2$ ,
3    $LDL^H \leftarrow PQ(A - \sigma B)Q^T P^T$ ,                                ▷  $Q$ : fill-reducing ordering for sparse matrices
4    $\nu := \nu_0(D, I)$ ,
5   if  $k \leq \nu$  then  $\sigma_{\text{upper}} := \sigma$ ,  $\nu_{\text{upper}} := \nu$  else  $\sigma_{\text{lower}} := \sigma$ ,  $\nu_{\text{lower}} := \nu$ .
```

3.4. Overview of the three-stage algorithm

Algorithm 4 presents a three-stage algorithm for solving the k -th eigenvalue problem. We first run Algorithm 2 to set an interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$ containing the k -th eigenvalue. We then run Algorithm 1' to narrow down the interval until it contains less than or equal to m_{max} eigenvalues that include λ_k . Compared with further narrowing down the interval to isolate λ_k from the other eigenvalues, approximately $\log_2 m_{\text{max}}$ bisection iterations can be reduced. Finally, we run Algorithm 3 to compute the k -th eigenpair along with the other eigenpairs of the interval while validating their index.

Algorithm 4: Three-stage algorithm for solving the k -th eigenvalue problem.

Input : matrices A, B of generalized eigenvalue problem (1), target index k ,
stopping criterion m_{max} for spectral bisection,
tolerance τ_{res} for relative residual 2-norm, tolerance τ_{diff} for relative difference 2-norm.

Output : approximate eigenpair $(\hat{\lambda}_k, \hat{\mathbf{x}}_k)$.

```

1 run Algorithm 2 to set an interval  $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$  containing  $\lambda_k$  and obtain  $\nu_{\text{lower}} = \nu_{\sigma_{\text{lower}}}(A, B)$  and  $\nu_{\text{upper}} = \nu_{\sigma_{\text{upper}}}(A, B)$ ,
2 run Algorithm 1' to narrow down the interval  $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$  until it contains less than or equal to  $m_{\text{max}}$  eigenvalues that include  $\lambda_k$ ,
3 run Algorithm 3 to obtain the  $k$ -th eigenpair  $(\hat{\lambda}_k, \hat{\mathbf{x}}_k)$  with its relative residual and difference 2-norms less than  $\tau_{\text{res}}$  and  $\tau_{\text{diff}}$ , respectively.
```

Here, m_{max} is an important parameter that influences the overall performance of the three-stage algorithm because there is a trade-off between the second and third stages (Algorithms 1' and 3), i.e., greater m_{max} results in fewer required bisection iterations, while more iterations are required for the SI Lanczos method. m_{max} was set the same for each problem in the numerical experiments. Tuning this parameter will be the focus of future work.

In the presence of multiple eigenvalues or a cluster of eigenvalues around λ_k , modification to Algorithm 4 is necessary because the algorithm is ineffective in detecting them and may end up in misconvergence; the stopping criterion (line 1 of Algorithm 1') for bisection and a convergence criterion (18) for the SI Lanczos method may not be satisfied. In addition, from the SI Lanczos method, only a one dimensional representation is obtained for the eigenspace corresponding to a multiple eigenvalue. To detect multiple or a cluster of eigenvalues during bisection, the length of the interval can be used along with the current stopping criterion. If detected, they and their corresponding eigenspace can be computed by, e.g., a block SI Lanczos method [28] whose block size can be determined from the number of eigenvalues in the interval. When a block eigensolver is used, the convergence criteria (line 10 of Algorithm 3) need to be modified accordingly to take account of the multiplicity or the cluster. Further investigation of the modification will be future work.

4. Numerical experiments

This section reports the numerical results of several real research problems from electronic structure calculations and a comparison of the proposed three-stage algorithm (Algorithm 4) and dense eigensolvers. In Section 4.1, we describe the matrix data used in the numerical experiments. Section 4.2 provides implementation details of dense eigensolvers and the three-stage algorithm. The numerical results are reported in Section 4.3.

Table 1
Matrix data.

Data	n	#nz	k	$[\lambda_1, \lambda_n]$	Material
APF4686	4686	53950	2343	$[-1.157, 5.581]$	amorphous-like conjugated polymer, poly(9,9-dioctyl-fluorene) [8]
AUNW9180	9180	1783313	5610	$[-0.210, 0.883]$	helical multishell gold nanowire with defects [29]
CPPE32346	32346	861764	16173	$[-1.169, 7.953]$	condensed polymer systems, poly(phenylene-ethynylene) [13,30]
NCCS430080	430080	10696416	215040	$[-1.195, 13.602]$	sp^2 - sp^3 nano-composite carbon solid [31]
VCNT1512000	1512000	294953351	336000	$[-1.098, 0.475]$	vibrating carbon nanotube within a supercell with spd orbitals [32]

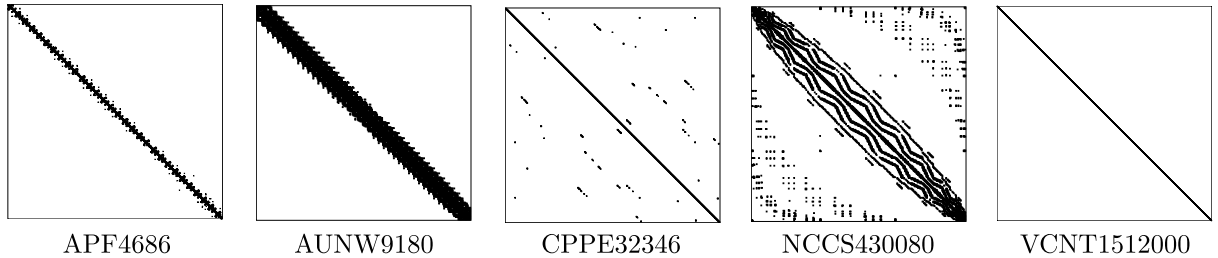


Fig. 3. Sparsity structures of matrix data.

4.1. Matrix data

Table 1 shows the matrix data used in the numerical experiments, which were generated by the ELSEES quantum mechanical nanomaterial simulator [8] and obtained from the ELSEES Matrix Library (<http://www.elses.jp/matrix/>). Here, the two $n \times n$ matrices A and B of each matrix data are real symmetric and real symmetric positive definite, respectively. A and B have the same sparsity structure shown in Fig. 3 with #nz non-zero elements (in their lower triangular part) in Table 1. The target index k is associated with the HO state. The entire spectrum is included in the interval $[\lambda_1, \lambda_n]$. The last column of Table 1 describes the origin of the matrix data.

4.2. Implementation details

As dense eigensolvers, we used the LAPACK [33] and ScaLAPACK [10] routines. Specifically, the LAPACK `dsygv` routine was used to solve APF4686 and AUNW9180. In `dsygv`, a generalized eigenvalue problem is transformed to a standard eigenvalue problem of a tridiagonal matrix, and then eigenpairs of the tridiagonal matrix are computed by the divide and conquer method [34–36]. To date, there is no single ScaLAPACK routine to perform the same task as `dsygv` in parallel. Therefore, to solve CPPE32346 and NCCS430080, the ScaLAPACK `pdpotrf`, `pdsygs`, `pdsytrd`, `pdstedc`, and `pdormtr` routines were utilized through EigenKernel (<https://github.com/eigenkernel/>) [12]. Note that the results for VCNT1512000 are not provided in the present paper because the problem size prevents it from being solved by a dense eigensolver in practical time.

In the three-stage algorithm (Algorithm 4), $v_\sigma(A, B)$ and solution of linear systems in the Lanczos and SI Lanczos methods were computed based on LDL^H factorization by the MUMPS sparse direct linear solver [37,38] with the METIS fill-reducing ordering [39]. The bisection narrowed down the initial interval until the number of eigenvalues in the interval became less than or equal to $m_{\max} = 20$. Tolerances for the relative residual and difference 2-norms in the SI Lanczos method were set to $\tau_{\text{res}} = 10^{-10}$ and $\tau_{\text{diff}} = 10^{-10}$, respectively. All codes were written in Fortran 90, and the numerical experiments were performed in double-precision.

4.3. Results

In this subsection, we first compare the k -th eigenpair computed by Algorithm 4 with that obtained by the dense eigensolvers described in Section 4.2 and report the computation time with some details about the computational environment and implementation. Then, in Sections 4.3.1 to 4.3.3, we present the detailed results of each algorithm (i.e., Algorithms 1', 2, and 3) of the three-stage algorithm.

Table 2 compares the k -th eigenpair of the three-stage algorithm ($\hat{\lambda}_k, \hat{\mathbf{x}}_k$) and that of the dense eigensolvers ($\lambda_k^{(d)}, \mathbf{x}_k^{(d)}$). As can be seen, at least 15 digits were the same for $\hat{\lambda}_k$ and $\lambda_k^{(d)}$. The last column shows the relative error 2-norm, where $\hat{\mathbf{x}}_k$ and $\mathbf{x}_k^{(d)}$ were normalized to satisfy $\|\hat{\mathbf{x}}_k\|_2 = \|\mathbf{x}_k^{(d)}\|_2$. The error norm had an order of magnitude less than -10 , indicating that the k -th eigenvector of the three-stage algorithm agrees well with that of the dense eigensolvers.

Table 3 shows the total computation time and computational resources consumed by Algorithm 4 (Alg. 4), its variant (Ger.), and the dense eigensolvers (Dense). In the variant, line 1 of Algorithm 4 was changed to set an interval including the

Table 2
 k -th eigenpair.

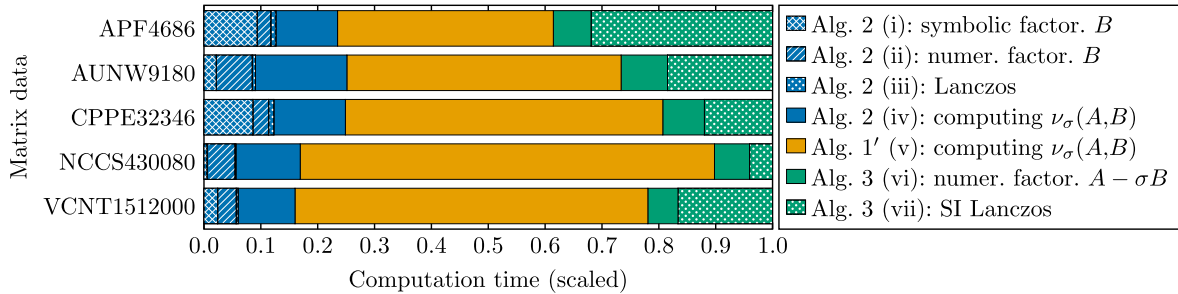
Data	k	$\hat{\lambda}_k$	$\lambda_k^{(d)}$	$\frac{ \hat{\lambda}_k - \lambda_k^{(d)} }{ \lambda_k^{(d)} }$	$\frac{\ \hat{\mathbf{x}}_k - \mathbf{x}_k^{(d)}\ _2}{\ \mathbf{x}_k^{(d)}\ _2}$
APF4686	2343	−0.4258775547956963	−0.4258775547956963	0	4×10^{-14}
AUNW9180	5610	0.1305388835941175	0.1305388835941177	2×10^{-15}	1×10^{-12}
CPPE32346	16173	−0.4332412034185730	−0.4332412034185731	2×10^{-16}	7×10^{-14}
NCCS430080	215040	−0.3689638375042860	−0.3689638375042869	2×10^{-15}	4×10^{-11}
VCNT1512000	336000	−0.5517499297808635	n/a	n/a	n/a

Table 3
Computation time and computational resources consumed by Algorithm 4, its variant, and dense eigensolvers.

Data	Time (s)			#Core			Memory (MB)		
	Alg. 4	Ger.	Dense	Alg. 4	Ger.	Dense	Alg. 4	Ger.	Dense
APF4686	0.3	0.8	82.1	1 ^(w)	1 ^(w)	1 ^(w)	13	12	629
AUNW9180	19.4	69.0	655.2			1 ^(w)	267	245	2836
CPPE32346	4.4	194.0	1366.8			32 ^(K)	121	116	33480
NCCS430080	2024	109597	10586			180000 ^(K)	5069	5055	5919002
VCNT1512000	5132	602525	n/a			n/a	38242	31618	n/a

(w) workstation with Xeon E5-2690 (2.90 GHz).

(K) K computer with SPARC64 VIIIfx (2.00 GHz) and Tofu interconnect.

**Fig. 4.** Computation time of algorithms and computational tasks.

entire spectrum based on the Gershgorin circle theorem.¹ Here, #Core is the number of cores used in the experiments, and superscripts ^(w) and ^(K) represent a workstation and the K computer, respectively. Memory indicates peak memory usage. Actual measurement of the memory usage was performed using the GNU `time` command. Estimation (in italics) shows the memory required to store $4n^2$ double-precision numbers, which is based on the memory requirement of the LAPACK `dsygv` routine.

Details about the computation time and implementation of the three-stage algorithm are shown in Fig. 4, where the total time is scaled to one. As described in the figure legend, the three-stage algorithm consists of Algorithms 1'–3, and our implementation can be divided into the following seven major computational tasks. (i) B is preprocessed in a symbolic manner to produce a fill-reducing ordering and an elimination tree for its LDL^H factorization. The ordering is recycled for the LDL^H factorization of shifted matrices $A - \sigma B$ because matrices A and B have the same sparsity structure in our numerical experiments. (ii) Based on the symbolic factorization, the numerical factorization of B is computed to solve linear systems in the Lanczos method. (iii) Ritz values are computed to set an initial interval. (iv–vi) Numerical factorization of shifted matrices is computed to set an initial interval, bisect the interval, and solve the linear systems in the SI Lanczos method. (vii) The k -th eigenpair is computed.

As can be seen in Fig. 4, Algorithm 1' dominates computation time as the problem size increases. This is because, as the problem size increases, more bisection iterations are expected to be required to narrow down an initial interval in order to make the number of eigenvalues in the interval less than or equal to $m_{\max} = 20$, which is the same value regardless of the problem size.

Fig. 5 shows the relationship between total computation time and the number of non-zero elements in factors L and D of LDL^H factorization, denoted #nzf, in log–log scale. Here, #nzf is the average of the factorizations of B and shifted

¹ Instead of some Gershgorin-type theorem, an inclusion set of the spectrum of $B^{-1}A$ was computed based on the original theorem because the diagonal dominance of A and B (described in Section 3.1) does not hold for all matrix data. To compute the inclusion set, columns of $B^{-1}A$ were obtained by solving linear systems with MUMPS, and then Gershgorin disks were calculated from the columns. In the VCNT1512000 case, the linear systems were solved in single-precision to reduce the time to solution.

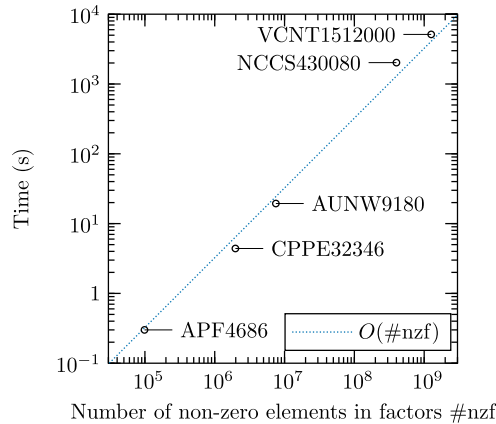


Fig. 5. Computation time vs. number of non-zero elements in factors L and D of LDL^H factorization.

Table 4

Initial interval.

Data	k	$[\sigma_{\text{lower}}, \sigma_{\text{upper}}]$	Length	$[i_{\text{lower}}, i_{\text{upper}}]$	m	Gap	Ratio
APF4686	2343	$[-0.777, 0.475]$	1.252	$[751, 3458]$	2708	5×10^{-4}	5.4
AUNW9180	5610	$[-0.079, 0.186]$	0.265	$[877, 5853]$	4977	5×10^{-5}	4.1
CPPE32346	16173	$[-0.731, 1.023]$	1.754	$[5586, 26409]$	20824	8×10^{-5}	5.2
NCCS430080	215040	$[-0.777, -0.275]$	0.502	$[64252, 224635]$	160384	3×10^{-6}	29.5
VCNT1512000	336000	$[-0.920, -0.429]$	0.491	$[84320, 422420]$	338101	1×10^{-6}	3.2

matrices $A - \sigma B$ with varying σ . The dotted line in the figure is of slope one, which corresponds to the linear scaling $O(\#nzf)$. As can be seen, computation time is proportional to $\#nzf$ (the slope for linear least squares fitting of the data points is 1.06). This is because the most time-consuming tasks in the three-stage algorithm, i.e., computation of $v_\sigma(A, B)$ and solving linear systems in the Lanczos and SI Lanczos methods, are performed based on factorizations by a sparse direct linear solver. Generally, the estimation of $\#nzf$ can be obtained in the symbolic factorization stage, which can be utilized to predict total computation time.

4.3.1. Initial interval

Table 4 shows the initial interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}}]$ obtained by Algorithm 2. The fourth column shows the length $\sigma_{\text{upper}} - \sigma_{\text{lower}}$ of the interval, which contains m eigenvalues with their index ranging from i_{lower} to i_{upper} . The seventh column shows the average gap between the eigenvalues in the interval defined as Length/m . In the last column, we compare the length of the initial interval with that of $[\lambda_1, \lambda_n]$ in Table 1 by calculating the ratio of $(\lambda_n - \lambda_1)/\text{Length}$. The initial intervals were 3.2 to 29.5 times narrower than $[\lambda_1, \lambda_n]$, i.e., the tightest interval that can be obtained from some Gershgorin-type theorem in general. Note that all problems required two iterations of Algorithm 2 (thus, two LDL^H factorizations) to set the interval, which is the minimum required iterations to obtain an interval validated as containing λ_k .

4.3.2. Bisection

The initial interval in Table 4 was narrowed down to the interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}}]$ in Table 5 based on Algorithm 1'. Fig. 6 shows the number of eigenvalues in the interval after each bisection iteration in log scale. The horizontal dotted line in the figure indicates the stopping criterion $m_{\text{max}} = 20$ for the bisection. In most cases, the number of eigenvalues was approximately halved after each iteration. However, the number remained unchanged after the fifth iteration of APF4686 and the sixth iteration of CPPE32346. In addition, there was a sharp decrease in the number of eigenvalues at the final iteration of CPPE32346, in which the number decreased by more than an order of magnitude. This convergence behavior implies that eigenvalues are distributed in a highly non-uniform manner and that there are clusters of eigenvalues or large gaps between eigenvalues. Indeed, in the CPPE32346 case, Gap in Table 5 is approximately 40 times greater than that shown in Table 4.

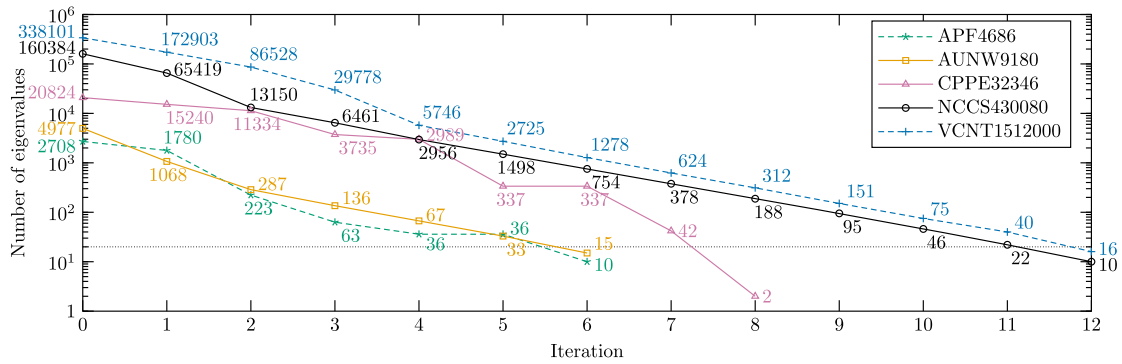
Here, we note that the straightforward bisection² required 47 to 49 iterations, which were roughly 4 to 8 times greater than those required for the three-stage algorithm shown in Fig. 6 (6 to 12 iterations), to compute the k -th eigenvalue to the accuracy very similar to that in Table 2.

² In the straightforward bisection, line 1 of Algorithm 1' was changed to use the length of the interval as a stopping criterion. Specifically, the stopping criterion was set to $(\sigma_{\text{upper}} - \sigma_{\text{lower}}) / \max\{|\sigma_{\text{lower}}|, |\sigma_{\text{upper}}|\} < 10^{-14}$. The k -th eigenvalue was computed using only bisection and from the equation $\hat{\lambda}_k = (\sigma_{\text{lower}} + \sigma_{\text{upper}})/2$.

Table 5

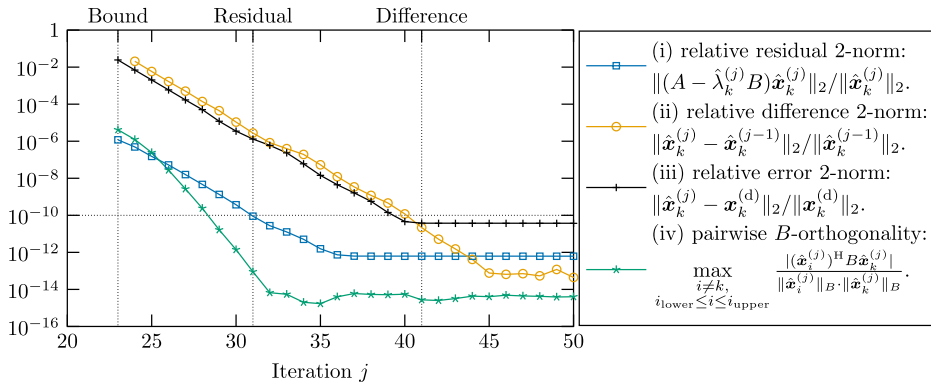
Interval narrowed down by bisection.

Data	k	$[\sigma_{\text{lower}}, \sigma_{\text{upper}})$	Length	$[i_{\text{lower}}, i_{\text{upper}}]$	m	Gap
APF4686	2343	$[-0.44450, -0.42494)$	0.01956	$[2334, 2343]$	10	2×10^{-3}
AUNW9180	5610	$[0.12826, 0.13240)$	0.00414	$[5601, 5615]$	15	3×10^{-4}
CPPE32346	16173	$[-0.43657, -0.42971)$	0.00685	$[16172, 16173]$	2	3×10^{-3}
NCCS430080	215040	$[-0.36897, -0.36884)$	0.00013	$[215040, 215049]$	10	1×10^{-5}
VCNT1512000	336000	$[-0.55178, -0.55166)$	0.00012	$[335995, 336010]$	16	8×10^{-6}

**Fig. 6.** Number of eigenvalues in an interval after each bisection iteration.**Table 6**

Iteration counts of the SI Lanczos method.

Data	m	Iteration		
		Bound	Residual	Difference
APF4686	10	24	33	37
AUNW9180	15	26	42	48
CPPE32346	2	7	19	23
NCCS430080	10	23	31	41
VCNT1512000	16	27	39	50

**Fig. 7.** Convergence history of the k -th eigenpair ($k = 215040$) of NCCS430080.

4.3.3. Computation of the k -th eigenpair

In Table 6, we show the iteration counts of Algorithm 3 for computing m eigenvalues in the interval $[\sigma_{\text{lower}}, \sigma_{\text{upper}})$ of Table 5. The third column is the iteration count required for (17) and (18) to be satisfied such that the index of each approximate eigenpair of the interval is validated. The fourth and fifth columns represent the iteration counts required for the relative 2-norm of the residual and difference (19) of each approximate eigenpair of the interval to become less than $\tau_{\text{res}} = 10^{-10}$ and $\tau_{\text{diff}} = 10^{-10}$, respectively.

Fig. 7 shows the convergence history of the k -th eigenpair ($k = 215040$) of NCCS430080. $(\hat{\lambda}_k^{(j)}, \hat{\mathbf{x}}_k^{(j)})$ in the figure legend denotes the k -th eigenpair computed at the j -th iteration of Algorithm 3. $\mathbf{x}_k^{(d)}$ represents the k -th eigenvector computed by the dense eigensolver. As described in the legend, the figure shows the relative 2-norm history of (i) the residual, (ii) the

difference between the $(j-1)$ -th and j -th iterations defined in (19), and (iii) the error compared with the dense eigensolver. The figure also shows (iv) the pairwise B -orthogonality (10) between the k -th eigenvector and the other $m-1$ eigenvectors of the interval in Table 5. Here, the three vertical dotted lines indicate the iteration counts of Bound, Residual, and Difference in Table 6. The horizontal dotted line indicates the convergence criteria $\tau_{\text{res}} = 10^{-10}$ and $\tau_{\text{diff}} = 10^{-10}$.

As can be seen in Fig. 7, a small residual norm does not necessarily imply that eigenvector $\hat{\mathbf{x}}_k^{(j)}$ is close to convergence. Indeed, the residual norm converged first, and convergence of the error and difference norms followed. Since the error norm cannot be measured in general, the difference norm (19) is utilized in Algorithm 3 to test for convergence.

5. Concluding remarks

The proposed three-stage algorithm obtained the validated k -th eigenpair $(\lambda_k, \mathbf{x}_k)$ for large sparse generalized Hermitian eigenvalue problems in electronic structure calculations with accuracy comparable with dense eigensolvers under limited computational resources. The three-stage algorithm (Algorithm 4) consists of Algorithms 1', 2, and 3, each of which has been found to be effective for computation of the eigenpair and validation of its index. In particular, from the numerical experiments, we have learned the following.

1. Algorithm 2 can set a narrow interval containing λ_k . The resulting intervals were 3 to 29 times narrower than $[\lambda_1, \lambda_n]$, i.e., the tightest interval that can be obtained from some Gershgorin-type theorem in general. In all experiments, only two iterations of Algorithm 2 were required, i.e., the minimum required iterations to obtain an interval validated as containing λ_k .
2. Algorithm 3 can compute the k -th eigenpair with high accuracy. The eigenpairs computed by Algorithm 3 agreed well with the results of dense eigensolvers, including a result obtained by a massively parallel eigenpair computation on the K computer. Specifically, the eigenvalues were the same to at least 15 digits, and the relative error norms of the eigenvectors were less than 10^{-10} .
3. By utilizing a sparse direct linear solver, large sparse matrices can be handled with efficiency. For example, a nano-composite carbon solid problem of size $n = 430080$ was solved in 0.6 hours using one core and 5.1 GB of memory on a workstation (a dense eigensolver required 2.9 hours using 180000 cores and an estimated 5.9 TB of memory on the K computer).

Fortran codes for the three-stage algorithm are available on GitHub (<https://github.com/lee-djl/k-ep>).

In future, we plan to examine parameter m_{max} . As explained in Section 3.4, this parameter influences the overall performance of the three-stage algorithm because there is a trade-off between the second and third stages (Algorithms 1' and 3). In addition, we plan to compare spectral bisection with its variants. As discussed in Section 2.1, it is possible to apply other root-finding algorithms to the second stage rather than bisection. As long as the algorithms are derivative-free and a root is bracketed in the algorithms, they can be readily applied to the second stage and can locate λ_k . Brent's method [40] is one such example. Finally, we plan to modify the three-stage algorithm to deal with the presence of multiple eigenvalues or a cluster of eigenvalues. Some possible modifications are described in Section 3.4.

Acknowledgements

The authors are very grateful to the reviewers for their careful reading of the manuscript and their valuable suggestions, which led to significant improvements in the presentation of the manuscript.

This work was supported by JSPS KAKENHI (17H02828, 16KT0016, 16K17550) and by MEXT, Japan as a Priority Issue (Creation of new functional devices and high-performance materials to support next-generation industries) to be tackled using a post-K computer system. This work used the computational resources of the K computer through the HPCI System Research Project (hp170147, hp170274), Oakforest-PACS through the JHPCN Project (jh170058-NAHI), and Sekirei provided by ISSP of the University of Tokyo.

Appendix A. Physical origin of the matrix eigenvalue problem

Practical electronic structure calculations use effective independent-electron theories, such as density functional theory [41–43], that are formulated with an effective Schrödinger-type equation for electronic wave functions $\{\psi_i(\mathbf{r})\}_{i \geq 1}$:

$$\mathcal{H}_{\text{eff}}\psi_i(\mathbf{r}) = \varepsilon_i\psi_i(\mathbf{r})$$

with the following Hamiltonian operator \mathcal{H}_{eff} :

$$\mathcal{H}_{\text{eff}} \equiv -\frac{\hbar^2}{2m_e}\Delta + V_{\text{eff}}(\mathbf{r}).$$

Here, m_e is the electron mass, and \hbar denotes the Planck constant. The scalar function $V_{\text{eff}}(\mathbf{r})$ is the potential function for electrons at the coordinate \mathbf{r} and varies among materials. The eigenvalues $\{\varepsilon_i\}$ are real and can be indexed in increasing order.

Independent-electron systems can be discretized by the Ritz variational method or the Galerkin method and can be formulated as a matrix eigenvalue problem. When an electronic wave function $\psi_i(\mathbf{r})$ is expanded (more accurately, approximated) by the linear combination of n non-orthogonal basis functions $\{\chi_j(\mathbf{r})\}_{j=1}^n$,

$$\psi_i(\mathbf{r}) = \sum_{j=1}^n x_j^{(i)} \chi_j(\mathbf{r}), \quad (\text{A.1})$$

generalized eigenvalue problem (1) appears with the $n \times n$ Hermitian matrices A and B whose i, j elements are defined as follows:

$$A_{ij} \equiv \int \chi_i^*(\mathbf{r}) \mathcal{H}_{\text{eff}} \chi_j(\mathbf{r}) d\mathbf{r}, \quad B_{ij} \equiv \int \chi_i^*(\mathbf{r}) \chi_j(\mathbf{r}) d\mathbf{r}. \quad (\text{A.2})$$

Here, the asterisk (*) denotes the complex conjugate of a function. Matrix B is positive definite from the definition, and its diagonal elements all equal one provided that the basis functions are normalized.

The size and structure of these matrices depend on the construction of the Hamiltonian operator \mathcal{H}_{eff} and the choice of the basis set $\{\chi_j(\mathbf{r})\}$. This paper is based on a first-principle-based modeled (transferable tight-binding) theory [8] in which basis functions, referred to as atomic orbitals, are localized in real space with their localization center being the position of an atom in a material. The index j of the basis functions χ_j can then be expressed as a composite of two indices l and m that represent a localization center, or an atom, and the shape of an orbital, respectively. Using indices l and m , we obtain an alternative expression for the expansion of wave functions (A.1):

$$\psi_i(\mathbf{r}) = \sum_{l=1}^{n_{\text{atom}}} \sum_{m=1}^{n_l} x_{lm}^{(i)} \chi_{lm}(\mathbf{r}).$$

Here, n_{atom} denotes the number of atoms in a material. n_l is the number of orbitals centered at the atom l and differs depending on the given atomic species. Therefore, the matrix size n is roughly proportional to the number of atoms n_{atom} in a material. Since orbitals are localized in real space, the matrix elements (A.2) decay rapidly as the distance between the localization centers of the orbitals increases. Thus, the matrices become sparse. Further details about the physical origin of the problem can be found in the literature [13].

Appendix B. Physical background of the k -th eigenvalue problem

The k -th eigenpair is associated with the HO state, which is in close relationship with several material properties, such as electronic transport and optical spectra [44, Chapter 2]. The target index k is a material-specific value that is uniquely determined by the number of electrons n_{elec} in a material. In para-spin materials calculations, which represent a typical case, the index is defined as one-half the number of electrons, or $k \equiv \lceil n_{\text{elec}}/2 \rceil$. The difference between the k -th and $(k+1)$ -th eigenvalues, or $\varepsilon_{k+1} - \varepsilon_k$, is referred to as the energy gap, which is crucial for electronic properties because the value is zero in metallic materials and non-zero in semiconducting or insulating materials. Therefore, the k -th and $(k+1)$ -th eigenvalues should be rigorously distinguished.

One of our recent motivations to address the k -th eigenvalue problem is electronic transport calculations of organic device materials using a quantum wave (wave packet) dynamics method [13,30]. In this method, an excited electronic wave (of an excited electron or a hole) is simulated by real-time dynamics with an effective time-dependent Schrödinger-type equation, and the wave function of the HO state or states near the HO state is set as the initial state of the wave.

References

- [1] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Natl. Bur. Stand.* 45 (4) (1950) 255–282.
- [2] A.V. Knyazev, Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method, *SIAM J. Sci. Comput.* 23 (2) (2001) 517–541.
- [3] T. Ericsson, A. Ruhe, The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems, *Math. Comput.* 35 (152) (1980) 1251–1268.
- [4] G.L.G. Sleijpen, H.A. van der Vorst, A Jacobi–Davidson iteration method for linear eigenvalue problems, *SIAM J. Matrix Anal. Appl.* 17 (2) (1996) 401–425.
- [5] T. Sakurai, H. Sugiura, A projection method for generalized eigenvalue problems using numerical integration, *J. Comput. Appl. Math.* 159 (1) (2003) 119–128.
- [6] E. Polizzi, Density-matrix-based algorithm for solving eigenvalue problems, *Phys. Rev. B* 79 (2009) 115112.
- [7] R. Li, Y. Xi, E. Vecharynski, C. Yang, Y. Saad, A thick-restart Lanczos algorithm with polynomial filtering for Hermitian eigenvalue problems, *SIAM J. Sci. Comput.* 38 (4) (2016) A2512–A2534.
- [8] T. Hoshi, S. Yamamoto, T. Fujiwara, T. Sogabe, S.-L. Zhang, An order- N electronic structure theory with generalized eigenvalue equations and its application to a ten-million-atom system, *J. Phys. Condens. Matter* 24 (16) (2012) 165502.
- [9] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, H. Lederer, The ELPA library: scalable parallel eigenvalue solutions for electronic structure theory and computational science, *J. Phys. Condens. Matter* 26 (21) (2014) 213201.

- [10] L.S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R.C. Whaley, *ScaLAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.
- [11] T. Imamura, S. Yamada, M. Machida, Development of a high-performance eigensolver on a peta-scale next-generation supercomputer system, *Prog. Nucl. Sci. Technol.* 2 (2011) 643–650.
- [12] H. Imachi, T. Hoshi, Hybrid numerical solvers for massively parallel eigenvalue computations and their benchmark with electronic structure calculations, *J. Inf. Process.* 24 (1) (2016) 164–172.
- [13] T. Hoshi, H. Imachi, K. Kumahata, M. Terai, K. Miyamoto, K. Minami, F. Shoji, Extremely scalable algorithm for 10^8 -atom quantum material simulation on the full system of the K computer, in: *Proceedings of the 7th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, 2016, pp. 33–40.
- [14] D.R. Bowler, T. Miyazaki, $O(N)$ methods in electronic structure calculations, *Rep. Prog. Phys.* 75 (3) (2012) 036503.
- [15] D. Lee, T. Miyata, T. Sogabe, T. Hoshi, S.-L. Zhang, An interior eigenvalue problem from electronic structure calculations, *Jpn. J. Ind. Appl. Math.* 30 (3) (2013) 625–633.
- [16] W. Givens, *Numerical Computation of the Characteristic Values of a Real Symmetric Matrix*, Tech. Rep. ORNL-1574, Oak Ridge National Laboratory, 1954.
- [17] J.R. Bunch, L. Kaufman, Some stable methods for calculating inertia and solving symmetric linear systems, *Math. Comput.* 31 (137) (1977) 163–179.
- [18] G. Peters, J.H. Wilkinson, Eigenvalues of $Ax = \lambda Bx$ with band symmetric A and B , *Comput. J.* 12 (4) (1969) 398–404.
- [19] B.N. Parlett, *The Symmetric Eigenvalue Problem*, Society for Industrial and Applied Mathematics, Philadelphia, 1998; First published by Prentice-Hall, Englewood Cliffs, 1980.
- [20] G. Meurant, *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*, Society for Industrial and Applied Mathematics, Philadelphia, 2006.
- [21] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [22] G.W. Stewart, Gershgorin theory for the generalized eigenvalue problem $Ax = \lambda Bx$, *Math. Comput.* 29 (130) (1975) 600–606.
- [23] V. Kostić, L.J. Cvetković, R.S. Varga, Geršgorin-type localizations of generalized eigenvalues, *Numer. Linear Algebra Appl.* 16 (11–12) (2009) 883–898.
- [24] Y. Nakatsukasa, Gershgorin's theorem for generalized eigenvalue problems in the Euclidean metric, *Math. Comput.* 80 (276) (2011) 2127–2142.
- [25] M. Mintner, The eigenvalue problem for the 2D Laplacian in \mathcal{H} -matrix arithmetic and application to the heat and wave equation, *Computing* 72 (3–4) (2004) 293–323.
- [26] P. Benner, T. Mach, Computing all or some eigenvalues of symmetric \mathcal{H}_t -matrices, *SIAM J. Sci. Comput.* 34 (1) (2012) A485–A496.
- [27] Y. Xi, J. Xia, R. Chan, A fast randomized eigensolver with structured LDL factorization update, *SIAM J. Matrix Anal. Appl.* 35 (3) (2014) 974–996.
- [28] R.G. Grimes, J.G. Lewis, H.D. Simon, A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems, *SIAM J. Matrix Anal. Appl.* 15 (1) (1994) 228–272.
- [29] T. Hoshi, T. Fujiwara, Domain boundary formation in helical multishell gold nanowires, *J. Phys. Condens. Matter* 21 (27) (2009) 272201.
- [30] H. Imachi, S. Yokoyama, T. Kaji, Y. Abe, T. Tada, T. Hoshi, One-hundred-nm-scale electronic structure and transport calculations of organic polymers on the K computer, *AIP Conf. Proc.* 1790 (2016) 020010.
- [31] T. Hoshi, Y. Akiyama, T. Tanaka, T. Ohno, Ten-million-atom electronic structure calculations on the K computer with a massively parallel order- N theory, *J. Phys. Soc. Jpn.* 82 (2) (2013) 023710.
- [32] J. Cerdá, F. Soria, Accurate and transferable extended Hückel-type tight-binding parameters, *Phys. Rev. B* 61 (2000) 7965–7971.
- [33] E. Anderson, Z. Bai, C. Bischof, L. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *LAPACK Users' Guide*, 3rd edition, Society for Industrial and Applied Mathematics, Philadelphia, 1999.
- [34] J.J.M. Cuppen, A divide and conquer method for the symmetric tridiagonal eigenproblem, *Numer. Math.* 36 (2) (1981) 177–195.
- [35] M. Gu, S.C. Eisenstat, A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem, *SIAM J. Matrix Anal. Appl.* 16 (1) (1995) 172–191.
- [36] F. Tisseur, J. Dongarra, A parallel divide and conquer algorithm for the symmetric eigenvalue problem on distributed memory architectures, *SIAM J. Sci. Comput.* 20 (6) (1999) 2223–2236.
- [37] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM J. Matrix Anal. Appl.* 23 (1) (2001) 15–41.
- [38] P.R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, *Parallel Comput.* 32 (2) (2006) 136–156.
- [39] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.* 20 (1) (1998) 359–392.
- [40] R.P. Brent, An algorithm with guaranteed convergence for finding a zero of a function, *Comput. J.* 14 (4) (1971) 422–425.
- [41] P. Hohenberg, W. Kohn, Inhomogeneous electron gas, *Phys. Rev.* 136 (3B) (1964) B864–B871.
- [42] W. Kohn, L.J. Sham, Self-consistent equations including exchange and correlation effects, *Phys. Rev.* 140 (4A) (1965) A1133–A1138.
- [43] W. Kohn, Nobel lecture: electronic structure of matter—wave functions and density functionals, *Rev. Mod. Phys.* 71 (1999) 1253–1266.
- [44] R.M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, Cambridge, 2004.