

GLOBALLY CONSTRUCTED ADAPTIVE LOCAL BASIS SET FOR SPECTRAL PROJECTORS OF SECOND ORDER DIFFERENTIAL OPERATORS*

YINGZHOU LI[†] AND LIN LIN[‡]

Abstract. Spectral projectors of second order differential operators play an important role in quantum physics and other scientific and engineering applications. In order to resolve local features and to obtain converged results, typically the number of degrees of freedom needed is much larger than the rank of the spectral projector. This leads to significant cost in terms of both computation and storage. In this paper, we develop a method to construct a basis set that is adaptive to the given differential operator. The basis set is systematically improvable, and the local features of the projector is built into the basis set. As a result the required number of degrees of freedom is only a small constant times the rank of the projector. The construction of the basis set uses a randomized procedure and only requires applying the differential operator to a small number of vectors on the global domain, while each basis function itself is supported on strictly local domains and is discontinuous across the global domain. The spectral projector on the global domain is systematically approximated from such a basis set using the discontinuous Galerkin method. The global construction procedure is very flexible and allows a local basis set to be consistently constructed even if the operator contains a nonlocal potential term. We verify the effectiveness of the globally constructed adaptive local basis set using one-, two- and three-dimensional linear problems with local potentials, as well as a one dimensional nonlinear problem with nonlocal potentials resembling the Hartree–Fock problem in quantum physics.

Key words. adaptive local basis, discontinuous Galerkin, spectral projector, differential operator, quantum physics, random sampling

AMS subject classifications. 65Z05, 65M99

DOI. 10.1137/17M1140236

1. Introduction. Consider the second order differential operator

$$(1.1) \quad H = -\Delta + V(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

where Ω is a rectangular, bounded domain in \mathbb{R}^d with periodic boundary conditions. V is a real, bounded, and smooth potential function. Then H is a self-adjoint operator on Ω . Using the eigen-decomposition $H\psi_i = \varepsilon_i\psi_i$, a spectral projector P is an integral operator with its kernel defined as

$$(1.2) \quad P(\mathbf{x}, \mathbf{x}') = \mathbf{1}_{\mathcal{I}}(H)(\mathbf{x}, \mathbf{x}') = \sum_{\varepsilon_i \in \mathcal{I}} \psi_i(\mathbf{x})\psi_i^*(\mathbf{x}').$$

Here \mathcal{I} is an interval that can be interpreted as an energy window indicating the eigenfunctions of interest, $\mathbf{1}_{\mathcal{I}}(\cdot)$ is an indicator function, and $\psi_i^*(\mathbf{x})$ is the complex

*Received by the editors July 24, 2017; accepted for publication (in revised form) November 20, 2018; published electronically January 22, 2019.

<http://www.siam.org/journals/mms/17-1/M114023.html>

Funding: The first author was partially supported by the supported by the National Science Foundation under award DMS-1328230 and the U.S. Department of Energy’s Advanced Scientific Computing Research program under award DE-FC02-13ER26134/DE-SC0009409. The second author was partially supported by the National Science Foundation under grant DMS-1652330 and by the U.S. Department of Energy under contract number DE-SC0017867, DE-AC02-05CH11231, and the Scientific Discovery through Advanced Computing (SciDAC) program.

[†]Department of Mathematics, Duke University, Durham, NC 27708-0320 (yingzhou.li@duke.edu).

[‡]Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720 and Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (linlin@math.berkeley.edu).

conjugation of $\psi_i(\mathbf{x})$. Denote by n the number of eigenfunctions in the summation of (1.2); then the rank of P is n . We assume that n is large, which can range from hundreds to hundreds of thousands. The spectral projector of such a form or of similar forms arises in many scientific and engineering problems. One notable example is the widely used Kohn–Sham density functional theory [20, 23] in quantum physics, where \mathcal{I} contains the lowest n eigenvalues of H . Typically, a large number of degrees of freedom associated with a fine numerical discretization are required to resolve the local features of ψ_i 's with sufficient accuracy. This is the case when standard discretization methods such as finite difference, finite element, discontinuous Galerkin (DG), planewave, and wavelet type of methods are used. The ratio between the total number of degrees of freedom (DOFs) and n can range from hundreds to hundreds of thousands in quantum physics applications [2, 40, 13]. As a consequence, both the storage cost and the computation cost associated with the spectral projector P can be large.

1.1. Contribution. In this paper, instead of using a general basis set, we introduce a new basis set that can be specifically tailored to represent the spectral projector P , for a given operator H and an interval \mathcal{I} . The key observation is as follows. Let us partition Ω into a suitable collection of nonoverlapping subdomains called *elements*. If the size of each element is small enough, then the numerical rank (also known as (a.k.a.) the approximate rank up to certain truncation tolerance ϵ [16]) of each row block of P restricted to any element can be bounded by a small constant. We shall quantify the details of the statement above later in the paper. The singular value decomposition of one such row block of P defines the *optimal* basis set on an element. Since the local features of the range of P is directly built into the basis set, we can expect that the number of DOFs in such an optimal basis set is much smaller than that in a general basis set. However, such an optimal basis set cannot be practically obtained, since it requires the knowledge of P a priori. We devise a numerical algorithm to compute a *nearly optimal* basis set. This is done by applying an approximate spectral projector, characterized by a matrix function $f(H)$, to a small number of random vectors defined on the global domain Ω . The number of random vectors is only slightly larger than the approximate rank of $f(H)$ restricted to each element. The range of P is then approximately a subspace of the span of this basis set, and we find that this is an efficient and accurate way to generate the basis functions on all elements. Due to the nonoverlapping condition, each basis function is only supported on one element and is discontinuous on the global domain Ω . We use the DG method [4] to patch the basis set to obtain an approximation to $\{\psi_i\}_{\varepsilon_i \in \mathcal{I}}$ or P . Motivated by our previous work of the locally constructed adaptive local basis (LC-ALB) set [29, 49, 21], the basis set in this work is dubbed the globally constructed adaptive local basis (GC-ALB) set. The LC-ALB approach has already been demonstrated to be able to be executed efficiently on massively parallel computers with over 100,000 processes and to efficiently perform large scale Kohn–Sham density functional theory calculations for systems over 20,000 atoms [21, 7].

The GC-ALB set has the following advantages: (1) systematically improvable. As the number of basis functions in each element increases, the accuracy of the projector represented in this basis set systematically improves towards the converged spectral projector. (2) Efficient. The number of basis functions is directly related to the numerical rank of the row blocks of the projector and is much smaller compared to the number of DOFs needed to resolve the local shape of $\{\psi_i\}$ in the real space. The

strict locality of the basis set can significantly reduce the computation and storage cost for $\{\psi_i\}$ and P . (3) Flexible. The construction of the basis set only requires matrix-vector multiplication of H defined on the global domain Ω . This allows existing matrix-vector multiplication routines for computing Hv to be readily used without the need of constructing auxiliary operators. This also facilitates generalizations to operators beyond the form in (1.1). This can occur, e.g., for $H = -\Delta + V + W$, where W is an integral operator and hence H becomes a nonlocal operator. Such an operator arises in applications such as the Kohn–Sham density functional theory with hybrid exchange–correlation functionals [8, 19] and the Hartree–Fock theory in quantum physics.

1.2. Related work. In the context of quantum physics, many tailored basis sets have been designed to reduce the number of DOFs to represent spectral projectors (or density matrices in physics terminology). Notable examples include the Gaussian basis set and the atomic orbital basis set [45, 38, 10]. Such basis sets are developed based on physical intuition and can provide relatively accurate solution with much reduced number of DOFs compared to more conventional basis sets. However, expert knowledge is often required to systematically converge the solution. These basis sets have also been used to “enrich” conventional basis sets to achieve a balance between the small number of DOFs and the systematic convergence property [43, 46]. However, the number of DOFs in the mixed basis representation is often much larger than those using Gaussian orbital or atomic orbital basis sets alone.

In order to achieve systematic convergence without sacrificing the number of DOFs, one may give up the concept of designing a basis set a priori, but instead generate the basis set *on the fly*. This has been demonstrated via a number of approaches based on filtration [29, 33, 41, 12, 37] as well as optimization [44, 30, 11, 36] principles. A common ingredient of these methods is to truncate the H operator into a series of operators defined only on different subdomains, and the basis set is then generated from the truncated operators. This requires each basis function to satisfy zero Dirichlet boundary conditions at each subdomain, which is not always achievable without sacrificing the accuracy of the resulting basis set. The method in this paper only uses matrix-vector multiplication on the global domain, and hence the concern from the choice of boundary conditions on local domains is completely removed. Our numerical results indicate that the GC-ALB set can also be more efficient than the LC-ALB set measured in terms of the number of DOFs to reach the same target accuracy. The partition of unity method (PUM) [6] is another commonly used option for enriching the basis set using local basis functions. Compared to PUM, the basis functions in the DG approach are strictly localized in nonoverlapping elements and hence are often better conditioned. In fact, one can easily obtain an orthonormal basis set in the DG approach through a local orthonormalization procedure. This also facilitates the usage of efficient numerical techniques such as the Chebyshev filtering techniques [51] for the computation of spectral projectors.

1.3. Outline of the paper. The rest of the paper is organized as follows. We review the interior penalty formulation of the DG framework and introduce the optimal discontinuous basis set and the locally constructed adaptive local basis set in section 2. We present the globally constructed adaptive local basis set in section 3. The numerical results are given in section 4, followed by the conclusion and discussion in section 5.

2. Preliminaries.

2.1. DG method. Without loss of generality, let $\Omega = (0, L)^d$ where $d = 1, 2, 3$, and \mathcal{K} be a regular partition of Ω into a set of nonoverlapping elements. For $\kappa \in \mathcal{K}$, we denote by $\bar{\kappa}$ the closure of κ . For any two elements $\kappa, \kappa' \in \mathcal{K}$, the periodic boundary condition on Ω implies that the partition is regular across the boundary $\partial\Omega$. We remark that generalizations to other boundary conditions such as Dirichlet or Neumann boundary conditions, as well as to nonrectangular domains, can be done with minor modification.

We denote by $H^1(\kappa)$ the standard Sobolev space of $L^2(\kappa)$ -functions such that the first partial derivatives are also in $L^2(\kappa)$. We denote the set of piecewise H^1 functions by

$$H^1(\mathcal{K}) = \{v \in L^2(\Omega) \mid v|_{\kappa} \in H^1(\kappa) \quad \forall \kappa \in \mathcal{K}\},$$

which is also referred to as the broken Sobolev space. For $v, w \in H^1(\mathcal{K})$, the inner product is

$$(2.1) \quad (v, w)_{\mathcal{K}} = \sum_{\kappa \in \mathcal{K}} (v, w)_{\kappa} := \sum_{\kappa \in \mathcal{K}} \int_{\kappa} v^*(\mathbf{x})w(\mathbf{x}) \, d\mathbf{x},$$

which induces a norm $\|v\|_{\mathcal{K}} = (v, v)_{\mathcal{K}}^{\frac{1}{2}}$.

For $v, w \in H^1(\mathcal{K})$ and $\kappa, \kappa' \in \mathcal{K}$, define the jump and average operators on a face $\bar{\kappa} \cap \bar{\kappa}'$ by

$$(2.2) \quad \{v\} = \frac{1}{2}(v|_{\kappa} + v|_{\kappa'}), \quad \{\nabla v\} = \frac{1}{2}(\nabla v|_{\kappa} + \nabla v|_{\kappa'}),$$

and

$$(2.3) \quad \llbracket v \rrbracket = v|_{\kappa} \mathbf{n}_{\kappa} + v|_{\kappa'} \mathbf{n}_{\kappa'}, \quad \llbracket \nabla v \rrbracket = \nabla v|_{\kappa} \cdot \mathbf{n}_{\kappa} + \nabla v|_{\kappa'} \cdot \mathbf{n}_{\kappa'},$$

where \mathbf{n}_{κ} denotes the exterior unit normal of the element κ .

In order to numerically solve the eigenvalue problem

$$H\psi_i = \varepsilon_i \psi_i,$$

we need to identify a basis set which spans a subspace of $H^1(\mathcal{K})$. Let N_{κ} be the number of DOFs on κ , and the total number of DOFs is $N_{\mathcal{K}} = \sum_{\kappa \in \mathcal{K}} N_{\kappa}$. Let $\mathbb{V}_N(\kappa) = \text{span} \{\varphi_{\kappa,j}\}_{j=1}^{N_{\kappa}}$, where each $\varphi_{\kappa,j}$ is a function defined on Ω with compact support only in κ . Hence $\mathbb{V}_N(\kappa)$ is a subspace of $H^1(\mathcal{K})$ and is associated with a finite dimensional approximation for $H^1(\kappa)$. Then $\mathbb{V}_N = \bigoplus_{\kappa \in \mathcal{K}} \mathbb{V}_N(\kappa)$ is a finite dimensional approximation to $H^1(\mathcal{K})$. We also assume all functions $\{\varphi_{\kappa,j}\}$ form an orthonormal set of vectors in the sense that

$$(2.4) \quad (\varphi_{\kappa,j}, \varphi_{\kappa',j'})_{\mathcal{K}} = \delta_{\kappa,\kappa'} \delta_{j,j'} \quad \forall \kappa, \kappa' \in \mathcal{K}, 1 \leq j \leq N_{\kappa}, 1 \leq j' \leq N_{\kappa'}.$$

The interior penalty formulation of the DG method [4] introduces the following bilinear form:

$$(2.5) \quad a(w, v) = \sum_{\kappa \in \mathcal{K}} \left[(\nabla w, \nabla v)_{\kappa} + (Vw, v)_{\kappa} \right] + \frac{1}{2} \sum_{\kappa \in \mathcal{K}} \left[-(\nabla w, \llbracket v \rrbracket)_{\partial\kappa} - (\llbracket w \rrbracket, \nabla v)_{\partial\kappa} \right] + \frac{1}{2} \sum_{\kappa \in \mathcal{K}} \left[\gamma_{\kappa} (\llbracket w \rrbracket, \llbracket v \rrbracket)_{\partial\kappa} \right].$$

Here the terms in the first bracket corresponds to the operator H . The terms in the second bracket are obtained from integration by parts, and the terms in the third bracket is a penalty term to guarantee the stability of the bilinear form [5]. The penalty parameter γ_κ on each element κ needs to be large enough, and the value of γ_κ depends on the choice of basis set \mathbb{V}_N . For general nonpolynomial basis functions the value of γ_κ is not known a priori. One possible solution is given in [31] which provides a formula for evaluating γ_κ on the fly for general nonpolynomial basis sets based on the solution of eigenvalue problems restricted to each element κ .

Using the bilinear form (2.5), the solution of

$$(2.6) \quad a(\psi_i^{\mathbb{V}_N}, v) = \varepsilon_i^{\mathbb{V}_N} (\psi_i^{\mathbb{V}_N}, v)_{\mathcal{K}} \quad \forall v \in \mathbb{V}_N$$

gives the numerical solution of eigenpairs of the form $(\varepsilon_i^{\mathbb{V}_N}, \psi_i^{\mathbb{V}_N})$ and $\psi_i^{\mathbb{V}_N} \in H^1(\mathcal{K})$. Equation (2.6) can be equivalently written as a standard linear eigenvalue problem

$$(2.7) \quad \sum_{\kappa', j'} H_{\kappa, j; \kappa', j'}^{\mathbb{V}_N} c_{\kappa', j'; i} = \varepsilon_i^{\mathbb{V}_N} c_{\kappa, j; i},$$

where $\{c_{\kappa, j; i}\}$ satisfies $\psi_i^{\mathbb{V}_N} = \sum_{\kappa, j} c_{\kappa, j; i} \varphi_{\kappa, j}$, and the reduced matrix $H^{\mathbb{V}_N}$ is of size $N_{\mathcal{K}} \times N_{\mathcal{K}}$ with matrix elements

$$(2.8) \quad H_{\kappa, j; \kappa', j'}^{\mathbb{V}_N} = a(\varphi_{\kappa, j}, \varphi_{\kappa', j'}).$$

Using the solution of (2.7), we can select $\varepsilon_i^{\mathbb{V}_N} \in \mathcal{I}$ and obtain an approximation to the spectral projector

$$(2.9) \quad P(\mathbf{x}, \mathbf{x}') \approx \sum_{\varepsilon_i^{\mathbb{V}_N} \in \mathcal{I}} \psi_i^{\mathbb{V}_N}(\mathbf{x}) \left(\psi_i^{\mathbb{V}_N}(\mathbf{x}') \right)^* = \sum_{\kappa, \kappa', j, j'} \varphi_{\kappa, j}(\mathbf{x}) \Gamma_{\kappa, j; \kappa', j'} \varphi_{\kappa', j'}^*(\mathbf{x}').$$

Here Γ is the $N_{\mathcal{K}} \times N_{\mathcal{K}}$ matrix representation of P in the basis set \mathbb{V}_N , and

$$(2.10) \quad \Gamma_{\kappa, j; \kappa', j'} = \sum_{\varepsilon_i^{\mathbb{V}_N} \in \mathcal{I}} c_{\kappa, j; i} c_{\kappa', j'; i}^*.$$

2.2. Optimal discontinuous basis set. The DG method in section 2.1 can be applied to very general basis set \mathbb{V}_N . Here we consider the *optimal* basis set \mathbb{V}_N for representing the spectral projector P with a discontinuous basis set. To simplify the discussion below, we also use linear algebra notation in this section when necessary. This means that we may not distinguish the kernel of an operator and a finite dimensional matrix consisting of its nodal values discretized on a fine set of real space grid points, with the number of grid points denoted by N_g . Then notation such as \mathbf{x}, \mathbf{x}' can be real space grid points in Ω , or row/column indices of vectors/matrices. We call $P(\mathbf{x}, \cdot) := \{P(\mathbf{x}, \mathbf{x}'), \mathbf{x}' \in \Omega\}$ a row vector and $P(\cdot, \mathbf{x}) := \{P(\mathbf{x}', \mathbf{x}), \mathbf{x}' \in \Omega\}$ a column vector, respectively. Similarly, we call $P(\kappa, \cdot) := \{P(\mathbf{x}, \mathbf{x}'), \mathbf{x} \in \kappa, \mathbf{x}' \in \Omega\}$ a row block and $P(\cdot, \kappa) := \{P(\mathbf{x}', \mathbf{x}), \mathbf{x} \in \kappa, \mathbf{x}' \in \Omega\}$ a column block, respectively.

Since the rank of the spectral projector P is n , if we choose a partition \mathcal{K} fine enough we may expect that the numerical rank of each row block $P(\kappa, \cdot)$ becomes small. In particular, note that the numerical of $P(\kappa, \cdot)$ cannot exceed the number of DOFs in κ , which is a constant and is independent of n . Our numerical results indicate that this rank can often be much lower than the number of DOFs in κ in practice. The singular value decomposition (SVD) of $P(\kappa, \cdot)$ can be written as

$$(2.11) \quad P(\kappa, \cdot) \approx \Phi_\kappa S_\kappa V_\kappa^*,$$

where S_κ is a diagonal matrix containing the leading N_κ singular values on κ , and $\Phi_\kappa(\mathbf{x}) = [\varphi_{\kappa,1}(\mathbf{x}), \dots, \varphi_{\kappa,N_\kappa}(\mathbf{x})]$ for $\mathbf{x} \in \kappa$. The support of each function $\varphi_{\kappa,j} \in H^1(\mathcal{K})$ is strictly in κ . Since Φ_κ 's are generated from the SVD of P , clearly the range of P is approximately contained in $\text{span}\{\varphi_{\kappa,j}\}$. For a given $\kappa \in \mathcal{K}$, the basis Φ_κ achieves the smallest error in 2-norm for representing $P(\kappa, \cdot)$ thanks to the optimal approximation property of the SVD [16] using N_κ basis functions. Hence the basis set $\{\Phi_\kappa\}_{\kappa \in \mathcal{K}}$ can be regarded as the optimal discontinuous basis set for representing P for a given set of DOFs $\{N_\kappa\}_{\kappa \in \mathcal{K}}$. We illustrate the decomposition (2.11) for the entire spectral projector P in Figure 2.1.

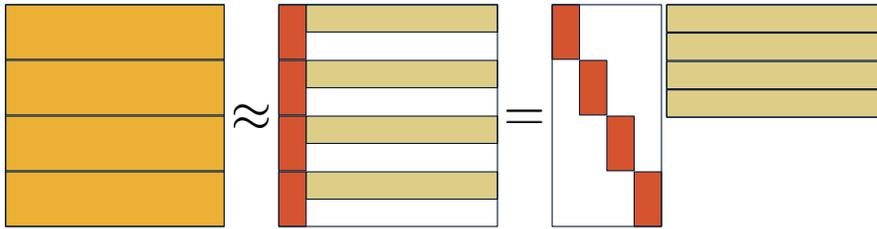


FIG. 2.1. Construction of the optimal discontinuous basis set for spectral projector. Left: the spectral projector P is partitioned into 4 row blocks. Middle: each $P(\kappa, \cdot)$ is low-rank factorized via SVD, i.e., $P(\kappa, \cdot) \approx \Phi_\kappa(S_\kappa V_\kappa^*)$. Right: subspace \mathbb{V}_N is assembled from $\{\Phi_\kappa\}$.

2.3. LC-ALB set. The optimal discontinuous basis set cannot be used for practical computation, since its construction depends on the knowledge of P . One possible approximation of such a basis set using nonpolynomial basis functions is the ALB set [29]. More specifically, we refer to this basis set the LC-ALB set, in order to distinguish from the globally constructed adaptive local basis set in section 3.

Consider the case that \mathcal{I} contains the lowest n eigenvalues of H . In the d -dimensional space, for each element κ , we form an *extended element* $\tilde{\kappa}$ around κ , and we refer to $\tilde{\kappa} \setminus \kappa$ as the *buffer region* for κ . Figure 2.2 illustrates a particular κ together with its buffer region. On $\tilde{\kappa}$ we solve the eigenvalue problem

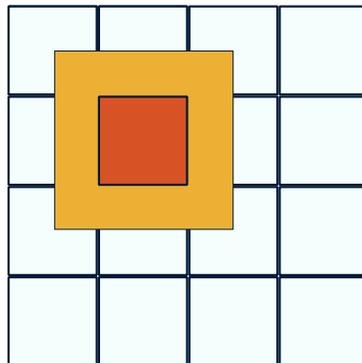


FIG. 2.2. The entire two-dimensional domain is partitioned into 4 by 4 blocks denoted by the white blocks. A particular element κ and its buffer region $\tilde{\kappa} \setminus \kappa$ are denoted as the red block and yellow block, respectively.

$$(2.12) \quad -\Delta \tilde{\varphi}_i + V \tilde{\varphi}_i = \lambda_i \tilde{\varphi}_i,$$

with certain boundary conditions on $\partial\tilde{\kappa}$. This eigenvalue problem can be solved using standard basis set such as finite difference, finite elements, or planewaves. For the numerical examples in this paper, the periodic boundary conditions is applied on each $\partial\tilde{\kappa}$, and the eigenvalue problem is solved via the pseudospectral method (the planewave basis set). Note that the size of the extended element $\tilde{\kappa}$ is independent of the size of the global domain and so is the number of basis functions per element. In order to obtain \mathbb{V}_N , the eigenfunctions corresponding to lowest N_κ eigenvalues are restricted from $\tilde{\kappa}$ to κ , i.e.,

$$\varphi_i(x) = \begin{cases} [\tilde{\varphi}_i]_\kappa(x), & x \in \kappa; \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, \dots, N_\kappa.$$

After orthonormalizing $\{\varphi_i\}$ locally on each element κ and removing the linearly dependent functions via a local SVD, the resulting set of orthonormal functions form the LC-ALB set.

The advantage of the LC-ALB set is that the basis functions for each element κ can be generated completely independently. However, due to the fictitious boundary conditions imposed on the extended element $\partial\tilde{\kappa}$, the effectiveness of the LC-ALB set depends on the size of the buffer region. On one extreme, if the size of the buffer region is 0 and when the periodic boundary condition is used, since V is in general not a periodic function on κ , the accuracy of the basis set can be severely affected by the Gibbs phenomena. On the other hand, if the buffer region is chosen to be too large, then the solution of the local eigenvalue problem (2.12) can become expensive. In practice we find that choosing $\tilde{\kappa}$ to contain κ and its $3^d - 1$ neighboring elements yields a relatively good balance between efficiency and accuracy, as has been demonstrated by the usage for solving PDEs [31, 32] and for solving practical Kohn–Sham equations [29, 21].

3. GC-ALB set. In this section, we propose a new strategy to construct an approximation to the optimal discontinuous basis set by using matrix-vector multiplication involving the operator H defined on the global domain Ω . This allows us to overcome the difficulty of choosing the boundary condition and the size of the extended element as in the LC-ALB set. Numerical results indicate that the resulting basis set is more effective in terms of the number of DOFs, and the strategy can be adapted to more general cases such as when V is an integral operator with a nonlocal kernel.

3.1. Formulation. We first introduce Algorithm 1, which is a variant of, e.g., Algorithm 4.1 in [18] for finding the approximate range of a numerically low rank matrix.

Algorithm 1. Randomized range finder for a given matrix A .

Input: $A \in \mathbb{C}^{p \times q}$. Approximate rank k .

Output: Left-singular vectors $U \in \mathbb{C}^{p \times k}$.

- 1: Generate an orthonormal random matrix $R \in \mathbb{C}^{q \times (k+c)}$ where c is a small oversampling constant.
 - 2: Compute $W = AR$.
 - 3: Perform the SVD for $W = USV^*$, with the diagonal entries of S ordered nonincreasingly.
 - 4: Return the first k columns of U .
-

If we treat A as a dense matrix and apply the SVD directly, the computational complexity will be $\mathcal{O}(pqk)$. On the other hand, Algorithm 1 only requires applying the matrix A to $(k + c)$ vectors, together with the SVD for W which costs $\mathcal{O}(pk^2)$ operations. Hence the randomized range finder algorithm significantly reduces the cost, if k is much smaller than q and if the matrix vector multiplication Av can be evaluated quickly. Usually, step 2 is the most expensive operation in Algorithm 1.

Assume \mathcal{K} is a partition of Ω so that each matrix row block $P(\kappa, :)$ is a numerically low rank matrix. If we apply Algorithm 1 to $P(\kappa, :)$, the output gives highly accurate approximation to the optimal basis set $\{\Phi_\kappa\}$ for $\kappa \in \mathcal{K}$. Furthermore, the random matrix R can be repeatedly used for different $\kappa \in \mathcal{K}$. Therefore, the matrix-vector multiplication for different matrix row blocks $P(\kappa, :)$ do not need to be applied independently. Instead it is equivalent to apply the entire matrix P to a random matrix R and to perform the SVD for each element independently to obtain an approximate range represented by Φ_κ for each $P(\kappa, :)$. The collection of the functions $\{\Phi_\kappa\}$ gives the GC-ALB set. Algorithm 2 describes this procedure for a general matrix $A \in \mathbb{C}^{N_g \times N_g}$, where N_g is the number of DOFs corresponding to a fine numerical discretization such as planewaves.

Algorithm 2. GC-ALB set for a given matrix A .

Input: $A \in \mathbb{C}^{N_g \times N_g}$. Partition $\mathcal{K} = \{\kappa\}$ with approximate rank for each element $\{N_\kappa\}$.

Output: The basis set $\{\Phi_\kappa\}$.

- 1: Generate an orthogonal random matrix $R \in \mathbb{C}^{N_g \times (\max_\kappa N_\kappa + c)}$, where c is a small oversampling constant.
 - 2: Compute $W = AR$.
 - 3: **for** $\kappa \in \mathcal{K}$ **do**
 - 4: Perform the SVD for $W(\kappa, :) = U_\kappa S_\kappa V_\kappa^*$, with the diagonal entries of S_κ ordered nonincreasingly.
 - 5: Obtain Φ_κ from the first N_κ columns of U_κ .
 - 6: **end for**
-

When taking the matrix A to be the spectral projector P , Figure 3.1 illustrates Algorithm 2 for the case that \mathcal{K} is partitions the domain Ω into 4 elements. Comparing to Figure 2.1 where each block of P is explicitly factorized, Figure 3.1 first applies

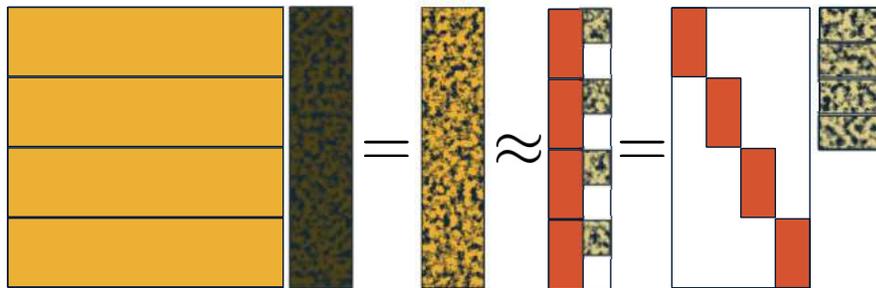


FIG. 3.1. GC-ALB functions. Left: the spectral projector P is applied to random vectors R . Middle-left: the result of the left part contains the column basis of each block. Middle-right: the column basis of each block is revealed via SVD. Right: subspace \mathbb{V}_N is assembled by $\{\Phi_\kappa\}$.

P to random vectors R and then factorizes each block of PR . Such an extra step is crucial here. Computing the dense P is expensive in terms of both computation and memory, whereas the matrix vector multiplication Pv can be efficiently calculated, which could be orders of magnitudes faster for large problems.

3.2. Rational approximation for matrix-vector multiplication. In order to construct the GC-ALB set for the projector P , Algorithm 2 requires an efficient method to compute the matrix-vector multiplication Pv . Since the spectral projector is a nonsmooth matrix function $\mathbf{1}_{\mathcal{I}}(\cdot)$, the computation of $Pv = \mathbf{1}_{\mathcal{I}}(H)v$ may still be a costly procedure. Fortunately, we only need Algorithm 2 to find an approximate range of P . Hence we may replace $\mathbf{1}_{\mathcal{I}}(\cdot)$ by a smooth function $f(\cdot)$, with the requirement that the support of f covers the interval \mathcal{I} , and that $f(H)v$ is relatively easy to compute. Then we can apply Algorithm 2 to find the approximate range of $f(H)$. The choice of f is certainly not unique. Here we use a modified Zolotarev's function to be $f(\cdot)$, which is an optimal rational approximation to the indicator function as to be demonstrated below.

Without loss of generality, we assume that $\mathcal{I} = [a, b]$ in the following discussion. Zolotarev's function $Z_{2r}(x; \ell)$ was initially proposed as the best rational approximant of type $(2r - 1, 2r)$ for the signum function $\text{sign}(x)$ on the interval $[-1, -\ell] \cup [\ell, 1]$ with a positive parameter $\ell < 1$ [52, 1]. Recently, it was composed with a Möbius transformation $T(x)$ and a linear transformation [17, 25] to become the best rational approximant of type $(2r, 2r)$ for an indicator function $\mathbf{1}_{[a,b]}(x)$ on the interval $(-\infty, a_-] \cup [a, b] \cup [b_+, +\infty)$, where a_- and b_+ are two parameters such that $a_- < a$ and $b < b_+$. Both the Möbius transformation $T(x)$ and the parameter ℓ in Zolotarev's function depend on a_- and b_+ . To be more precise, the Möbius transformation is defined as follows,

$$(3.1) \quad T(x) = \gamma \frac{x - \alpha}{x - \beta},$$

with $\alpha \in (a_-, a)$ and $\beta \in (b, b_+)$ such that

$$(3.2) \quad T(a_-) = -1, \quad T(a) = 1, \quad T(b) = \ell, \quad \text{and} \quad T(b_+) = -\ell.$$

Here, the variables α, β, γ , and ℓ are determined by a_-, a, b , and b_+ via solving the equations in (3.2). Combining with a simple linear transformation, $(x + 1)/2$, we arrive at a modified Zolotarev's function,

$$(3.3) \quad \begin{aligned} R(x) &= \frac{Z_{2r}(T(x); \ell) + 1}{2} \\ &= \frac{M}{2} \sum_{j=1}^r \frac{a_j \gamma}{\gamma^2 + c_{2j-1}} + \frac{1}{2} + \frac{M}{2} \sum_{j=1}^r \left(\frac{w_j}{x - \sigma_j} + \frac{\bar{w}_j}{x - \bar{\sigma}_j} \right), \end{aligned}$$

where γ is the same as in (3.1); M, a_j, c_j, w_j , and σ_j are constants as defined in [25]; and $\bar{\cdot}$ denotes the complex conjugate. $\{\sigma_j, \bar{\sigma}_j\}_{j=1}^r$ are known as the poles of the modified Zolotarev's function.

When the modified Zolotarev's function is used as $f(\cdot)$ and the matrix A in Algorithm 2 is replaced by $f(H)$, the line 2 in Algorithm 2 can be evaluated via

$$\begin{aligned}
 (3.4) \quad f(H)R &= \left(\frac{M}{2} \sum_{j=1}^r \frac{a_j \gamma}{\gamma^2 + c_{2j-1}} + \frac{1}{2} \right) R \\
 &\quad + \frac{M}{2} \sum_{j=1}^r \left(w_j (H - \sigma_j I)^{-1} R + \bar{w}_j (H - \bar{\sigma}_j I)^{-1} R \right).
 \end{aligned}$$

This requires solving $2r$ complex-shifted linear systems, where I denotes the identity matrix of the same size as H . If both H and R are real matrices, then $\bar{w}_j (H - \bar{\sigma}_j I)^{-1} R$ is the complex conjugate of $w_j (H - \sigma_j)^{-1} R$. Therefore, solving $2r$ shifted linear systems in (3.4) can be reduced to solving r shifted linear systems instead. These shifted linear systems can be solved via standard iterative methods such as generalized minimal residual method (GMRES) [42] and minimum residual method (MINRES) [39] with a preconditioner. The condition number of the shifted linear systems depends on the minimal distance between the eigenvalues of H and the shifts $\{\sigma_j, \bar{\sigma}_j\}$, as well as the spectral radius of H . The minimal distance can be systematically controlled by tuning the smoothness of the function $f(\cdot)$, through the adjustment of the interval (a_-, a) and (b, b_+) . When the spectral radius of H is large such as in the case of the planewave discretization, we find that the inverse of a shifted Laplacian [48] is an efficient preconditioner to reduce the condition number. Therefore, as shown in the numerical results, the number of iterations needed for solving the linear systems can be systematically controlled and relatively small.

Through the discussion above, the choice of a_- and b_+ remains to be determined. For a fixed indicator function $\mathbf{1}_{\mathcal{I}}$ with the given interval $\mathcal{I} = [a, b]$, a_- and b_+ determine the quality of the approximation of the modified Zolotarev’s function in (3.3). Generally, if either interval $[a_-, a]$ or $[b, b_+]$ becomes too narrow, it may require a large number of terms r in (3.4) to reach the same target accuracy. This translates to solving more shifted linear systems. However, the situation simplifies when $\mathcal{I} = [a, b]$ covers the lowest n eigenvalues of H , as will be demonstrated in the numerical results. Let the eigenvalues of H be $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq \lambda_{n+1} \leq \dots \leq \lambda_{N_g}$. The parameter a_- can be an arbitrary number in $(-\infty, a)$, and we can set a_- to be $-\infty$. The choice of b_+ relies on the spectrum property of H around b . When there is an eigenvalue gap around b ; i.e., $\lambda_n \leq b < \lambda_{n+1}$, b_+ is set to be λ_{n+1} or its estimated lower bound calculated via a few steps of Lanczos method [51]. In the case that H has continuous spectrum around b , we construct a small gap as $b_- = b$ and $b_+ = b + \delta$ for some small positive constant δ . The consequence of such a gap is that the approximated projector of P would include extra eigenvectors with nonzero weights. In practice, we find that the GC-ALB method is robust to this choice of δ . Such an observation even allows us to choose b_+ to be larger than λ_{n+1} even in the presence of a gap, in order to reduce r and hence the computational cost. When the location of λ_1, λ_n is not known a priori, similar to the situation in Chebyshev filtering techniques [51, 50], the initial guess of (a, b) can be efficiently obtained through a few Lanczos [24] iterations in practice.

Figure 3.2 gives an example of the modified Zolotarev’s function for the approximation of $\mathbf{1}_{\mathcal{I}}$ where $\mathcal{I} = [-1, 1]$. We assume 1 is in the continuous spectrum of H . We choose $a_- = -\infty$ and $b_+ = 1.1$ so that $r = 16$ is sufficient to approximate the indicator function with error below 10^{-10} in the interval $(a, b) \cup (b_+, \infty)$.

3.3. Complexity. In practical computation of the spectral projector, the following two scenarios are often encountered when counting the complexity with respect to the increase of the number of DOFs.

1. The size of the global domain Ω is fixed, and the number of DOFs increases due to the refinement of the discretization.

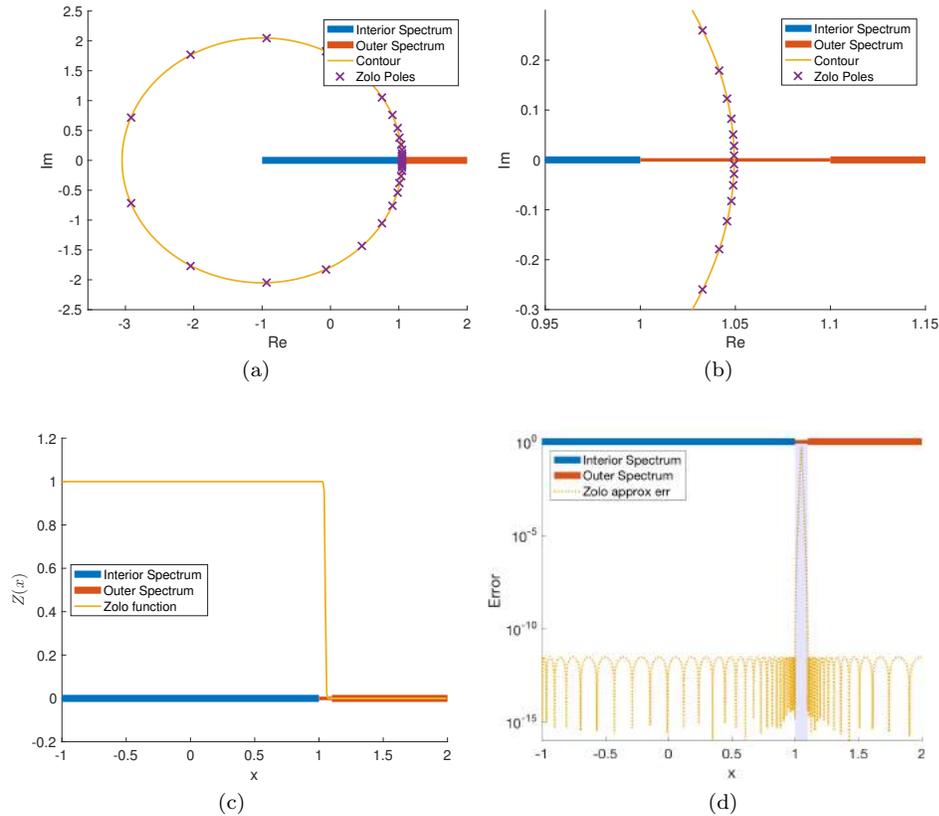


FIG. 3.2. An example with $\mathcal{I} = [-1, 1]$ and $(1, \infty)$ being the excluded spectrum. A modified Zolotarev's function is constructed with an artificial gap $(1, 1.1)$ and $r = 16$. (a) and (b) describe the locations of the poles on the complex plane. (c) is the modified Zolotarev's function and (d) is the absolute error comparing to indicator function $\mathbf{1}_{\mathcal{I}}(\cdot)$.

2. The size of the global domain Ω increases, and the number of DOFs increases proportionally to the volume of Ω .

Let M be the number of elements in \mathcal{K} , and N_g be the number of DOFs corresponding to a fine discretization on the global domain. For simplicity let all elements have the same number of basis functions denoted by n_b , and the number of DOFs corresponding to a fine discretization on κ is N_g/M . Hence the total number of basis functions is $N_{\mathcal{K}} = n_b M$. We also assume n_b is bounded by a constant while M, N_g can increase. In scenario 1, we increase N_g and fix M . In scenario 2, N_g is proportional to M while the ratio N_g/M is fixed.

The computational cost of the matrix vector multiplication associated with applying $f(H)$ to n_b random vectors is $N_{\text{pole}} N_{\text{it}} c_H n_b$. Here $N_{\text{pole}} = r$ is the number of poles in the rational approximation, N_{it} is the number of iterations to solve for each pole, and c_H is the cost of per iteration. Since $f(H)$ is a smooth function, N_{pole} is bounded by a constant independent of N_g, M, n_b . When a good preconditioner is available, N_{it} can also be bounded by a constant. c_H often is dominated by the matrix-vector multiplication associated with H . Furthermore, when V is a local potential and when the planewave basis set is used, the cost of applying H is dominated by applying the Laplacian operator which can be performed using the fast Fourier

transform (FFT). Then $c_H \sim \mathcal{O}(N_g \log N_g)$. Since n_b is fixed, in both scenarios the cost of the matrix-vector multiplication is $\mathcal{O}(N_g \log N_g)$. The cost of each SVD in step 4 is $\mathcal{O}((N_g/M) \times (n_b + c)^2)$, and the cost for all SVDs is $\mathcal{O}(N_g(n_b + c)^2)$. Hence the overall complexity for constructing the GC-ALB set is $\mathcal{O}(N_g \log N_g)$. Note that the LC-ALB approach uses a domain decomposition method, and the computational complexity of is trivially $\mathcal{O}(N_g)$. However, GC-ALB removes redundant calculations due to overlapping extended elements, and our numerical results indicate that the efficiency of the GC-ALB approach can be comparable or even faster when compared to LC-ALB.

The use of the GC-ALB set can also significantly reduce the storage cost for the spectral projector P . The storage cost for the GC-ALB set is $(N_g/M) \times n_b \times M = N_g n_b$. Viewed as a matrix, the storage cost for P is N_g^2 . This is generally very expensive, and P is usually stored using a low rank format as $P = \Psi\Psi^*$, where Ψ is of size $N_g \times n$. Then the storage cost for the coefficient matrix $c_{\kappa,j;i}$ as in (2.7) is $n_b M n$, and the total storage cost for representing P in the GC-ALB set is $N_g n_b + n_b M n$. Hence when the rank of the projector satisfies

$$n > \frac{n_b}{1 - n_b M / N_g},$$

the use of the GC-ALB set leads to reduction in the storage cost for Ψ . In practical applications such as Kohn–Sham equations, this condition is easy to satisfy since n increases with respect to the system size, while n_b is usually a constant on the order of $10 \sim 100$.

Similarly, the computational cost for Ψ using standard iterative eigensolvers is asymptotically dominated by the need of orthonormalizing Ψ when n is large. The complexity of this orthonormalization step scales as $\mathcal{O}(N_g n^2)$. In the GC-ALB set is constructed, the cost for the orthonormalization is reduced to $\mathcal{O}(n_b M n^2)$.

Algorithm 2 for constructing the GC-ALB set can also be efficiently parallelized. For the computation of $W = AR$, which is often the most time consuming step, the solution of the N_{pole} shifted linear systems for each column of R are all independent of each other. Therefore, the computation can be embarrassingly parallelized up to $N_{\text{pole}} n_b$ processors. If more than $N_{\text{pole}} n_b$ processors are available, they will be organized into $N_{\text{pole}} n_b$ processor groups, and the application of H can be parallelized as well. For the second part of Algorithm 2, all calculations can be carried out independently on each element.

3.4. Generalization to nonlocal potentials. Another advantage of the GC-ALB approach is that it handles local and nonlocal potentials on the same footing. The need of computing spectral projectors associated with nonlocal potentials arise, for instance, in solving the Hartree–Fock-like equations in quantum chemistry [47, 34]. The Hartree–Fock-like equations require the self-consistent computation of the projector

$$(3.5) \quad H[P] = -\frac{1}{2}\Delta + V_{\text{ion}} + V_{\text{Hxc}}[P] + V_X[P], \quad P = \mathbf{1}_{\mathcal{I}}(H[P]).$$

Here the interval \mathcal{I} contains the lowest n eigenvalues of $H[P]$. $V_{\text{ion}}, V_{\text{Hxc}}[P]$ are local potentials, and $V_X[P]$ is an integral operator with a nonlocal kernel. Here $[P]$ indicates the nonlinear dependence with respect to P . There is no natural way to consistently incorporate the nonlocal term $V_X[P]$ in the LC-ALB approach, while GC-ALB only requires the matrix-vector multiplication associated with $V_X[P]$. A detailed example of (3.5) will be given in section 4.2.

4. Numerical examples. We demonstrate the effectiveness of the GC-ALB method for finding the spectral projector for a linear problem in one, two, and three dimensions in section 4.1 and for a nonlinear problem in one dimension in section 4.2. Numerical examples are performed on Stanford Sherlock cluster bigmem node with quad socket Intel(R) Xeon(R) CPU E5-4640 @ 2.40GHz and 1.5 TB RAM. In all numerical examples, we assume the global domain Ω satisfies the periodic boundary condition. The pseudospectral discretization (a.k.a. the planewave basis set) provides the reference solution to the spectral projector, as well as the discretized operator for performing the matrix-vector multiplication on the global domain in order to construct the GC-ALB set. We measure the accuracy of the DG based methods in terms of the relative error of the eigenvalues within the range of the spectral projector compared to the reference solution, defined as

$$\frac{\sum_{i \in \mathcal{I}} |\varepsilon_i^{\mathbb{V}^N} - \varepsilon_i|}{\sum_{i \in \mathcal{I}} |\varepsilon_i|}.$$

The pseudospectral discretization can be identified with a set of uniform grid to discretize Ω . The integrals needed to construct the DG bilinear form is done using the Legendre–Gauss–Lobatto (LGL) grid. A Fourier interpolation procedure is used to interpolate functions from the uniform grid to the LGL grid, and a stable barycentric Lagrange interpolation [9] procedure is used to interpolate functions from the LGL grid back to the uniform grid when needed. In the rational approximation for the matrix vector multiplication, 16 poles on the upper half complex plane are actually solved. Since the potential function $V(x)$ in all numerical examples are real, the rest of the 16 poles are evaluated via the complex conjugation as in (3.4). For each pole, we use the GMRES [42] method to solve the associated equations with 30 being the restarting number and 10^{-12} being the tolerance. The preconditioner is the inverse of a shifted Laplacian [48] with the pole being the shift, which can be carried out efficiently using FFTs. The oversampling parameter c in Algorithm 2 is set to be 5. All pseudospectral discretized systems, including the systems for the reference solutions and the system on each extended element in LC-ALB method, are solved via the locally optimal block preconditioned conjugate gradient method (LOBPCG) [22], and the associated tolerance is 10^{-12} measured in terms of the maximal residual norm. We use the interior penalty formulation to patch the discontinuous basis functions to approximate the eigenfunctions, and the penalty parameter is determined automatically by solving a local eigenvalue problem as in [31].

4.1. Linear problems with local potentials.

4.1.1. One-dimensional case. Our first example is a second order differential operator (1.1) on $\Omega = (0, 2\pi)$ in one dimension. V is a local potential with four Gaussian potential wells at positions $x = \{1.0367, 2.4504, 3.8642, 5.2779\}$. The depth for each well is -10.0 whereas the standard deviation is set to be 0.2. Figure 4.1a shows the potential $V(x)$. The interval \mathcal{I} associated with the spectral projector P is assumed to cover the lowest 16 eigenvalues.

The global domain Ω is partitioned into 7 elements. Within each element, 40 LGL grid points are used to evaluate the integrals in the DG bilinear form accurately. The pseudospectral method discretizes Ω using 140 planewave basis functions, which can be identified with a uniform grid with 140 grid points. Under these settings, three adaptive local basis construction methods are considered, i.e., LC-ALB, GC-ALB with rational approximation for the projector, and the optimal basis set (Opt). For

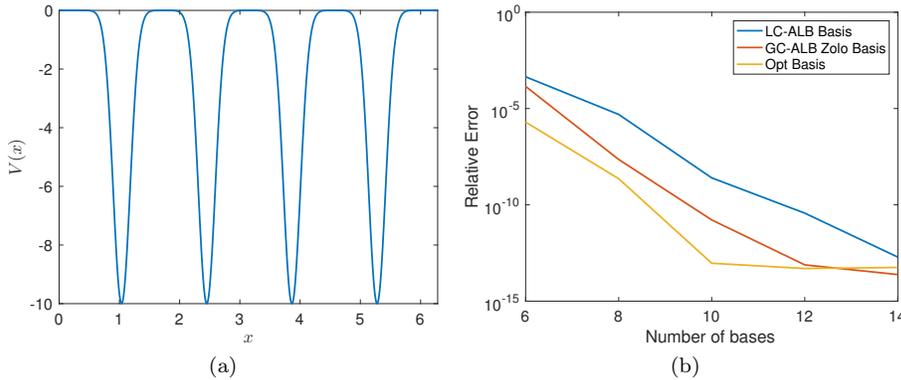


FIG. 4.1. (a) the one-dimensional potential function (b) the relative errors of LC-ALB, GC-ALB, and Opt method for different number of basis functions.

different methods, we vary the number of basis functions used in each element from 6 to 14. The relative error of the smallest 16 eigenvalues is measured against a reference solution, which is calculated via the pseudospectral method with 500 planewave basis functions directly. In the GC-ALB, we set the interval as $\mathcal{I} = [a, b] = [\lambda_1, \lambda_{16}]$ and the gap parameters as $a_- = -\infty$ and $b_+ = \lambda_{16} + 1.0$, where λ_1 and λ_{16} denote the smallest eigenvalue and the 16th smallest eigenvalue, respectively. Figure 4.1b shows the relative errors for different methods with varying number of basis functions. More details are reported in Table 4.1.

TABLE 4.1

Numerical results for GC-ALB method and LC-ALB method. n_b is the maximum number of basis functions for each element, err is the relative error of the smallest 16 eigenvalues, T_{Basis} and T_{DG} are the runtime for basis construction and DG solving, respectively, $n_{tot\ iter}$ is the total number of iterations for solving linear systems throughout the algorithm.

Method	n_b	err	T_{Basis} (sec)	T_{DG} (sec)	$n_{tot\ iter}$
GC-ALB	6	1.41e-04	5.69e-01	1.12e-02	147
	8	2.27e-08	6.14e-01	1.06e-02	147
	10	1.65e-11	6.22e-01	1.18e-02	148
	12	7.64e-14	7.21e-01	1.49e-02	149
	14	2.41e-14	7.24e-01	1.26e-02	150
LC-ALB	6	4.45e-04	2.91e-01	1.05e-02	-
	8	4.93e-06	3.24e-01	1.06e-02	-
	10	2.51e-09	3.59e-01	1.55e-02	-
	12	3.71e-11	3.91e-01	1.42e-02	-
	14	1.93e-13	4.20e-01	1.25e-02	-

For the one-dimensional operator, as shown in Figure 4.1b, the relative errors for all three methods decay exponentially as the number of basis functions increases. As discussed in section 2.2, the Opt basis defines the optimal discontinuous basis set for a given partition of the global domain and number of basis functions in each element, and this is confirmed in Figure 4.1b. On the other hand, the performance both GC-ALB and LC-ALB closely follow the Opt basis. Given the same number of basis functions, GC-ALB is about one digit more accurate than LC-ALB. When the number of basis functions is larger than or equal to 14, both methods reach the numerical accuracy limit and can not be further improved. The runtime of the

GC-ALB method and LC-ALB are about the same. The numbers of total iterations are about 148, which means the iteration number for solving each pole in (3.4) is on average smaller than 10.

4.1.2. Two-dimensional case. This example is a second order differential operator (1.1) on $\Omega = (0, 2\pi)^2$ in two dimensions. V is a local potential with four Gaussian wells as shown in Figure 4.2a. The depth for each well is -10.0 and the standard deviation is 0.2 . Similar to the one-dimensional example, the interval \mathcal{I} associated with the spectral projector P is assumed to cover the lowest 16 eigenvalues.

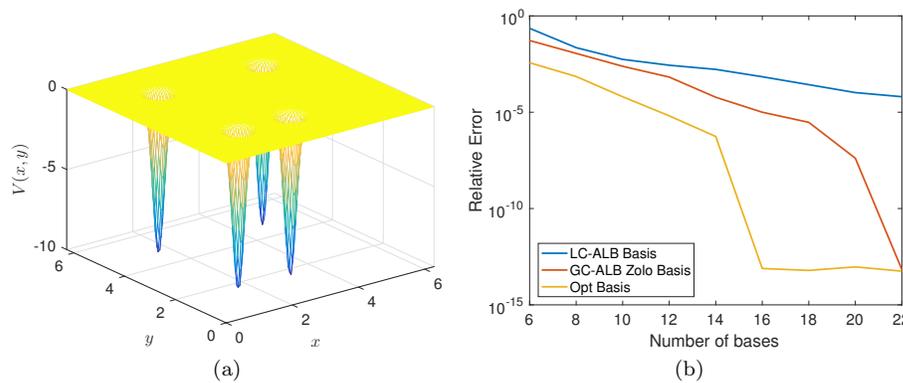


FIG. 4.2. (a) the two-dimensional potential function and (b) the relative errors of LC-ALB, GC-ALB, and Opt method for different number of basis functions.

The global domain Ω is partitioned into 7×7 elements. Within each element, 40×40 two-dimensional LGL grid points are used to evaluate the integrals in the DG bilinear form accurately. The pseudospectral method discretizes Ω using 140^2 planewave basis functions, which can be identified with a uniform two-dimensional grid with 140×140 grid points. Similar name conventions for LC-ALB, GC-ALB, and Opt are used as in the one-dimensional example. For different methods, we vary the number of basis functions used in each element from 8 to 22. The relative error of the smallest 16 eigenvalues is measured against a reference solution, which is calculated via the pseudospectral method with 300^2 planewave basis functions directly. In the GC-ALB, we set the interval as $\mathcal{I} = [a, b] = [\lambda_1, \lambda_{16}]$ and the gap parameters as $a_- = -\infty$ and $b_+ = \lambda_{16} + 0.1$, where λ_1 and λ_{16} denote the smallest eigenvalue and the 16th smallest eigenvalue. Figure 4.2b shows the relative errors for different methods with varying number of basis functions. More details are reported in Table 4.2.

Figure 4.2b shows that the differences among LC-ALB, GC-ALB, and Opt basis sets become more significant in two dimensions. The relative errors for Opt and GC-ALB decreases to the level of 10^{-14} when 16 and 22 basis functions are constructed for each element, respectively. On the other hand, the relative errors for LC-ALB method remains around 6.37×10^{-5} when 22 basis functions are used for each element. In order to achieve a relative error that is below 10^{-12} , we also find that 120 basis functions per element are needed in the LC-ALB approach. Table 4.2 shows that the cost for the GC-ALB and LC-ALB approaches are comparable in two dimensions. The fluctuation of the runtime in the LC-ALB approach is mostly due to the fluctuation of the number of iterations for the LOBPCG solver. In the GC-ALB approach, the

TABLE 4.2

Numerical results of GC-ALB method and LC-ALB method for the two-dimensional example.

Method	n_b	err	T_{Basis} (sec)	T_{DG} (sec)	$n_{tot\ iter}$
GC-ALB	6	5.27e-02	2.72e+01	2.53e-01	156
	8	1.13e-02	2.95e+01	2.80e-01	156
	10	2.43e-03	3.37e+01	3.29e-01	156
	12	6.67e-04	3.71e+01	3.95e-01	156
	14	5.98e-05	3.83e+01	4.76e-01	156
	16	9.99e-06	4.15e+01	4.61e-01	156
	18	2.98e-06	4.12e+01	6.31e-01	156
	20	4.10e-08	4.59e+01	6.51e-01	157
LC-ALB	22	6.87e-14	4.99e+01	6.78e-01	156
	6	2.27e-01	1.41e+01	2.21e-01	-
	8	2.25e-02	1.49e+01	2.45e-01	-
	10	5.53e-03	2.59e+01	3.12e-01	-
	12	2.75e-03	2.52e+01	3.75e-01	-
	14	1.67e-03	3.30e+01	4.75e-01	-
	16	6.95e-04	2.76e+01	5.17e-01	-
	18	2.69e-04	2.85e+01	5.90e-01	-
20	1.05e-04	2.90e+01	6.21e-01	-	
22	6.37e-05	4.38e+01	7.39e-01	-	

number of iterations for solving each pole here is around 9 on average for all cases, which gives $n_{tot\ iter}$ to be around 156 in all cases.

Below we demonstrate the weak scaling performance of the GC-ALB set in two dimensions. Starting from the potential in Figure 4.2a, we increase the size of the domain by periodically repeating the potential along x and y directions by a factor of ℓ . We vary ℓ from 1 to 6, as shown in Table 4.3, and the domain Ω is extended from $(0, 2\pi)^2$ to $(0, 12\pi)^2$. The number of planewave basis functions, the number of elements, as well as the number of eigenvalues to be computed, are proportional to the size of Ω . The parameters used within each element are the same as before, and 20 basis functions are constructed for each element. In terms of the parameters in Zolotarev’s function approximation, we set the interval as $\mathcal{I} = [a, b] = [\lambda_1, \lambda_n]$ and the gap parameters as $a_- = -\infty$ and $b_+ = \lambda_n + 0.1$, where λ_1 and λ_n denote the smallest eigenvalue and the n th smallest eigenvalue. The relative error of the smallest n eigenvalues is measured against reference solutions, which are calculated via the pseudospectral method with $(300\ell)^2$ planewave basis functions directly. Since the reference solution for $\ell = 6$ cannot be finished within the limited runtime on the Sherlock system, only the GC-ALB runtime and the total iteration number are reported here.

TABLE 4.3

Numerical results of the weak scaling of the GC-ALB method for the two-dimensional example. Here 20 basis functions are used within each element, ℓ denotes the number of repeated domain in each dimension, n denotes the number of calculated eigenvalues.

ℓ	Ω	n	err	T_{Basis} (sec)	T_{DG} (sec)	$n_{tot\ iter}$
1	$(0, 2\pi)^2$	16	1.14e-07	4.97e+01	8.74e-01	156
2	$(0, 4\pi)^2$	64	4.47e-07	3.17e+02	6.29e+00	197
3	$(0, 6\pi)^2$	144	5.78e-07	6.21e+02	4.64e+01	209
4	$(0, 8\pi)^2$	256	7.29e-07	2.09e+03	1.87e+02	271
5	$(0, 10\pi)^2$	400	6.33e-07	3.53e+03	6.11e+02	268
6	$(0, 12\pi)^2$	576	-	7.52e+03	1.66e+03	268

Table 4.3 shows that the relative errors are approximately the same for all ℓ . The runtime, T_{Basis} , for the basis construction in GC-ALB increases proportional to $\ell^2 \log \ell$, which means T_{Basis} is quasilinear in the number of plane-wave basis functions (see Figure 4.3a). Meanwhile T_{DG} , which is the cost for solving the DG problem is super-linear with respect to the number of plane-wave basis functions (see Figure 4.3b). T_{Basis} is consistent with the complexity analysis, and T_{DG} is close aligned with the complexity analysis when N_g is large. We observe that when ℓ is relatively small, the number of total iterations mildly increases with respect to ℓ . As ℓ keeps on increasing, the total iteration number stays around 270.

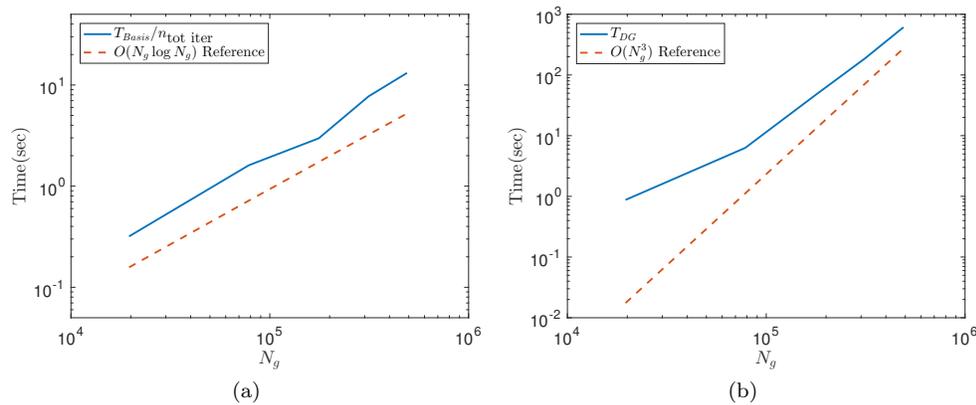


FIG. 4.3. Scalings of (a) average single iteration runtime and (b) DG solving time.

4.1.3. Three-dimensional case. This example is a second order differential operator (1.1) on $\Omega = (0, 2\pi)^3$ in three dimensions. V is a local potential with four Gaussian wells. The depth for each well is -10 whereas the standard deviation is set to be 0.2 . Figure 4.2a shows the isosurface for the potential function $V(x, y, z) = -1$. Similar to previous examples, the interval \mathcal{I} associated with the spectral projector P is assumed to cover the lowest 16 eigenvalues.

The global domain Ω is partitioned into $4 \times 4 \times 4$ elements. Within each element, $30 \times 30 \times 30$ three-dimensional LGL grid points are used to evaluate the integrals in the DG bilinear form accurately. The pseudospectral method discretizes Ω using 60^3 plane-wave basis functions, which can be identified with a uniform three-dimensional grid with $60 \times 60 \times 60$ grid points. Similar name conventions for LC-ALB, GC-ALB, and Opt are used as in the one-dimensional example. For different methods, we vary the number of basis functions used in each element from 8 to 24. The relative error of the smallest 16 eigenvalues is measured against a reference solution, which is calculated via the pseudospectral method with 100^3 plane-wave basis functions directly. In the GC-ALB, we set the interval as $\mathcal{I} = [a, b] = [\lambda_1, \lambda_{16}]$ and the gap parameters as $a_- = -\infty$ and $b_+ = \lambda_{16} + 0.01$, where λ_1 and λ_{16} denote the smallest eigenvalue and the 16th smallest eigenvalue. Figure 4.4b shows the relative errors for different methods with varying number of basis functions. More details are reported in Table 4.4.

For three-dimensional systems, the GC-ALB method exhibits even clearer advantage over the LC-ALB method. In Figure 4.4b and Table 4.4, the relative errors for GC-ALB method decay quickly to the level of 10^{-11} , while the asymptotic decay rate

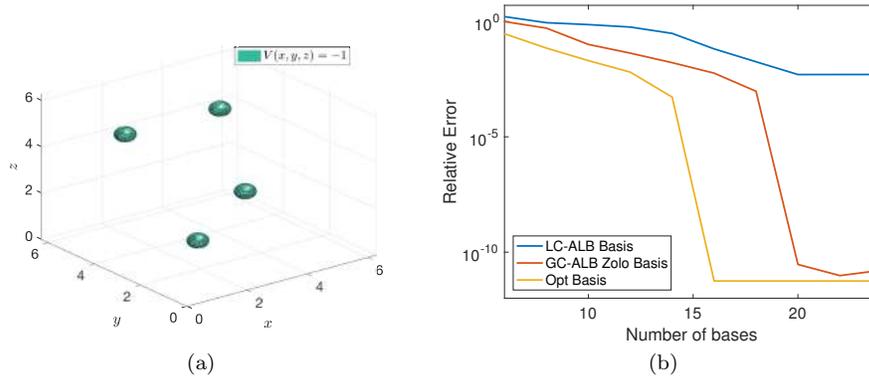


FIG. 4.4. (a) isosurface plot $V(x, y, z) = -1$ for the three-dimensional potential function and (b) the relative error of LC-ALB, GC-ALB, and Opt methods with different numbers of basis functions.

TABLE 4.4

Numerical results of GC-ALB method and LC-ALB method for the three-dimensional example.

Method	n_b	err	T_{Basis} (sec)	T_{DG} (sec)	n_{tot} iter
GC-ALB	6	9.89e-01	1.52e+02	3.07e+00	128
	8	5.15e-01	1.60e+02	3.60e+00	128
	10	1.01e-01	2.07e+02	4.34e+00	129
	12	4.24e-02	1.81e+02	4.87e+00	129
	14	1.64e-02	2.43e+02	5.68e+00	129
	16	5.76e-03	2.40e+02	6.48e+00	129
	18	9.42e-04	2.89e+02	7.66e+00	129
	20	2.92e-11	3.04e+02	8.30e+00	129
	22	9.63e-12	3.19e+02	8.84e+00	129
	24	1.65e-11	3.51e+02	1.23e+01	129
LC-ALB	6	1.62e+00	5.25e+02	3.15e+00	-
	8	8.84e-01	2.02e+03	3.31e+00	-
	10	7.32e-01	1.54e+03	5.23e+00	-
	12	5.73e-01	1.22e+03	6.18e+00	-
	14	3.04e-01	1.16e+03	7.05e+00	-
	16	6.48e-02	1.26e+03	7.82e+00	-
	18	1.79e-02	1.46e+03	9.00e+00	-
	20	5.00e-03	1.87e+03	9.19e+00	-
	22	5.02e-03	1.81e+03	1.07e+01	-
	24	5.04e-03	1.96e+03	1.20e+01	-

of the LC-ALB method is much slower. The GC-ALB approach is also more efficient in terms of the runtime. For most of the cases in Table 4.4, the GC-ALB method is about 6 times faster than LC-ALB method. The numbers of the applications of the operator to test vectors are 129 in GC-ALB method for all different number of bases. In addition, Table 4.5 shows that the number of DOFs for the GC-ALB set is much smaller than that needed for the planewave basis set to reach the same level of accuracy. Here the DOFs for the GC-ALB set is equal to the dimension of the DG matrix, and the DOFs for the planewave basis set is the number of planewave basis functions.

TABLE 4.5

Comparison of the degrees of freedom for GC-ALB method and planewave method for the three dimensional example.

	n_b	err	DOFs		n	err	DOFs
GC-ALB	14	1.69e-02	896	planewave	16	9.50e-03	4096
	16	4.34e-03	1024		20	2.13e-03	8000
	18	1.36e-04	1152		26	1.60e-04	17576
	20	9.19e-12	1280		60	1.10e-10	216000

4.2. Nonlinear problems with nonlocal potentials. In order to test the effectiveness of the GC-ALB approach for nonlocal potentials, we consider the following model for Hartree–Fock-like equations in one dimension. The Hamiltonian operator acting on a function ψ is given by

$$(4.1) \quad \begin{aligned} (H[P]\psi)(x) = & -\frac{1}{2} \frac{d^2}{dx^2} \psi(x) + \left(\int K(x, y)(m(y) + P(y, y)) dy \right) \psi(x) \\ & - \alpha \int K(x, y)P(x, y)\psi(y) dy. \end{aligned}$$

Compared to (3.5), the second term on the right-hand side of (4.1) corresponds to V_{ion} and $V_{\text{Hxc}}[P]$ and is a local potential, while the third term corresponds to $V_X[P]$ and is a nonlocal potential. Here $m(x) = \sum_{i=1}^M m_i(x - R_i)$, with the position of the i th nuclei denoted by R_i . Each function $m_i(x)$ takes the form

$$(4.2) \quad m_i(x) = -\frac{Z_i}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{x^2}{2\sigma_i^2}},$$

where Z_i is an integer representing the charge of the i th nucleus. Instead of using a bare Coulomb interaction, which diverges in one dimension, we adopt a Yukawa kernel

$$(4.3) \quad K(x, y) = \frac{2\pi e^{-\mu|x-y|}}{\mu\epsilon_0},$$

which satisfies the equation

$$(4.4) \quad -\frac{d^2}{dx^2} K(x, y) + \mu^2 K(x, y) = \frac{4\pi}{\epsilon_0} \delta(x - y).$$

As $\mu \rightarrow 0$, the Yukawa kernel approaches the bare Coulomb interaction given by the Poisson equation. The parameters ϵ_0, α are used to ensure that the contribution from different terms are comparable. And the notations here are different from the ones in section 3.2. In this example, we choose $\Omega = (0, 80)$, $M = 8$, $\sigma_i = 3.0$, $Z_i = 2.0$, $\mu = 0.01$, $\epsilon_0 = 10$, $\alpha = 0.05$. Besides these parameters, for the Zolotarev's function approximation in every iteration, 16 poles are used; $a_- = -\infty$, a is the smallest eigenvalue calculated each iteration; $b = -3.388$ which is the converged Fermi level; and $b_+ = 0$. The self-consistent spectral projector P is given by the lowest 16 eigenfunctions of $H[P]$.

In order to find the self-consistent spectral projector, we use a two level self consistent field (SCF) iteration that is commonly adopted to solve such Hartree–Fock-like equations [14, 26]. The SCF iterations are split into an outer loop and an inner loop. At the beginning of each outer SCF loop, we update the nonlocal potential $V_X[P]$ using a fixed point iteration; i.e., P is updated by the converged spectral projector P from the inner SCF loop. In the inner SCF loop, we fix the nonlocal potential $V_X[P]$ as if it were independent of P and update the local potential via the diagonal part of the projector $P(x, x)$ using the Anderson mixing method for charge mixing [3]. The convergence of the outer iteration is measured by the convergence of the exchange energy defined as

$$(4.5) \quad E_X = - \int P(x, y)K(x, y)P(x, y) dx dy.$$

In each inner SCF iteration, we apply the GC-ALB method with Zolotarev’s function approximation together with the DG method to construct the spectral projector efficiently, which is denoted as “GC-ALB” in the rest of this paper. As a comparison, we also conduct the inner and outer SCF iterations with the spectral projector calculated via planewave method, which is denoted as “planewave.”

Figure 4.5a and b show the convergence behavior of the two level SCF iterations using the GC-ALB set. Figure 4.5a shows the relative error of the total local potential for each inner SCF iteration, where x -axis is the total number of inner SCF iterations and each line represents the inner SCF iterations for an outer SCF iteration. The jump between the end of previous line and beginning of the next line is introduced by the update of the nonlocal potential. This is a typical behavior in the two-level SCF iteration for solving Hartree–Fock-like equations. As the converged spectral projector in the inner SCF iteration getting closer to the final convergence, the magnitude of the jump also decreases. Figure 4.5b shows the relative error of the energy associated with the nonlocal potential for each outer SCF iteration.

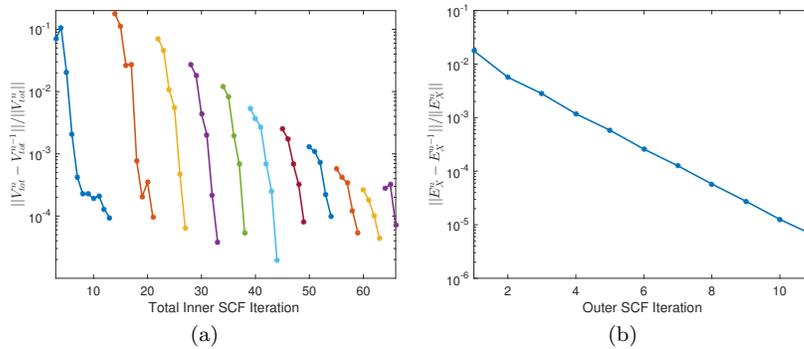


FIG. 4.5. GC-ALB: (a) The relative errors of the total local potential. Each point is a inner SCF iteration whereas each color line indicates a outer SCF iteration. (b) The relative errors of the energy associated with the nonlocal potential.

Table 4.6 indicates that the calculation using both the GC-ALB set and the planewave basis set converges within 11 outer SCF iterations to a relative error around 6×10^{-6} , and the number of inner iterations in each outer iteration is comparable in both methods. This indicates that the use of the GC-ALB set does not increase

TABLE 4.6

Comparison of the GC-ALB method and the planewave method in SCF iteration. The Hamiltonian operator defined in (4.1) is solved by two levels of SCF iteration combined with either the GC-ALB method or the planewave method. No. SCF_{in} denotes the number of inner SCF iterations, E_X denotes the energy associated with the nonlocal potential, and rel err is the relative change of the E_X every outer SCF iteration.

Outer SCF	GC-ALB			Planewave		
	No. SCF_{in}	E_X	rel err	No. SCF_{in}	E_X	rel err
1	10	-2.825403	1.77e-02	8	-2.825400	1.77e-02
2	7	-2.841545	5.68e-03	7	-2.841543	5.68e-03
3	6	-2.849557	2.81e-03	6	-2.849555	2.81e-03
4	5	-2.852925	1.18e-03	5	-2.852922	1.18e-03
5	6	-2.854591	5.84e-04	6	-2.854588	5.84e-04
6	5	-2.855331	2.59e-04	4	-2.855328	2.59e-04
7	6	-2.855691	1.26e-04	6	-2.855688	1.26e-04
8	5	-2.855855	5.73e-05	5	-2.855852	5.73e-05
9	5	-2.855934	2.77e-05	4	-2.855932	2.80e-05
10	4	-2.855970	1.27e-05	5	-2.855968	1.24e-05
11	3	-2.855989	6.45e-06	2	-2.855986	6.49e-06

the number of the SCF iterations in the nonlinear setup. The relative error from both methods also behaves similarly throughout the SCF iteration. The spectral projector, as well as electron density defined to be diagonal of the converged projector $\rho(x) = P(x, x)$, for both methods are given in Figure 4.6. The pointwise relative differences for the projector and the density are provided at the last row of Figure 4.6, where the errors are about the same level as that of the relative error in Table 4.6.

5. Conclusion. We developed a new method to construct an efficient basis set to represent the spectral projector of a second order differential operator H with reduced DOFs. For a given partition of the global domain into subdomains called elements, the optimal discontinuous basis set on any element can be given by the SVD of the matrix row block of the spectral projector associated with the element. Our GC-ALB can efficiently approximate such an optimal basis set in practice. The GC-ALB set can be obtained by only applying a matrix function $f(H)$ to a small number of random vectors on the global domain, without the need of any buffer areas to define a series of local problems. The GC-ALB set can be used in the context of the DG framework to approximate the spectral projector on the global domain. When the potential is local, the reduced DG matrix is a block sparse matrix. Hence the evaluation of the matrix representation of the spectral projector can be evaluated using fast methods based on sparse linear algebra operations, such as the pole expansion and selected inversion method [28, 27], and the purification methods [35, 15]. Our method is also flexible and can be applied to operators with local and nonlocal potentials. We verified the effectiveness of the basis set using one-, two-, and three-dimensional linear problems, as well as one-dimensional nonlocal, as well as nonlinear problems resembling Hartree–Fock problems. Numerical results indicate that the GC-ALB set achieve nearly optimal performance in terms of the number of DOFs per element, which reduces both the storage and the computational cost. In the near future, we will explore the usage of the GC-ALB set for Kohn–Sham density functional theory calculations for real materials.

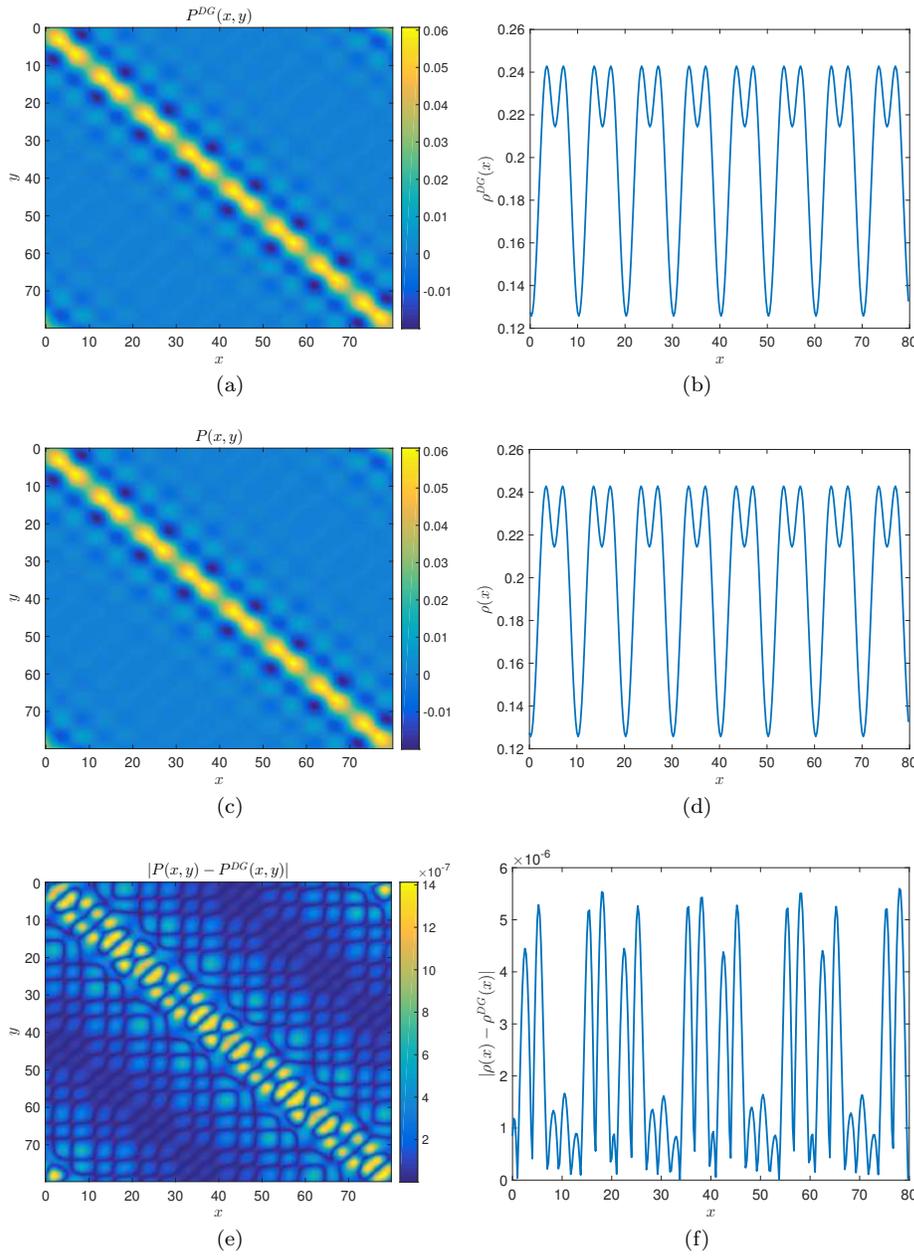


FIG. 4.6. (a) Kernel of the spectral projector and (b) electron density associated with the one-dimensional model for Hartree-Fock-like equation calculated by the GC-ALB method, whereas (c) kernel of the spectral projector and (d) electron density are calculated by the planewave method. (e) is the absolute difference between (a) and (c), and (f) is the absolute difference between (b) and (d).

Acknowledgments. We would like to thank Lexing Ying for fruitful discussions.

REFERENCES

- [1] N. I. AKHIEZER, *Elements of the Theory of Elliptic Functions*, American Mathematical Society, Providence, RI 1990.
- [2] M. ALEMANY, M. JAIN, L. KRONIK, AND J. CHELIKOWSKY, *Real-space pseudopotential method for computing the electronic properties of periodic systems*, Phys. Rev. B, 69 (2004), 075101.
- [3] D. G. ANDERSON, *Iterative procedures for nonlinear integral equations*, J. ACM, 12 (1965), pp. 547–560.
- [4] D. N. ARNOLD, *An interior penalty finite element method with discontinuous elements*, SIAM J. Numer. Anal., 19 (1982), pp. 742 – 760.
- [5] D. N. ARNOLD, F. BREZZI, B. COCKBURN, AND L. D. MARINI, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., 39 (2002), pp. 1749–1779.
- [6] I. BABUŠKA AND J. M. MELENK, *The partition of unity method*, Int. J. Numer. Methods Eng., 40 (1997), pp. 727–758.
- [7] A. S. BANERJEE, L. LIN, W. HU, C. YANG, AND J. E. PASK, *Chebyshev polynomial filtered subspace iteration in the discontinuous Galerkin method for large-scale electronic structure calculations*, J. Chem. Phys., 145 (2016), 154101.
- [8] A. D. BECKE, *Density functional thermochemistry. III. The role of exact exchange*, J. Chem. Phys., 98 (1993), 5648.
- [9] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric Lagrange interpolation*, SIAM Rev., 46 (2004), pp. 501–517.
- [10] V. BLUM, R. GEHRKE, F. HANKE, P. HAVU, V. HAVU, X. REN, K. REUTER, AND M. SCHEFFLER, *Ab initio molecular simulations with numeric atom-centered orbitals*, Comput. Phys. Commun., 180 (2009), pp. 2175–2196.
- [11] J. L. FATTEBERT AND J. BERNHOLC, *Towards grid-based $O(N)$ density-functional theory methods: Optimized nonorthogonal orbitals and multigrid acceleration*, Phys. Rev. B, 62 (2000), pp. 1713–1722.
- [12] C. J. GARCÍA-CERVERA, J. LU, Y. XUAN, AND W. E, *Linear-scaling subspace-iteration algorithm with optimally localized nonorthogonal wave functions for Kohn-Sham density functional theory*, Phys. Rev. B, 79 (2009), 115110.
- [13] L. GENOVESE, A. NEELOV, S. GOEDECKER, T. DEUTSCH, S. A. GHASEMI, A. WILLAND, D. CALISTE, O. ZILBERBERG, M. RAYSON, A. BERGMAN, AND R. SCHNEIDER, *Daubechies wavelets as a basis set for density functional pseudopotential calculations*, J. Chem. Phys., 129 (2008), 014109.
- [14] P. GIANNOZZI, S. BARONI, N. BONINI, M. CALANDRA, R. CAR, C. CAVAZZONI, D. CERESOLI, G. L. CHIAROTTI, M. COCCIONI, I. DABO, A. D. CORSO, S. DE GIRONCOLI, S. FABRIS, G. FRATESI, R. GEBAUER, U. GERSTMANN, C. GOUGOUSSIS, A. KOKALJ, M. LAZZERI, L. MARTIN-SAMOS, N. MARZARI, F. MAURI, R. MAZZARELLO, S. PAOLINI, A. PASQUARELLO, L. PAULATTO, C. SBRACCIA, S. SCANDOLO, G. SCLAUZERO, A. P. SEITSONEN, A. SMOGUNOV, P. UMARI, AND R. M. WENTZCOVITCH, *QUANTUM ESPRESSO: A modular and open-source software project for quantum simulations of materials*, J. Phys.: Condens. Matter, 21 (2009), pp. 395502–395520.
- [15] S. GOEDECKER, *Linear scaling electronic structure methods*, Rev. Mod. Phys., 71 (1999), pp. 1085–1123.
- [16] G. H. GOLUB AND C. F. VAN LOAN, *Matrix computations*, 4th ed. Johns Hopkins University Press, Baltimore, 2013.
- [17] S. GÜTTEL, E. POLIZZI, P. T. P. TANG, AND G. VIAUD, *Zolotarev quadrature rules and load balancing for the FEAST eigensolver*, SIAM J. Sci. Comput., 37 (2015), pp. A2100–A2122, <https://doi.org/10.1137/140980090>.
- [18] N. HALKO, P.-G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.
- [19] J. HEYD, G. E. SCUSERIA, AND M. ERNZERHOF, *Hybrid functionals based on a screened Coulomb potential*, J. Chem. Phys., 118 (2003), pp. 8207–8215.
- [20] P. HOHENBERG AND W. KOHN, *Inhomogeneous electron gas*, Phys. Rev., 136 (1964), pp. B864–B871.
- [21] W. HU, L. LIN, AND C. YANG, *DGDFE: A massively parallel method for large scale density functional theory calculations*, J. Chem. Phys., 143 (2015), 124110.

- [22] A. V. KNYAZEV, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comp., 23 (2001), pp. 517–541.
- [23] W. KOHN AND L. SHAM, *Self-consistent equations including exchange and correlation effects*, Phys. Rev., 140 (1965), pp. A1133–A1138.
- [24] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Stand., 45 (1950), pp. 255–282.
- [25] Y. LI AND H. YANG, *Spectrum slicing for sparse Hermitian definite matrices based on Zolotarev’s functions*, preprint, arXiv:1701.08935 [math.NA], 2017, <http://arxiv.org/abs/1701.08935>.
- [26] L. LIN, *Adaptively compressed exchange operator*, J. Chem. Theory Comput., 12 (2016), 2242.
- [27] L. LIN, M. CHEN, C. YANG, AND L. HE, *Accelerating atomic orbital-based electronic structure calculation via pole expansion and selected inversion*, J. Phys.: Condens. Matter, 25 (2013), 295501.
- [28] L. LIN, J. LU, L. YING, R. CAR, AND W. E, *Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems*, Comm. Math. Sci., 7 (2009), 755.
- [29] L. LIN, J. LU, L. YING, AND W. E, *Adaptive local basis set for Kohn-Sham density functional theory in a discontinuous Galerkin framework I: Total energy calculation*, J. Comput. Phys., 231 (2012), pp. 2140–2154.
- [30] L. LIN, J. LU, L. YING, AND W. E, *Optimized local basis function for Kohn-Sham density functional theory*, J. Chem. Phys., 231 (2012), 4515.
- [31] L. LIN AND B. STAMM, *A posteriori error estimates for discontinuous Galerkin methods using non-polynomial basis functions. Part I: Second order linear PDE*, Math. Model. Numer. Anal., 50 (2016), 1193.
- [32] L. LIN AND B. STAMM, *A posteriori error estimates for discontinuous Galerkin methods using non-polynomial basis functions. Part II: Eigenvalue problems*, Math. Model. Numer. Anal., (2017, in press).
- [33] L. LIN AND C. YANG, *Elliptic preconditioner for accelerating self consistent field iteration in Kohn-Sham density functional theory*, SIAM J. Sci. Comp., 35 (2013), pp. S277–S298.
- [34] R. MARTIN, *Electronic structure – Basic theory and practical methods*, Cambridge University Press, West Nyack, NY, 2004.
- [35] R. MCWEENY, *Some recent advances in density matrix theory*, Rev. Mod. Phys., 32 (1960), pp. 335–369.
- [36] S. MOHR, L. E. RATCLIFF, P. BOULANGER, L. GENOVESE, D. CALISTE, T. DEUTSCH, AND S. GOEDECKER, *Daubechies wavelets for linear scaling density functional theory*, J. Chem. Phys., 140 (2014), 204110.
- [37] P. MOTAMARRI AND V. GAVINI, *Subquadratic-scaling subspace projection method for large-scale Kohn-Sham density functional theory calculations using spectral finite-element discretization*, Phys. Rev. B, 90 (2014), 115127.
- [38] T. OZAKI, *Variationally optimized atomic orbitals for large-scale electronic structures*, Phys. Rev. B, 67 (2003), pp. 155108–155112.
- [39] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
- [40] J. E. PASK, B. M. KLEIN, C. Y. FONG, AND P. A. STERNE, *Real-space local polynomial basis for solid-state electronic-structure calculations: A finite-element approach*, Phys. Rev. B, 59 (1999), 12352.
- [41] M. J. RAYSON AND P. R. BRIDDON, *Highly efficient method for Kohn-Sham density functional calculations of 500–10000 atom systems*, Phys. Rev. B, 80 (2009), 205104.
- [42] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
- [43] K. SCHWARZ, P. BLAHA, AND G. K. H. MADSEN, *Electronic structure calculations of solids using the WIEN2k package for material sciences*, Comput. Phys. Commun., 147 (2002), pp. 71–76.
- [44] C. SKYLARIS, P. HAYNES, A. MOSTOFI, AND M. PAYNE, *Introducing ONETEP: Linear-scaling density functional simulations on parallel computers*, J. Chem. Phys., 122 (2005), 084119.
- [45] J. M. SOLER, E. ARTACHO, J. D. GALE, A. GARCÍA, J. JUNQUERA, P. ORDEJÓN, AND D. SÁNCHEZ-PORTAL, *The SIESTA method for ab initio order-N materials simulation*, J. Phys.: Condens. Matter, 14 (2002), pp. 2745–2779.
- [46] N. SUKUMAR AND J. E. PASK, *Classical and enriched finite element formulations for Bloch-periodic boundary conditions*, Int. J. Numer. Methods Eng., 77 (2009), 1121.
- [47] A. SZABO AND N. OSTLUND, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, McGraw-Hill, New York, 1989.

- [48] M. B. VAN GIJZEN, Y. A. ERLANGGA, AND C. VUIK, *Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian*, SIAM J. Sci. Comput., 29 (2007), pp. 1942–1958.
- [49] G. ZHANG, L. LIN, W. HU, C. YANG, AND J. E. PASK, *Adaptive local basis set for Kohn–Sham density functional theory in a discontinuous Galerkin framework II: Force, vibration, and molecular dynamics calculations*, J. Comput. Phys., 335 (2017), 426.
- [50] Y. ZHOU, J. R. CHELIKOWSKY, AND Y. SAAD, *Chebyshev-filtered subspace iteration method free of sparse diagonalization for solving the Kohn–Sham equation*, J. Comput. Phys., 274 (2014), pp. 770–782.
- [51] Y. ZHOU, Y. SAAD, M. L. TIAGO, AND J. R. CHELIKOWSKY, *Self-consistent-field calculations using Chebyshev-filtered subspace iteration*, J. Comput. Phys., 219 (2006), pp. 172–184.
- [52] E. ZOLOTAREV, *Application of elliptic functions to questions of functions deviating least and most from zero*, Zap. Imp. Akad. Nauk. St. Petersburg., 30 (1877), pp. 1–59.