

# Adaptive local basis set for Kohn–Sham density functional theory in a discontinuous Galerkin framework I: Total energy calculation

Lin Lin<sup>a,\*</sup>, Jianfeng Lu<sup>b</sup>, Lexing Ying<sup>c</sup>, Weinan E<sup>d</sup>

<sup>a</sup> Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544, United States

<sup>b</sup> Department of Mathematics, Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States

<sup>c</sup> Department of Mathematics and ICES, University of Texas at Austin, Austin, TX 78712, United States

<sup>d</sup> Department of Mathematics and PACM, Princeton University, Princeton, NJ 08544, United States

## ARTICLE INFO

### Article history:

Received 12 February 2011

Received in revised form 21 October 2011

Accepted 22 November 2011

Available online 2 December 2011

### Keywords:

Electronic structure

Kohn–Sham density functional theory

Discontinuous Galerkin

Adaptive local basis set

Enrichment functions

Eigenvalue problem

## ABSTRACT

Kohn–Sham density functional theory is one of the most widely used electronic structure theories. In the pseudopotential framework, uniform discretization of the Kohn–Sham Hamiltonian generally results in a large number of basis functions per atom in order to resolve the rapid oscillations of the Kohn–Sham orbitals around the nuclei. Previous attempts to reduce the number of basis functions per atom include the usage of atomic orbitals and similar objects, but the atomic orbitals generally require fine tuning in order to reach high accuracy. We present a novel discretization scheme that adaptively and systematically builds the rapid oscillations of the Kohn–Sham orbitals around the nuclei as well as environmental effects into the basis functions. The resulting basis functions are localized in the real space, and are discontinuous in the global domain. The continuous Kohn–Sham orbitals and the electron density are evaluated from the discontinuous basis functions using the discontinuous Galerkin (DG) framework. Our method is implemented in parallel and the current implementation is able to handle systems with at least thousands of atoms. Numerical examples indicate that our method can reach very high accuracy (less than 1 meV) with a very small number (4–40) of basis functions per atom.

© 2011 Elsevier Inc. All rights reserved.

## 1. Introduction

Electronic structure theory describes the energies and distributions of electrons, and is essential in characterizing the microscopic structures of molecules and materials in condensed phases. Among all the different formalisms of electronic structure theory, Kohn–Sham density functional theory (KSDFT) [1,2] achieves so far the best compromise between accuracy and efficiency, and has become the most widely used electronic structure model for condensed matter systems. Kohn–Sham density functional theory gives rise to a nonlinear eigenvalue problem, which is commonly solved using the self-consistent field iteration method [3]. In each iteration, the Kohn–Sham Hamiltonian is constructed from a trial electron density and is discretized into a finite dimensional matrix. The electron density is then obtained from the low-lying eigenfunctions, called Kohn–Sham orbitals, of the discretized Hamiltonian. The resulting electron density and the trial electron density are then mixed and form a new trial electron density. The loop continues until self-consistency of the electron density is reached. An efficient algorithm therefore contains three phases: discretization of the Hamiltonian; evaluation of the electron density

\* Corresponding author.

E-mail addresses: [linlin@math.princeton.edu](mailto:linlin@math.princeton.edu) (L. Lin), [jianfeng@cims.nyu.edu](mailto:jianfeng@cims.nyu.edu) (J. Lu), [lexing@math.utexas.edu](mailto:lexing@math.utexas.edu) (L. Ying), [weinan@math.princeton.edu](mailto:weinan@math.princeton.edu) (W. E).

from the discretized Hamiltonian; and self-consistent iteration. In this paper, we focus on the discretization of the Hamiltonian and the evaluation of the electron density in the pseudopotential framework [3].

If space is uniformly discretized, the Kohn–Sham Hamiltonian generally requires a basis set with a large number of degrees of freedom per atom. For most chemical systems, the kinetic energy cutoff typically ranges from 15 Ry to 90 Ry for standard planewave discretization in the norm-conserving pseudopotential framework [4], which amounts to about 500–5000 basis functions per atom. The required number of basis functions per atom is even larger for uniform discretization methods other than planewaves, such as finite difference method [5,6] and finite element method [7–9].

The large number of basis functions per atom originates from the rapid oscillation of the Kohn–Sham orbitals. The Kohn–Sham orbitals oscillate rapidly around the nuclei and become smooth in the interstitial region of the nuclei. Physical intuition suggests that the rapid oscillations around the nuclei are inert to changes in the environment. A significant part of the rapid oscillations can already be captured by the orbitals associated with isolated atoms. These orbitals are called atomic orbitals. Numerical methods based on atomic orbitals or similar ideas have been designed based on this observation [10–17]. Environmental effect is not built into the atomic orbitals directly, but can only be approximated by fine tuning the adjustable parameters in these atomic orbitals. The values of the adjustable parameters therefore vary among different chemical elements and exchange–correlation potentials, and sometimes vary among the different ambient environment of atoms. The quality of the atomic orbitals are difficult to be improved systematically, but relies heavily on the experience of the underlying chemical system.

Atomic orbitals and uniform discretization methods can be combined, as in the mixed basis methods [18–21]. The quality of the basis functions can therefore be systematically improved by incorporating the uniform discretization methods. However, fine tuning the adjustable parameters is still necessary due to the absence of the environmental effect in the basis functions, and in certain circumstances the number of basis functions per atom is still large.

In this paper we propose a novel discretization method to build the environmental effects into the basis set to achieve further dimension reduction of the basis set. The basis functions are constructed adaptively and seamlessly from the atomic configuration in local domains, called elements. The basis functions are discontinuous at the boundary of the elements, and they form the basis set used in the discontinuous Galerkin (DG) framework. The flexibility of the DG framework allows us to employ these discontinuous basis functions to approximate the continuous Kohn–Sham orbitals, and allows us to achieve high accuracy (less than 1 meV) in the total energy calculation with a very small number (4–40) of basis functions per atom. Our method is implemented in parallel with a rather general data communication framework, and the current implementation is able to calculate the total energy for systems consisting of thousands of atoms.

The discontinuous Galerkin framework has been widely used in numerical solutions of partial differential equations (PDE) for more than four decades, see for example [22–27] and the references therein. One of the main advantages of the DG method is its flexibility in the choice of the basis functions. The idea of constructing basis functions adaptively from the local environment has also been explored in other circumstances in numerical analysis such as reduced basis method [28–31] and multi-scale discontinuous Galerkin method [32–34] for solving PDE. In the current context, we apply the DG algorithm to solve eigenvalue problems with oscillatory eigenfunctions, and the basis functions are constructed by solving auxiliary local problems numerically.

The paper is organized as follows. Section 2 introduces the discontinuous Galerkin framework for Kohn–Sham density functional theory. The construction of the adaptive local basis functions is introduced in Section 3. Section 4 discusses implementation issues in more detail. The performance of our method is reported in Section 5, followed by the discussion and conclusion in Section 6.

## 2. Discontinuous Galerkin framework for Kohn–Sham density functional theory

### 2.1. Brief introduction of KSDFE

The Kohn–Sham energy functional in the pseudopotential framework [3] is given by:

$$E_{\text{KS}}(\{\psi_i\}) = \frac{1}{2} \sum_{i=1}^N \int |\nabla \psi_i|^2 dx + \int V_{\text{ext}} \rho dx + \sum_{\ell} \gamma_{\ell} \sum_{i=1}^N \left| \int b_{\ell}^* \psi_i dx \right|^2 + \frac{1}{2} \iint \frac{\rho(x)\rho(y)}{|x-y|} dx dy + \int \epsilon_{\text{xc}}[\rho(x)] dx, \quad (1)$$

where  $\rho(x) = \sum_i |\psi_i|^2(x)$  and the  $\{\psi_i\}$ 's satisfy the orthonormal constraints:

$$\int \psi_i^* \psi_j dx = \delta_{ij}. \quad (2)$$

In (1), we have taken the Kleinman–Bylander form of the pseudopotential [35]. The pseudopotential is given by

$$V_{\text{PS}} = V_{\text{ext}} + \sum_{\ell} \gamma_{\ell} |b_{\ell}\rangle \langle b_{\ell}|.$$

For each  $\ell$ ,  $b_{\ell}$  is a function supported locally in the real space around the position of one of the atoms,  $\gamma_{\ell} = +1$  or  $-1$ , and we have used the Dirac bra-ket notation. We have ignored the spin degeneracy and have adopted the local density approxima-

tion (LDA) [36,37] for the exchange–correlation functional. The proposed method can also be used for more complicated exchange–correlation functionals and when spin degeneracy is involved.

The Kohn–Sham equation, or the Euler–Lagrange equation associated with (1) reads

$$H_{\text{eff}}[\rho]\psi_i = \left( -\frac{1}{2}\Delta + V_{\text{eff}}[\rho] + \sum_{\ell} \gamma_{\ell} |b_{\ell}\rangle \langle b_{\ell}| \right) \psi_i = E_i \psi_i, \quad (3)$$

where the effective one-body potential  $V_{\text{eff}}$  is given by

$$V_{\text{eff}}[\rho](x) = V_{\text{ext}}(x) + \int \frac{\rho(y)}{|x-y|} dy + \epsilon'_{\text{xc}}[\rho(x)]. \quad (4)$$

Note that (3) is a nonlinear eigenvalue problem, as  $V_{\text{eff}}$  depends on  $\rho$ , which is in turn determined by  $\{\psi_i\}$ . The electron density is self-consistent if both (3) and (4) are satisfied. After obtaining the self-consistent electron density, the total energy of the system can be expressed using the eigenvalues  $\{E_i\}$  and  $\rho$  as [3]

$$E_{\text{tot}} = \sum_{i=1}^N E_i - \frac{1}{2} \iint \frac{\rho(x)\rho(y)}{|x-y|} dx dy + \int \epsilon_{\text{xc}}[\rho(x)] dx - \int \epsilon'_{\text{xc}}[\rho(x)] \rho(x) dx. \quad (5)$$

The goal of Kohn–Sham density functional theory is to calculate the total energy  $E_{\text{tot}}$  and the self-consistent electron density  $\rho$  given the atomic configuration.

Numerical algorithms for Kohn–Sham density functional theory can be broadly divided into two categories: one may try to directly minimize the energy functional (1) with respect to the Kohn–Sham orbitals  $\{\psi_i\}$  (see, e.g. [38]); one may also try to look for a solution for (3), usually by using the self-consistent iteration.

The self-consistent iteration goes as follows. Starting with an initial guess  $\rho_0$ , one looks for a solution of (3) iteratively:

1. Discretization of the Hamiltonian: determine the effective Hamiltonian  $H_{\text{eff}}[\rho_n]$  from the input density at the  $n$ th step  $\rho_n$ .
2. Evaluation of the electron density: obtain  $\tilde{\rho} = \sum_i |\psi_i|^2$  from the effective Hamiltonian  $H_{\text{eff}}[\rho_n]$ .
3. Self-consistent iteration: determine the input density at the  $(n+1)$ th step  $\rho_{n+1}$  from  $\rho_n$  and  $\tilde{\rho}$ , for instance:

$$\rho_{n+1} = \alpha \rho_n + (1 - \alpha) \tilde{\rho}$$

with some parameter  $\alpha$ .

4. If  $\|\rho_n - \tilde{\rho}\| \leq \delta$ , stop; otherwise, go to step (1) with  $n \leftarrow n + 1$ .

**Remark.** The mixing step above is called linear mixing in the literature, which is the simplest choice. More advanced mixing schemes [39,40] can be used as well. The mixing scheme used in our current implementation is the Anderson mixing scheme [39], but we will not go into the details of mixing schemes in this work.

In this paper we focus on the discretization of the Hamiltonian and the evaluation of the electron density. Given an effective potential  $V_{\text{eff}}$ , we find  $\tilde{\rho}$  from

$$\tilde{\rho}(x) = \sum_{i=1}^N |\psi_i|^2(x), \quad (6)$$

where the  $\{\psi_i\}$ 's are the first  $N$  eigenfunctions of  $H_{\text{eff}}$ .

$$H_{\text{eff}}\psi_i = \left( -\frac{1}{2}\Delta + V_{\text{eff}} + \sum_{\ell} \gamma_{\ell} |b_{\ell}\rangle \langle b_{\ell}| \right) \psi_i = E_i \psi_i. \quad (7)$$

Note that the  $\{\psi_i\}$ 's minimize the variational problem

$$E_{\text{eff}}(\{\psi_i\}) = \frac{1}{2} \sum_{i=1}^N \int |\nabla \psi_i(x)|^2 dx + \int V_{\text{eff}}(x) \rho(x) dx + \sum_{\ell} \gamma_{\ell} \sum_{i=1}^N |\langle b_{\ell}, \psi_i \rangle|^2, \quad (8)$$

with the orthonormality constraints  $\langle \psi_i, \psi_j \rangle = \delta_{ij}$ .

The evaluation of the electron density is clearly the main bottleneck in the self-consistent iteration, which is the focus of the numerical algorithms for Kohn–Sham density functional theory. We consider efficient and accurate discretization for the evaluation of the electron density in this work.

## 2.2. Discontinuous Galerkin method for KSDFE

The discontinuous Galerkin (DG) methods have been developed for different types of partial differential equations [22–27]. One of the main advantages of the DG method is its flexibility in the choice of the approximation space, as the DG method does not require the continuity condition of the basis functions across the interfaces of the elements. This flexibility is important for constructing effective discretization schemes for Kohn–Sham density functional theory.

We present in the following a DG method for the evaluation of the electron density. Among the different formalisms in the DG framework, we will use the interior penalty method [22,24]. The interior penalty method naturally generalizes the variational principle (8).

We denote by  $\Omega$  the computational domain with the periodic boundary condition, which corresponds to  $\Gamma$  point sampling in the Brillouin zone.  $\Omega$  is also referred to as the global domain in the following discussion. Bloch boundary conditions can be taken into account as well, and this will appear in future publications. Let  $\mathcal{T}$  be a collection of quasi-uniform rectangular partitions of  $\Omega$  (see Fig. 2 for an example with four elements):

$$\mathcal{T} = \{E_1, E_2, \dots, E_M\} \tag{9}$$

and  $S$  be the collection of surfaces that correspond to  $\mathcal{T}$ . Each  $E_k$  is called an element of  $\Omega$ . For a typical choice of partitions used in practice, the elements are chosen to be of the same size. For example, for a crystalline material, elements can be chosen as integer multiples of the conventional cell of the underlying lattice. As a result, unlike the usual finite element analysis, the element size will remain the same.<sup>1</sup>

In the following discussion, we use extensively the inner products defined as below

$$\langle v, w \rangle_E = \int_E v^*(x)w(x) dx, \tag{10}$$

$$\langle \mathbf{v}, \mathbf{w} \rangle_S = \int_S \mathbf{v}^*(x) \cdot \mathbf{w}(x) ds(x), \tag{11}$$

$$\langle v, w \rangle_{\mathcal{T}} = \sum_{i=1}^M \langle v, w \rangle_{E_i}, \tag{12}$$

$$\langle \mathbf{v}, \mathbf{w} \rangle_S = \sum_{S \in \mathcal{S}} \langle \mathbf{v}, \mathbf{w} \rangle_S. \tag{13}$$

In the discontinuous Galerkin method (the interior penalty method), the discrete energy functional corresponding to (8) is given by

$$E_{\text{DG}}(\{\psi_i\}) = \frac{1}{2} \sum_{i=1}^N \langle \nabla \psi_i, \nabla \psi_i \rangle_{\mathcal{T}} - \sum_{i=1}^N \langle \{\{\nabla \psi_i\}\}, [\psi_i] \rangle_S + \langle V_{\text{eff}}, \rho \rangle_{\mathcal{T}} + \frac{\alpha}{h} \sum_{i=1}^N \langle [\psi_i], [\psi_i] \rangle_S + \sum_{\ell} \gamma_{\ell} \sum_{i=1}^N |\langle \mathbf{b}_{\ell}, \psi_i \rangle_{\mathcal{T}}|^2. \tag{14}$$

Here the last term comes from the non-local terms in Eq. (8), and  $\{\{\cdot\}\}$  and  $[\cdot]$  are the average and the jump operators across surfaces, defined as follows. For  $S \in \mathcal{S}^{\circ}$  the set of interior surfaces, we assume  $S$  is shared by elements  $K_1$  and  $K_2$ . Denote by  $n_1$  and  $n_2$  the unit normal vectors on  $S$  pointing exterior to  $K_1$  and  $K_2$ , respectively. With  $u_i = u|_{\partial K_i}$ ,  $i = 1, 2$ , we set

$$[[u]] = u_1 n_1 + u_2 n_2 \quad \text{on } S. \tag{15}$$

For  $S \in \mathcal{S}^{\partial}$  where  $\mathcal{S}^{\partial}$  is the union of the surfaces on the boundary, we set

$$[[u]] = un \quad \text{on } S, \tag{16}$$

where  $n$  is the outward unit normal. For vector-valued function  $q$ , we define

$$\{\{q\}\} = \frac{1}{2}(q_1 + q_2) \quad \text{on } S \in \mathcal{S}^{\circ}, \tag{17}$$

where  $q_i = q|_{\partial K_i}$ , and

$$\{\{q\}\} = q \quad \text{on } S \in \mathcal{S}^{\partial}. \tag{18}$$

Note that in the current context  $\mathcal{S} = \mathcal{S}^{\circ}$  since we assume periodic boundary condition for the computational domain, and every surface is an interior surface. The constant  $\alpha$  in (14) is a positive penalty parameter, which penalizes the jumps of functions across element surfaces to guarantee stability. The choice of  $\alpha$  will be further discussed in Section 5.

Assume that we have chosen for each element  $E_k$  a set of basis functions  $\{\varphi_{k,j}\}_{j=1}^{J_k}$ , where  $J_k$  is the number of basis functions in  $E_k$ . We extend each  $\varphi_{k,j}$  to the whole computational domain  $\Omega$  by setting it to be 0 on the complement set of  $E_k$ . Define the function space  $\mathcal{V}$  as

$$\mathcal{V} = \text{span}\{\varphi_{k,j}, E_k \in \mathcal{T}, j = 1, \dots, J_k\}. \tag{19}$$

We minimize (14) for  $\{\psi_i\} \subset \mathcal{V}$ . The energy functional (14) in the approximation space  $\mathcal{V}$  leads to the following eigenvalue problem for  $\{\psi_i\}_{i=1}^N$ . For any  $v \in \mathcal{V}$ ,

$$\frac{1}{2} \langle \nabla v, \nabla \psi_i \rangle_{\mathcal{T}} - \frac{1}{2} \langle [[v]], \{\{\nabla \psi_i\}\} \rangle_S - \frac{1}{2} \langle \{\{\nabla v\}\}, [\psi_i] \rangle_S + \frac{\alpha}{h} \langle [[v]], [\psi_i] \rangle_S + \langle v, V_{\text{eff}} \psi_i \rangle_{\mathcal{T}} + \sum_{\ell} \gamma_{\ell} \langle v, \mathbf{b}_{\ell} \rangle_{\mathcal{T}} \langle \mathbf{b}_{\ell}, \psi_i \rangle_{\mathcal{T}} = \lambda_i \langle v, \psi_i \rangle_{\mathcal{T}}. \tag{20}$$

<sup>1</sup> In the language of finite element method, we will not use the  $h$ -refinement.

Setting  $v = \varphi_{k'j}$  and

$$\psi_i = \sum_{E_k \in \mathcal{T}} \sum_{j=1}^{J_k} c_{i;kj} \varphi_{kj}, \quad (21)$$

we arrive at the following linear system

$$\sum_{kj} \left( \frac{1}{2} \langle \nabla \varphi_{k'j}, \nabla \varphi_{kj} \rangle_{\mathcal{T}} - \frac{1}{2} \langle \llbracket \varphi_{k'j} \rrbracket, \{ \{ \nabla \varphi_{kj} \} \} \rangle_S - \frac{1}{2} \langle \{ \{ \nabla \varphi_{k'j} \} \}, \llbracket \varphi_{kj} \rrbracket \rangle_S + \frac{\alpha}{\hbar} \langle \llbracket \varphi_{k'j} \rrbracket, \llbracket \varphi_{kj} \rrbracket \rangle_S + \langle \varphi_{k'j}, V_{\text{eff}} \varphi_{kj} \rangle_{\mathcal{T}} + \sum_{\ell} \gamma_{\ell} \langle \varphi_{k'j}, \mathbf{b}_{\ell} \rangle_{\mathcal{T}} \langle \mathbf{b}_{\ell}, \varphi_{kj} \rangle_{\mathcal{T}} \right) c_{i;kj} = \lambda_i \sum_{kj} \langle \varphi_{k'j}, \varphi_{kj} \rangle c_{i;kj}. \quad (22)$$

We define  $A$  to be the matrix with entries given by the expression in the parentheses,  $B$  to be the matrix with entries  $\langle \varphi_{k'j}, \varphi_{kj} \rangle$ , and  $c_i$  to be the vector with components  $(c_{i;kj})_{kj}$ , we have the following simple form of generalized eigenvalue problem

$$Ac_i = \lambda_i Bc_i$$

for  $i = 1, 2, \dots, N$ . Following the standard terminologies in the finite element method, we call  $A$  the (DG) stiffness matrix, and  $B$  the (DG) mass matrix. In the special case when the DG mass matrix  $B$  is equal to the identity matrix, we have a standard eigenvalue problem  $Ac_i = \lambda_i c_i$ . Once  $\{c_i\}$  are available, the electron density is calculated by

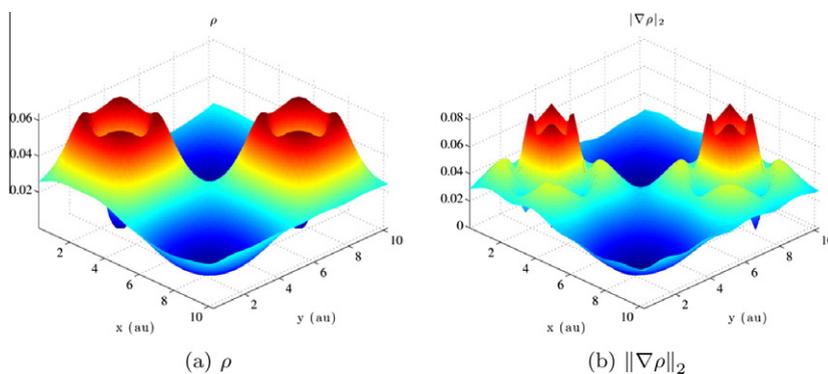
$$\tilde{\rho} = \sum_{i=1}^N \sum_{E_k \in \mathcal{T}} \left| \sum_{j=1}^{J_k} c_{i;kj} \varphi_{kj} \right|^2. \quad (23)$$

### 3. Basis functions adapted to the local environment

The proposed framework in the last section is valid for any choice of basis functions. To improve the efficiency of the algorithm, it is desirable to use less number of basis functions while maintaining the same accuracy. In order to achieve this goal, the choice of the functions  $\{\varphi_{kj}\}$  is important. In this section, we discuss a way to construct the basis functions  $\{\varphi_{kj}\}$  that are adapted to the local environment.

The starting point is the following observation. The Kohn–Sham orbitals  $\{\psi_i\}$  exhibit oscillatory behavior around the nuclei. In a full electron calculation, the nuclei charge density is the summation of delta functions located at the positions of the nuclei (or numerical delta function after discretization) and the Kohn–Sham orbitals have cusp points at the positions of the atoms. In the pseudopotential framework which involves only valence electrons, one can still see that the Kohn–Sham orbitals and the electron density are much more oscillatory near the atom cores than in the interstitial region, as illustrated in Fig. 1. In the setting of real space method or planewave method, in order to resolve the Kohn–Sham orbitals around the atom cores where the derivatives of Kohn–Sham orbitals become large one has to use a uniform fine mesh. Therefore, the number of mesh points becomes huge even for a small system. This makes the electronic structure calculation expensive.

In order to reduce the cost, we note that the Kohn–Sham orbitals are smooth away from the atoms and the uniform fine discretization is not efficient enough. Adaptive refinement techniques can be used to improve the efficiency by reducing the number of basis functions per atoms. Techniques of this type include finite element based adaptive mesh refinement method [41], finite volume based adaptive mesh refinement method, and multiresolution basis functions [42–44], to name a few. Our



**Fig. 1.** The electron density (a) and the norm of the gradient of the electron density (b) on a (001) slice of a mono-crystalline silicon system passing through two Si atoms. The two Si atoms are located at (2.57, 2.57) au and at (7.70, 7.70) au in this plane, respectively. Even in the pseudopotential framework, the electron density is much more oscillatory around the nuclei of the Si atoms and is smooth in the interstitial region.

approach builds the oscillatory behavior the Kohn–Sham orbitals near the atom cores into the basis functions. Hence, a small number of basis functions are enough to characterize the Kohn–Sham orbitals. This idea is not entirely new. For example, the philosophy of pseudopotential techniques is quite similar, though the reduction is done at the analytic level. On the side of numerical methods, the ideas behind atomic orbital basis or numerical atomic basis are closely related [12,17].

The main difference from the previous approaches is that instead of predetermining basis functions based on the information from isolated atoms, our approach builds the information from the local environment into the basis functions as well. Thanks to the flexibility of the discontinuous Galerkin framework, this can be done in a seamless and systematic way. The basis functions form a complete basis set in the global domain  $\Omega$ . The basis set is therefore efficient, and at the same time the accuracy can be improved systematically. This is an important difference between this approach and the previous methods along the same line.

The basis functions  $\{\varphi_{k,j}\}$  are determined as follows. Given the partition  $\mathcal{T}$  and the effective potential  $V_{\text{eff}}$ , let us focus on the construction of  $\{\varphi_{k,j}\}, j = 1, \dots, J_k$  for one element  $E_k \in \mathcal{T}$ . As discussed above, our approach is to adapt  $\{\varphi_{k,j}\}$  to the local environment in  $E_k$ .

For each element  $E_k$ , we take a region  $Q_k \supset E_k$ .  $Q_k$  is called the extended element associated with the element  $E_k$ . The set  $Q_k \setminus E_k$  is called the buffer area. We will choose  $Q_k$  which extends symmetrically along the  $\pm x(y, z)$  directions from the boundary of  $E_k$ . The length of the buffer area extended beyond the boundary of  $E_k$  along the  $\pm x(y, z)$  direction is called the “buffer size along the  $x(y, z)$  direction”. We restrict the effective Hamiltonian on  $Q_k$  by assuming the periodic boundary condition on  $\partial Q_k$  and denote by  $H_{\text{eff}, Q_k}$  the restricted Hamiltonian.  $H_{\text{eff}, Q_k}$  is discretized and diagonalized, and the corresponding eigenfunctions are denoted by  $\{\tilde{\varphi}_{k,j}\}$ , indexed in increasing order of the associated eigenvalues. We restrict the first  $J_k$  eigenfunctions  $\{\tilde{\varphi}_{k,j}\}$  from  $Q_k$  to  $E_k$ , denoted by  $\{\varphi_{k,j}\}$ . Each  $\varphi_{k,j}$  is therefore defined locally on  $E_k$ . As discussed before we extend each  $\varphi_{k,j}$  to the global domain  $\Omega$  by setting the value to be 0 on the complement of  $E_k$ . The resulting functions, still denoted by  $\{\varphi_{k,j}\}$  are called the adaptive local basis functions. Numerical result suggests that we can take very small  $J_k$  to achieve chemical accuracy.

The reason why we choose the periodic boundary condition on  $Q_k$  for the restriction  $H_{\text{eff}, Q_k}$  is twofold. On one hand, the periodic boundary condition captures better the bulk behavior of the system (than the Dirichlet boundary condition for example); On the other hand, the periodic boundary condition makes the solution of  $H_{\text{eff}, Q_k}$  more easily adapted to existing DFT algorithms and packages, as most of them can treat periodic boundary conditions. Other choices such as the Neumann boundary condition are possible, and the optimal choice of boundary conditions remains to be an open question.

The basis functions constructed from the buffer region capture well the local singular behavior of Kohn–Sham orbitals near the nuclei. Hence, the approximation space formed by  $\{\varphi_{k,j}\}$  gives an efficient and accurate discretization to the problem, as will be illustrated by numerical examples in Section 5. Note that the  $\{\tilde{\varphi}_{k,j}\}$ ’s are the eigenfunctions of the self-adjoint operator  $H_{\text{eff}, Q_k}$  on  $Q_k$ , and therefore form a complete basis set on  $Q_k$  when  $J_k \rightarrow \infty$ . This implies that after restriction, the functions  $\{\varphi_{k,j}\}$  also form a complete basis set on  $E_k$  as  $J_k \rightarrow \infty$ . The accuracy can therefore be systematically improved in the electronic structure calculation.

Eq. (22) proposes a generalized eigenvalue problem. From numerical point of view it would be more efficient if we can choose  $\{\varphi_{k,j}\}$  such that the DG mass matrix is an identity matrix and that Eq. (22) becomes a standard eigenvalue problem. Moreover, as  $J_k$  increases, the basis functions  $\{\varphi_{k,j}\}$  can become degenerate or nearly degenerate, which increases the condition number. Both problems can be solved at the same time by applying a singular value decomposition (SVD) filtering step, resulting in an orthonormal basis set  $\{\varphi_{k,j}\}$ :

1. For each  $k$ , form a matrix  $M_k = (\varphi_{k,1}, \varphi_{k,2}, \dots, \varphi_{k,J_k})$  with  $\varphi_{k,j}$ .
2. Calculate SVD decomposition  $U D V^* = M_k$ ,

$$D = \text{diag}(\lambda_{k,1}, \lambda_{k,2}, \dots, \lambda_{k,J_k}),$$

where  $\lambda_{k,j}$  are singular values of  $M_k$  ordered decreasingly in magnitude.

3. For a threshold  $\delta$ , find  $\tilde{J}_k$  such that  $|\lambda_{k,\tilde{J}_k}| > \delta$  and  $|\lambda_{k,\tilde{J}_k+1}| < \delta$  ( $\tilde{J}_k = J_k$  if all singular values are larger than the threshold). Take  $U_j$  be the  $j$ th column of  $U$ ,  $j = 1, \dots, \tilde{J}_k$ .
4. Set  $J_k \leftarrow \tilde{J}_k$  and  $\varphi_{k,j} \leftarrow U_{k,j}$  for  $j = 1, \dots, J_k$ .

**Remark.** Although the threshold  $\delta$  can avoid numerical degeneracy of the basis functions, the numerical degeneracy is not observed for the cases studied in Section 5. In other words, we will take  $\delta = 0$ ,  $J_k = \tilde{J}_k$ .

After constructing the basis functions  $\{\varphi_{k,j}\}$ , we then apply the discontinuous Galerkin framework to solve  $\{\psi_i\}$  and hence  $\rho$  corresponding to  $H_{\text{eff}}$ . We summarize the overall algorithm as follows:

1. Set  $n = 0$ , let  $\mathcal{T}$  be a partition of  $\Omega$  into elements, and  $\rho_0$  be an initial trial electron density.
2. Form the effective potential  $V_{\text{eff}}[\rho_n]$  and the effective Hamiltonian  $H_{\text{eff}}[\rho_n]$ .
3. For each element  $E_k \in \mathcal{T}$ , calculate the eigenfunctions corresponding to the Hamiltonian  $H_{\text{eff}, Q_k}$  on the extended element  $Q_k$ , and obtain the orthonormal adaptive local basis functions  $\{\varphi_{k,j}\}$ .
4. Solve (22) to obtain the coefficients  $\{c_{i,k,j}\}$  for the Kohn–Sham orbitals and reconstruct the electron density  $\tilde{\rho}$  by (23).
5. Mixing step: determine  $\rho_{n+1}$  from  $\rho_n$  and  $\tilde{\rho}$ . If  $\|\rho_n - \tilde{\rho}\| \leq \delta$ , stop; otherwise, go to step (2) with  $n \leftarrow n + 1$ .

We remark that due to the flexibility of the DG framework one can supplement the functions  $\{\varphi_{k,j}\}$  constructed above by other functions in  $E_k$ , such as local polynomials in  $E_k$ , Gaussian functions restricted to  $E_k$ , and other effective basis functions based on physical and chemical intuition. From practical point of view, we find that the adaptive basis set constructed above already achieves satisfactory performance.

#### 4. Implementation details

This section explains the implementation details for the above algorithm. Specialists of the DG methods can skip this section and go directly to the numerical results in Section 5. This section is mostly written for the readers who are less familiar with the DG implementation.

##### 4.1. Grids and interpolation

The above algorithm involves three types of domains: the global domain  $\Omega$ , the extended elements  $\{Q_k\}$ , and the elements  $\{E_k\}$ . Quantities defined on these domains are discretized with different types of grids.

- On  $\Omega$ , the quantities such as  $\rho$  and  $V_{\text{eff}}$  are discretized with a uniform Cartesian grid with a spacing fine enough to capture the singularities and oscillations in these quantities.
- The grid on  $Q_k$  is simply the restriction of the uniform grid of  $\Omega$  on  $Q_k$ . This is due to the consideration that all quantities on  $Q_k$  are treated as periodic and hence a uniform grid is the natural choice.
- The grid on  $E_k$  is a three-dimensional Cartesian Legendre–Gauss–Lobatto (LGL) grid in order to accurately carry out the operations of the basis functions  $\{\varphi_{k,j}\}$  such as numerical integration and trace operator for each element  $E_k$ .

Transferring various quantities between these three grids requires the following interpolation operators.

- $\Omega$  to  $Q_k$ . This is used when we restrict the density  $\rho_n$  and the effective potential  $V_{\text{eff}}$  to the extended element  $Q_k$ . Since the grid on  $Q_k$  is the restriction of the grid on  $\Omega$ , this interpolation operator simply copies the required values.
- $Q_k$  to  $E_k$ . This is used when one restricts  $\{\tilde{\varphi}_{k,j}\}$  and their derivatives to  $E_k$ . As the grid on  $Q_k$  is uniform, the interpolation is done by Fourier transform. Due to the fact that both grids are Cartesian, the interpolation can be carried out dimension by dimension, which greatly improves the efficiency.
- $E_k$  to  $\Omega$ . This is used when one assembles the Kohn–Sham orbitals  $\{\psi_i\}$  from the coefficients  $\{c_{i,k,j}\}$  of the elements. The interpolation from the LGL grid to the uniform grid is done by Lagrange interpolation, again carried out dimension by dimension. Averaging is performed for the grid points of  $\Omega$  shared by multiple elements.

The non-local pseudopotentials are used both in solving  $\{\tilde{\varphi}_{k,j}\}$  on each  $Q_k$  and in the numerical integration step on the LGL grid of each  $E_k$ . In our implementation, the non-local pseudopotentials are directly generated in real space on  $Q_k$  and on  $E_k$  without further interpolation between the grids.

##### 4.2. Implementation of the discontinuous Galerkin method

We use planewaves in each extended element  $Q_k$  to discretize the local effective Hamiltonian  $H_{\text{eff},Q_k}$  and the LOBPCG algorithm [45] with the preconditioner proposed in [46] to diagonalize the discretized Hamiltonian. The resulting eigenfunctions  $\{\tilde{\varphi}_{k,j}\}_{j=1}^k$  of  $H_{\text{eff},Q_k}$  are restricted to  $E_k$  and interpolated onto its LGL grid. Within the SVD filtering step, the inner product that we adopt is the discrete weighted  $\ell_2$  product with the LGL weights inside  $E_k$ . The main advantage of the SVD filtering step is that the discontinuous Galerkin method results in a standard eigenvalue problem.

The assembly of the DG stiffness matrix follows (22) and consists of the following steps.

- For the first term  $\frac{1}{2}\langle\nabla\varphi_{k',j},\nabla\varphi_{k,j}\rangle_{\mathcal{T}}$  and the fifth term  $\langle\varphi_{k',j},V_{\text{eff}}\varphi_{k,j}\rangle_{\mathcal{T}}$ , their contributions are non-zero only when  $k=k'$  since otherwise two basis functions have disjoint support. Hence, for each fixed  $k$ , we compute  $\langle\nabla\varphi_{k,j},\nabla\varphi_{k,j}\rangle_{E_k}$  and  $\langle\varphi_{k,j},V_{\text{eff}}\varphi_{k,j}\rangle_{E_k}$ . The integration is done numerically using the LGL grid on  $E_k$ . Part of the stiffness matrix corresponding to these two terms clearly has a block diagonal form.
- For the second, third, and fourth terms of (22), one needs to restrict basis functions and their derivatives to element faces. As the one-dimensional LGL grid contains the endpoints of its defining interval, this is done simply by restricting the values of the three-dimensional LGL grid to the element faces. One then calculates these three terms using numerical integration on these resulting two-dimensional LGL grids. Since the integral is non-zero only when  $E_k$  and  $E_{k'}$  are the same element or share a common face, part of the stiffness matrix corresponding to these three terms is again sparse.
- The last term of (22) is  $\sum_{\ell}\gamma_{\ell}\langle\varphi_{k',j},b_{\ell}\rangle_{\mathcal{T}}\langle b_{\ell},\varphi_{k,j}\rangle_{\mathcal{T}}$ . The integration is again approximated using the LGL grids of the elements. Notice that the contribution is non-zero iff  $\varphi_{k',j}$  and  $\varphi_{k,j}$  overlap with the support of a common  $b_{\ell}$ . Since each  $b_{\ell}$  is localized around a fixed atom,  $\varphi_{k,j}$  and  $\varphi_{k',j}$  need to be sufficiently close for this term to be non-zero. As a result, part of the stiffness matrix corresponding to this last term is also sparse.

Though the DG stiffness matrix  $A$  is sparse, this property is not yet exploited in the current implementation. The eigenvalues and eigenvectors of the DG stiffness matrix are calculated using the `pdsyevd` routine of ScaLAPACK by treating it as a dense matrix. We plan to replace it with more sophisticated solvers that leverage the sparsity of  $A$  in future.

#### 4.3. Parallelization

Our algorithm is implemented fully in parallel for message-passing environment. To simplify the discussion, we assume that the number of processors is equal to the number of elements. It is then convenient to index the processors  $\{P_k\}$  with the same index  $k$  used for the elements. In the more general setting where the number of elements is larger than the number of processors, each processor takes a couple of elements and the following discussion will apply with only minor modification. Each processor  $P_k$  locally stores the basis functions  $\{\varphi_{kj}\}$  for  $j = 1, 2, \dots, J_k$  and the unknowns  $\{c_{i;k,j}\}$  for  $i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, J_k$ . We further partition the non-local pseudopotentials  $\{b_\ell\}$  by assigning  $b_\ell$  to the processor  $P_k$  if and only if the atom associated to  $b_\ell$  is located in the element  $E_k$ .

The eigenfunctions of the local Hamiltonian  $H_{\text{eff},Q_k}$  are calculated on each processor  $P_k$ . In order to build the local Hamiltonian  $H_{\text{eff},Q_k}$ , the processor  $P_k$  needs to access all the non-local pseudopotentials of which the associated atoms are located in  $Q_k$ . This can be achieved by communication among  $E_k$  and its nearby elements. Once these pseudopotentials are available locally, the eigenfunctions of  $H_{\text{eff},Q_k}$  are computed in parallel without any extra communication between the processors.

The parallel implementation of the DG solve is more complicated:

- For the calculation of the first and the fifth terms of the DG stiffness matrix  $A$  in Eq. (22), each processor  $P_k$  performs numerical integration on  $E_k$ . Since the local basis functions  $\{\varphi_{kj}\}$  are only non-zero on  $E_k$ , this step is carried out fully in parallel.
- To calculate the second, third, and fourth terms, each processor  $P_k$  computes the surface integrals restricted to the *left*, *front*, and *bottom* faces of  $E_k$ . This requires the basis functions of the left, front, and bottom neighboring elements.
- To calculate the sixth term, each processor  $P_k$  computes the parts associated with the non-local pseudopotentials  $\{b_\ell\}$  located on  $P_k$ . This requires the access to the basis functions of all elements that overlap with  $b_\ell$ .

To summarize, each processor  $P_k$  needs to access the basis functions from its neighboring elements and from the elements that overlap with the support set of the non-local pseudopotentials located on the elements associated with  $P_k$ . Due to the locality of the non-local pseudopotentials, these elements are geometrically close to  $P_k$ . Since the size of the elements is generally equal to or larger than one unit cell, the support set of the non-local pseudopotentials are generally within the range of the neighboring elements. Therefore, the number of the non-local basis functions required by  $P_k$  is bounded by a small constant times the typical number of the basis functions in an element.

The use of the `pdsyevd` routine of ScaLAPACK for solving the eigenvalue problem (22) results in another source of communication. ScaLAPACK requires  $A$  to be stored in its block cyclic form and this form is quite different from the distribution in which the DG stiffness matrix is assembled (as mentioned above). As a result, one needs to redistribute  $A$  into this block cyclic form before calling `pdsyevd` and then redistribute the eigenfunctions afterwards.

In order to support these two sources of data communication, we have implemented a rather general communication framework that only requires the programmer to specify the desired non-local data. This framework then automatically fetches the data from the processors that store them locally. The actual communication is mostly done using asynchronous communication routines `MPI_Isend` and `MPI_Irecv`.

## 5. Numerical examples

In order to illustrate how our method works in practice, we present numerical results for the ground state electronic structure calculation, using sodium (Na) and silicon (Si) as representative examples for metallic and insulating systems, respectively. We find that very high accuracy (less than  $10^{-6}$  au per atom) can be achieved by using only a small number of adaptive local basis functions. Because of the small number of basis functions per atom, the DG scheme already exhibits significant speedup in computational time for a small system containing 128 Na atoms. We demonstrate that the current implementation is able to solve systems with thousands of atoms, and that the algorithm has a potential to be applied to much larger systems with a more advanced implementation.

This section is organized as follows: Section 5.1 introduces the setup of the test systems and how the error is quantified. Section 5.2 applies the adaptive local basis functions to disordered quasi-1D sodium and silicon system, followed by the result for the disordered quasi-2D and bulk 3D systems in Section 5.3. We discuss the effect of the penalty parameter  $\alpha$  in Section 5.4. Finally we demonstrate the computational performance of our parallel implementation of the adaptive local basis functions in Section 5.5.

### 5.1. Setup

We use the local density approximation (LDA) [36,37] for the exchange–correlation functional, and Hartwigsen–Goedecker–Hutter (HGH) pseudopotential [47] with the local and non-local pseudopotential fully implemented in the real space

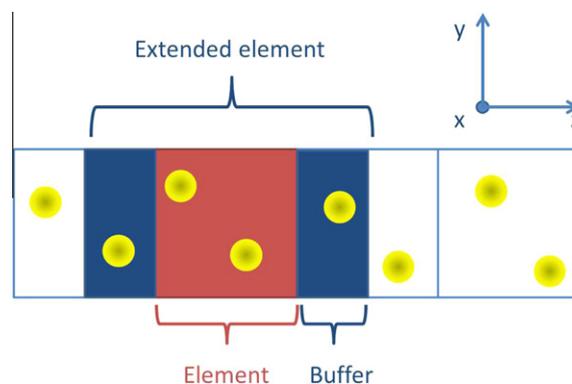
[48]. All quantities are reported in atomic units (au). All calculations are carried out on the Hopper system maintained at National Energy Research Scientific Computing Center (NERSC). Each compute node on Hopper has 24 processors (cores) with 32 gigabyte (GB) of memory (1.33 GB per core).

The performance of the adaptive local basis functions are tested using Na and Si as representative examples for simple metallic and insulating systems, respectively. The crystalline Na has a body centered cubic (bcc) unit cell, with 2 atoms per cell and a lattice constant of 7.994 au. The crystalline Si has a diamond cubic unit cell, with 8 atoms per cell and a lattice constant of 10.261 au. Each atomic configuration in the following tests is obtained by forming a supercell consisting  $m \times n \times p$  unit cells with perfect crystal structure, and a random displacement uniformly distributed in  $[-0.2, 0.2]$  au is then applied to each Cartesian coordinate of each atom in the supercell. The resulting atomic configuration is therefore mildly disordered in order to avoid the possible cancellation of errors for the case of perfect crystalline systems. A system is called quasi-1D if  $1 = m = n < p$ , quasi-2D if  $1 = m < n = p$ , and 3D bulk if  $1 < m = n = p$ , respectively. In all the tests below, the element is chosen to be the (conventional) unit cell of the lattice. Fig. 2 shows how a quasi-1D Na system with 8 atoms extended along the  $z$  direction are partitioned in order to generate adaptive local basis functions. The global domain is partitioned into 4 elements  $\{E_k\}_{k=1}^4$  with 2 atoms per element. The red area represents one of the elements  $E_2$ , and the corresponding extended element  $Q_2$  consists of both the red area and the blue area (buffer). We recall that the buffer size along the  $x(y, z)$  direction refers to the length of the buffer area extended beyond the boundary of the element  $E_k$  along the  $x(y, z)$  direction. The unit of buffer size is the lattice constant for the perfect crystalline system. Fig. 2 shows the case with the buffer size of 0.50 along the  $z$  direction, and 0.0 along the  $x$  and  $y$  directions.

We quantify the error of the adaptive local basis functions the error of the total energy per atom which is defined as follows. First, the electronic structure problem is solved by using planewaves on the global domain starting from a random initial guess of the electron wavefunctions. The total energy after reaching self-consistency is denoted by  $E_{GLB}$ . Then, the same electronic structure problem is solved by the DG formulation starting from a random initial guess of the adaptive local basis functions on each element. The total energy after reaching self-consistency is denoted by  $E_{DG}$ . The global domain calculation and the DG calculation using adaptive local basis functions are therefore completely independent, and the error of the total energy per atom is defined to be  $|E_{GLB} - E_{DG}|/N_{atom}$ . For simplicity only  $\Gamma$  point is used in the Brillouin zone sampling. The proposed method can be easily generalized to  $k$ -point sampling. 10 LOBPCG iterations are used in each SCF iteration for the global domain calculation, and 3 LOBPCG iterations are used in each SCF iteration for generating the adaptive local basis functions in the DG calculation. A small number of LOBPCG iterations is already sufficient, since the electron wavefunctions in the global domain calculation and the adaptive local basis functions in the DG calculations at the end of each SCF iteration can be reused as the initial guess in the consequent SCF iteration for continuous refinement. Anderson mixing is used for the SCF iteration with a fictitious electron temperature set to be 2000 K to facilitate the convergence of the SCF iteration.

The grid spacing for the global domain calculation is 0.4 au for Na and 0.32 au for Si. This translates to a grid of size  $20 \times 20 \times 20$  to discretize one Na unit cell and a grid of size  $32 \times 32 \times 32$  to discretize one Si unit cell. The Legendre-Gauss-Lobatto (LGL) grid for each element is  $20 \times 20 \times 20$  for Na and  $40 \times 40 \times 40$  for Si. The LGL grid is only used for the purpose of numerical integration in the assembly process of the DG matrix. We remark that this grid is denser than what is commonly used for the electronic structure calculation for three reasons: (1) the GHG pseudopotential used in the present calculation is more stiff than many other pseudopotentials such as the Troullier–Martins pseudopotential [4]; (2) the potentials and wavefunctions are represented in the real space rather than in the Fourier space; (3) most importantly, a dense grid in the real space is needed in both global domain calculations and DG calculations in order to reliably reflect the error of the total energy per atom.

We remarked in the end of Section 3 that the DG framework is very flexible and can incorporate not only the adaptive local basis functions but also other basis functions such as local polynomials. In practice we find that the adaptive local basis



**Fig. 2.** A quasi-1D disordered Na system with 8 atoms extended along the  $z$  direction, viewed along the  $x$  direction. The length of each empty box is equal to the lattice constant for the perfect Na crystal. The red area represents one of the elements  $E_2$ . The corresponding extended element  $Q_2$  consists of both the red area and the blue area (buffer). The buffer size is 0.50 (in the unit of lattice constant) along the  $z$  direction, and is 0.0 along the  $x$  and  $y$  directions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

functions are computationally more efficient than polynomials. Therefore in the following discussion only adaptive local functions will be used in the basis set. The number of adaptive local functions per atom is also referred to as the degrees of freedom (DOF) per atom.

### 5.2. Disordered quasi-1D system

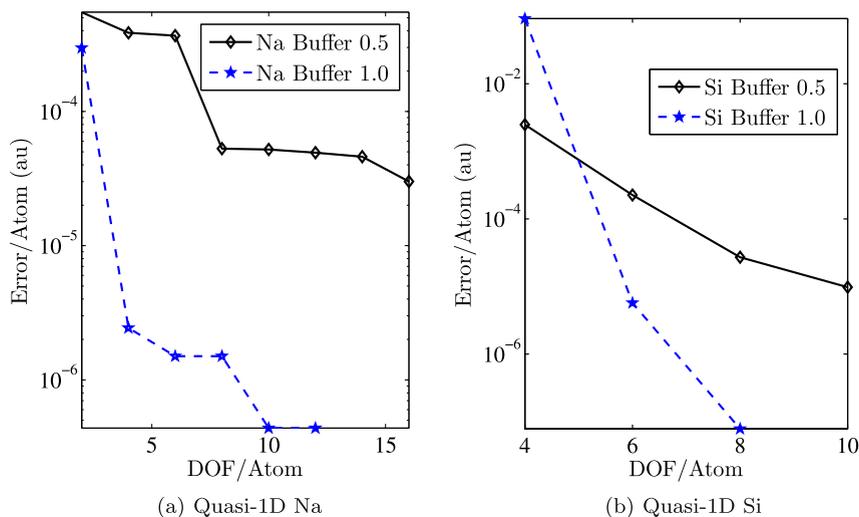
The error of the total energy per atom with respect to different buffer sizes and different numbers of basis functions per atom (DOF per atom) is illustrated for the disordered quasi-1D sodium system in Fig. 3(a) and for the disordered quasi-1D silicon system in Fig. 3(b). The penalty parameter  $\alpha$  is 20. In both cases, the error decreases systematically when the buffer size and the number of basis functions per atom increase. For Na, the error of the total energy per atom is already below  $10^{-3}$  au using as few as 4 basis functions per atom with a small buffer of size 0.50 (black diamond with solid line). When the buffer size is increased to 1.00 (blue star with dashed line), the error of the total energy per atom is  $4.3 \times 10^{-7}$  au or 0.01 meV using 10 adaptive local basis functions per atom.

Similar behavior is found for the silicon system. For a small buffer of size 0.50 (black diamond with solid line), the error of the total energy per atom is  $2.3 \times 10^{-4}$  au with 6 basis functions per atom. For the buffer of size 1.00 (blue star with dashed line), the error of the total energy per atom is  $7.8 \times 10^{-8}$  au or 0.002 meV using as few as 8 basis functions per atom. Physical intuition suggests that the minimum number of basis functions is 4, which reflects one  $2s$  and three  $2p$  atomic orbitals. 20–40 number of basis functions per atom is generally required to achieve good accuracy if Gaussian type orbitals or numerical atomic orbitals are to be used [17]. Therefore for the quasi-1D systems tested here, our algorithm achieves nearly the optimal performance in terms of the number of basis functions per atom.

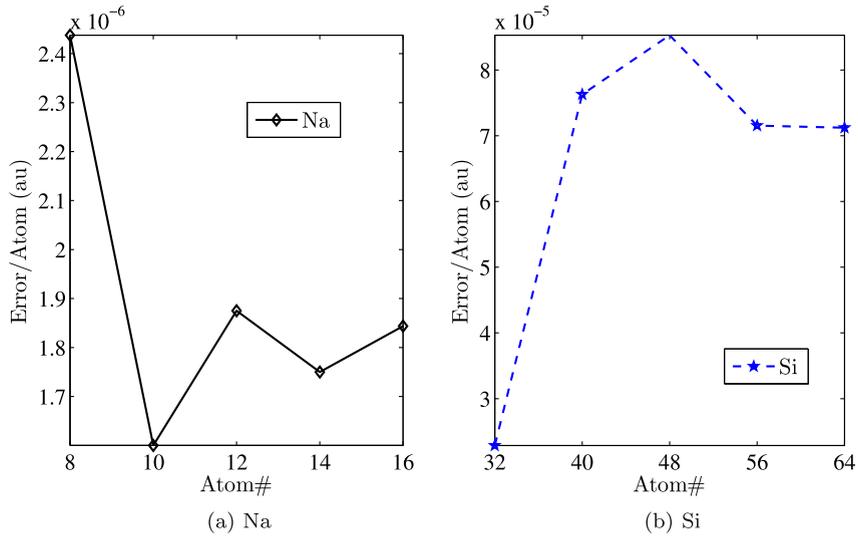
The behavior of the error found above depends weakly on the number of atoms of the quasi-1D system extended along the  $z$  direction. The error of the total energy per atom for disordered quasi-1D systems of different numbers of atoms is shown for Na in Fig. 4(a) and for Si in Fig. 4(b), respectively. In both cases the buffer size is 0.50, and the penalty parameter is 20. Here 4 and 6 adaptive local basis functions per atom are used for Na and Si, respectively.

### 5.3. Disordered quasi-2D and 3D bulk systems

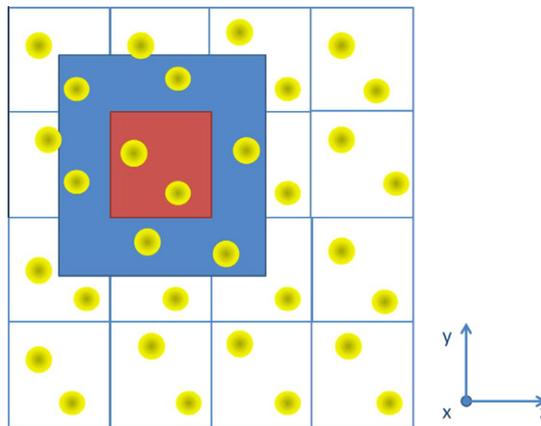
This section studies the relation between the error of the total energy per atom and the dimensionality of the system. The partition of the domain for systems of higher dimension is similar to that in the quasi-1D case. Fig. 5 shows the partition of a quasi-2D system with 32 sodium atoms, viewed along the  $x$  direction. The domain is partitioned into 16 disjoint elements. The length of each element (red area) is equal to the length of the lattice constant of the crystalline unit cell. The corresponding extended element for solving the adaptive local basis functions includes both the element (red area) and the buffer (blue area). Fig. 6(a) shows the behavior of the error for a disordered quasi-2D sodium system containing 32 atoms with the buffer of size 0.50 (black diamond with solid line) and of size 1.00 (blue star with dashed line), respectively. For the case with the buffer size equal to 0.50, the error of the total energy per atom is  $1.0 \times 10^{-3}$  au using 8 basis functions per atom. The error of the total energy per atom can reach  $2.8 \times 10^{-6}$  au with 16 basis functions per atom and buffer size 1.00. Fig. 6(b) shows the behavior of the error for a disordered bulk 3D sodium system containing 128 atoms with the buffer of size 0.50 (black



**Fig. 3.** (a) The error of the total energy per atom (the y axis, plotted in log-scale) for a disordered quasi-1D sodium system consisting of 8 atoms, with respect to the number of adaptive local basis functions per atom (the x axis). The buffer sizes are chosen to be 0.50 (black diamond with solid line), and 1.00 (blue star with dashed line). (b) The error of the total energy per atom (the y axis, plotted in log-scale) for a disordered quasi-1D silicon system consisting of 32 atoms, with respect to the number of adaptive local basis functions per atom (the x axis). The legend is the same as in (a). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 4.** (a) The error of the total energy per atom (the y axis) for disordered quasi-1D sodium systems of different numbers of atoms (the x axis) extended along the z direction. The buffer size is 0.50, and 4 adaptive local basis functions per atom are used in each calculation. (b) The error of the total energy per atom for the disordered quasi-1D silicon systems of different numbers of atoms (the x axis) extended along the z direction. The buffer size is 0.50, and 6 adaptive local basis functions per atom are used in each calculation.

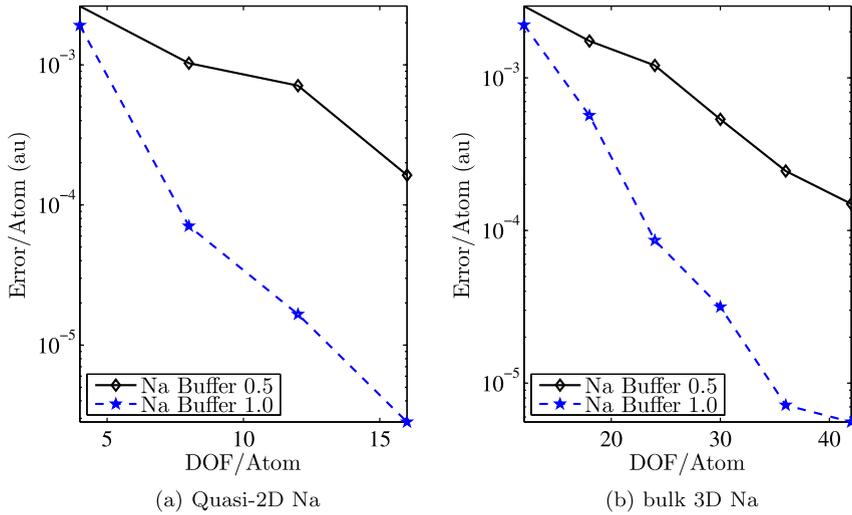


**Fig. 5.** A quasi-2D disordered Na system with 32 atoms extended along the y and the z directions, viewed along the x direction. The red area represents one of the elements  $E_{2,2}$ , and the corresponding extended element  $Q_{2,2}$  consists of both the red area and the blue area (buffer). The buffer size is 0.50 (in the unit of lattice constant) along the y and the z directions, and is 0.0 along the x direction. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

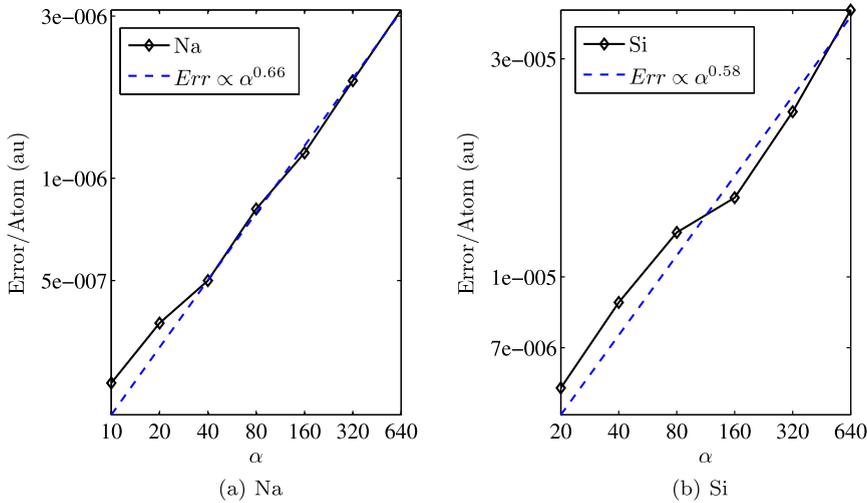
diamond with solid line) and of size 1.00 (blue star with dashed line), respectively. For the case with the buffer size equal to 0.50, the error of the total energy per atom is  $1.2 \times 10^{-3}$  au using 24 basis functions per atom. The error of the total energy per atom can reach  $5.6 \times 10^{-6}$  au or 0.15 meV with 42 basis functions per atom and buffer size 1.00. Compared to the quasi-1D case, the number of adaptive local basis functions per atom increases significantly in order to reach the same accuracy. The increasing number of basis functions is partly due to the increasing number of Na atoms in the extended element. In this case, the numbers of the Na atoms in the extended element with a buffer size of 1.00 are 4, 18, 54 for quasi-1D, quasi-2D and bulk 3D systems, respectively. The increased number of Na atoms in the extended elements requires more eigenfunctions in the extended elements, and therefore more adaptive local basis functions per atom in the elements.

#### 5.4. The penalty parameter

The interior penalty formulation of the discontinuous Galerkin method contains an important parameter  $\alpha$  to guarantee stability.  $\alpha = 20$  has been applied uniformly to all the examples studied so far. The  $\alpha$ -dependence of the error of the total energy per atom is shown for the quasi-1D sodium system in Fig. 7(a) and for the quasi-1D silicon system in Fig. 7(b), respectively. For Na, the buffer size is 1.00, and the number of basis functions per atom is 8. The error of the total energy per atom is



**Fig. 6.** (a) The error of the total energy per atom (the y axis, plotted in log-scale) for a disordered quasi-2D sodium system containing 32 atoms, with respect to the number of basis functions per atom (the x axis). The buffer size is chosen to be 0.50 (black diamond with solid line), and 1.00 (blue star with dashed line), respectively. (b) The error of the total energy per atom for a disordered bulk 3D sodium system (the y axis, plotted in log-scale) containing 128 atoms, with respect to the number of basis functions per atom (the x axis). The buffer size is chosen to be 0.50 (black diamond with solid line), and 1.00 (blue star with dashed line), respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 7.** (a) Log–log plot for the error of the total energy per atom (the y axis) with respect to the penalty parameter  $\alpha$  (the x axis), for a quasi-1D sodium system with 8 atoms. The buffer size is 1.00 and the number of basis functions per atom is 12. The error (black diamond with solid line) can be fitted with a polynomial function of  $\alpha$  (blue dashed line). (b) Log–log plot for the error of the total energy per atom (the y axis) with respect to the penalty parameter  $\alpha$  (the x axis), for a quasi-1D silicon system with 32 atoms. The buffer size is 1.00 and the number of basis functions per atom is 6. The error (black diamond with solid line) can be fitted with a polynomial function of  $\alpha$  (blue dashed line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

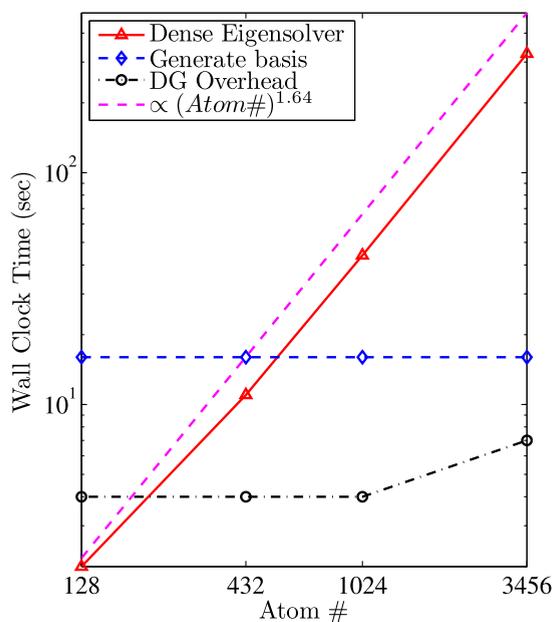
empirically proportional to  $\alpha^{0.66}$  up to  $\alpha = 640$ . For Si, the buffer size is 1.00, and the number of basis functions per atom is 6. The error of the total energy per atom is empirically proportional to  $\alpha^{0.58}$  up to  $\alpha = 640$ . We also remark that the DG formulation can become unstable for  $\alpha$  smaller than a certain threshold value. For example, the error of the total energy per atom is  $2.9 \times 10^{-1}$  au for Na with  $\alpha = 5$ , and the error of the total energy per atom is  $1.7 \times 10^{-2}$  au for Si with  $\alpha = 10$ . Therefore the penalty parameter  $\alpha$  plays an important role in the stability of the algorithm, but the DG scheme can be accurate and stable with respect to a large range of  $\alpha$  values.

5.5. Computational efficiency

The small number of the adaptive basis functions per atom can lead to significant savings of the computational time. We illustrate the efficiency of our algorithm using a disordered bulk 3D sodium system with the buffer size of 1.00 and with 16

basis functions per atom. Fig. 6(b) suggests that the error of the total energy per atom is about  $10^{-3}$  au for this choice of the parameters. The size of each element is equal to the lattice constant with 2 Na atoms in each element. The size of the global domain  $\Omega$  ranges from  $4 \times 4 \times 4$  unit cells with 128 Na atoms to  $12 \times 12 \times 12$  elements with 3456 atoms. The number of processors (cores) used is proportional to the number of elements, and 1728 processors are used in the problem with  $12 \times 12 \times 12$  elements. We compare the wall clock time for one step self consistent iteration with 3 LOBPCG iterations for solving the adaptive basis functions in the extended elements. Fig. 8 compares the wall clock time for solving the DG eigenvalue problem using ScaLAPACK function `pdsyevd` (red triangle with solid line), the time for generating the adaptive local basis functions in the extended elements using LOBPCG solver (blue diamond with dashed line), and the time for the overhead in the DG calculation (black circle with dot dashed line). The buffer size is 1.00, and the number of basis functions per atom is 16. Since both the size of the extended elements and the number of basis functions per atom are fixed, the computational time for solving the adaptive basis functions does not depend on the global domain size. The overhead in the DG calculation method includes mainly the assembly process of the DG Hamiltonian matrix via numerical integration and data communication. All numerical integrations are localized inside each element and its neighboring elements. Our implementation ensures that the data communication is restricted to be within nearest neighboring elements. Therefore the time for the overhead increases mildly with respect to the global system size. The complexity of the DG eigensolver using `pdsyevd` scales cubically with respect to global system size in the asymptotic limit, and starts to dominate the cost of computational time for system containing more than 1000 atoms. Since the number of processors is proportional to the number of elements, the asymptotic wall clock time for the DG eigensolver should scales quadratically with respect to the number of atoms. The practical wall clock time for solving the DG eigensolver is found to be proportional to  $(N_{atom})^{1.64}$  (magenta dashed line in Fig. 8), indicating that the asymptotic cubic scaling has not yet been reached. In the largest example with 3456 atoms, the matrix size of the DG Hamiltonian matrix is 55,296.

The efficiency due to the dimension reduction of the adaptive basis functions can be illustrated by the comparison between the cost of the computational time of the LOBPCG eigensolver in the global domain calculation (Global), and that of the DG eigenvalue problem with the adaptive basis functions (DG), as reported in Table 1. The global domain calculation uses 10 LOBPCG iteration steps per SCF iteration. On a single processor, the global domain calculation costs 806 s for the bulk 3D sodium system with 128 atoms, and 19,112 s for the bulk 3D sodium system with 432 atoms. By assuming that the global domain calculation can be ideally parallelized, the third column of Table 1 reports the computational time of the global domain calculation measured on a single processor divided by the number of processors used in the corresponding DG eigensolver. The fourth column reports the wall clock time for the DG eigensolver executed in parallel. We remark that the computational time for solving the adaptive local basis functions is not taken into account, since we are comparing the sav-



**Fig. 8.** Log-log plot for the wall clock time (y axis) for solving disordered bulk 3D sodium systems of different sizes (x axis) with one step self-consistent field iteration. The number of processors is chosen to be proportional to the number of atoms, with 1728 processors used for the largest problem solved here (3456 Na atoms). The total wall clock time is broken down into the time for solving the DG eigenvalue problem using ScaLAPACK function `pdsyevd` (red triangle with solid line), the time for generating the adaptive local basis functions in the extended elements using LOBPCG solver (blue diamond with dashed line), and the time for the overhead in the DG calculation, including the matrix assembly and data communication (black circle with dot dashed line). The buffer size is 1.00, and the number of basis functions per atom is 16. The scaling of the wall clock time for solving the DG eigenvalue problem using `pdsyevd` with respect to the number of atoms is illustrated by the magenta dashed line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 1**

The comparison of the cost of the wall clock time using the LOBPCG iteration on the global domain (performed with a single processor and divide the time by the number of processors in column 2, assuming that the LOBPCG are perfectly parallelized) and the wall clock time using the adaptive local basis functions (only count the DG eigenvalue solver using ScaLAPACK with the number of processors in column 2). The systems under study are the bulk 3D sodium system with  $4 \times 4 \times 4$  elements (128 Na atoms), and with  $6 \times 6 \times 6$  elements (432 Na atoms), respectively.

Atom #	Proc. #	Global (s)	DG (s)
128	64	13	1
432	216	88	14

ings of the computational time due to the dimension reduction of the basis functions. It is found that the saving of the computational time is already significant even when the system size is relatively small.

## 6. Discussion and conclusion

In this paper we proposed the adaptive local basis functions for discretizing the Kohn–Sham Hamiltonian operator, and demonstrated that the adaptive local basis functions are efficient for calculating the total energy and electron density, and can reach high accuracy with a very small number of basis functions per atom. The adaptive local basis functions are discontinuous in the global domain, and the continuous Kohn–Sham orbitals and electron density are reconstructed from these discontinuous basis functions using the discontinuous Galerkin (DG) framework. The environmental effect is automatically built into the basis functions, thanks to the flexibility provided by the DG framework.

The current implementation of the DG method is already able to perform the total energy calculation for systems consisting of thousands of atoms. The performance of the DG method can be improved by taking into account the block sparsity of the DG stiffness matrix. Furthermore, the local nature of the adaptive basis functions allows us to incorporate the recently developed pole expansion and selected inversion type fast algorithms [49–52] into the DG framework. The capability of the resulting algorithm is expected to be greatly enhanced compared to the current implementation. This is our ongoing work.

In order to generalize the current framework to the force calculation and further to the geometry optimization and the *ab initio* molecular dynamics simulation, the adaptive local basis functions and their derivatives with respect to the positions of the atoms (called Pulay force [53]) should be both accessible. Recently we propose the optimized local basis functions [54] that is able to systematically control the magnitude of the Pulay force, which is a further improvement of the adaptive local basis functions. This is also our ongoing work.

## Acknowledgement

This work is partially supported by DOE under Contract No. DE-FG02-03ER25587 and by ONR under Contract No. N00014-01-1-0674 (W. E and L. L.), and by a Sloan Research Fellowship and by NSF CAREER Grant DMS-0846501 (L. Y.). We thank the National Energy Research Scientific Computing Center (NERSC), and the Texas Advanced Computing Center (TACC) for the support to perform the calculations. L. L. and J. L. thank the University of Texas at Austin for the hospitality where the idea of this paper starts.

## References

- [1] P. Hohenberg, W. Kohn, Inhomogeneous electron gas, *Phys. Rev.* 136 (1964) B864–B871.
- [2] W. Kohn, L. Sham, Self-consistent equations including exchange and correlation effects, *Phys. Rev.* 140 (1965) A1133–A1138.
- [3] R. Martin, *Electronic structure: basic theory and practical methods*, Cambridge Univ. Press, Cambridge, 2004.
- [4] N. Troullier, J. Martins, Efficient pseudopotentials for plane-wave calculations, *Phys. Rev. B* 43 (1991) 1993–2006.
- [5] J. Chelikowsky, N. Troullier, Y. Saad, Finite-difference-pseudopotential method: electronic structure calculations without a basis, *Phys. Rev. Lett.* 72 (1994) 1240–1243.
- [6] M. Alemany, M. Jain, L. Kronik, J. Chelikowsky, Real-space pseudopotential method for computing the electronic properties of periodic systems, *Phys. Rev. B* 69 (2004) 075101.
- [7] E. Tsuchida, M. Tsukada, Electronic-structure calculations based on the finite-element method, *Phys. Rev. B* 52 (1995) 5573–5578.
- [8] J. Pask, B. Klein, C. Fong, P. Sterne, Real-space local polynomial basis for solid-state electronic-structure calculations: a finite-element approach, *Phys. Rev. B* 59 (1999) 12352–12358.
- [9] J. Pask, P. Sterne, Finite element methods in *ab initio* electronic structure calculations, *Modelling Simul. Mater. Sci. Eng.* 13 (2005) R71.
- [10] F. Averill, D. Ellis, An efficient numerical multicenter basis set for molecular orbital calculations: application to  $\text{FeCl}_4$ , *J. Chem. Phys.* 59 (1973) 6412–6418.
- [11] B. Delley, D. Ellis, Efficient and accurate expansion methods for molecules in local density models, *J. Chem. Phys.* 76 (1982) 1949–1960.
- [12] H. Eschrig, *Optimized LCAO Method and the Electronic Structure of Extended Systems*, Akademie Verlag and Springer, Berlin, 1988.
- [13] K. Koepf, H. Eschrig, Full-potential nonorthogonal local-orbital minimum-basis band-structure scheme, *Phys. Rev. B* 59 (1999) 1743–1757.
- [14] S. Kenny, A. Horsfield, H. Fujitani, Transferable atomic-type orbital basis sets for solids, *Phys. Rev. B* 62 (2000) 4899–4905.
- [15] J. Junquera, O. Paz, D. Sanchez-Portal, E. Artacho, Numerical atomic orbitals for linear-scaling calculations, *Phys. Rev. B* 64 (2001) 235111.
- [16] T. Ozaki, Variationally optimized atomic orbitals for large-scale electronic structures, *Phys. Rev. B* 67 (2003) 155108.
- [17] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, M. Scheffler, *Ab initio* molecular simulations with numeric atom-centered orbitals, *Comput. Phys. Commun.* 180 (2009) 2175–2196.
- [18] J.C. Slater, Wave functions in a periodic potential, *Phys. Rev.* 51 (1937) 846–851.

- [19] O.K. Andersen, Linear methods in band theory, *Phys. Rev. B* 12 (1975) 3060–3083.
- [20] P.E. Blöchl, Projector augmented-wave method, *Phys. Rev. B* 50 (1994) 17953–17979.
- [21] N. Sukumar, J. Pask, Classical and enriched finite element formulations for Bloch-periodic boundary conditions, *Int. J. Numer. Methods Eng.* 77 (2009) 1121–1138.
- [22] I. Babuška, M. Zlámal, Nonconforming elements in the finite element method with penalty, *SIAM J. Numer. Anal.* 10 (1973) 863–875.
- [23] M. Wheeler, An elliptic collocation-finite element method with interior penalties, *SIAM J. Numer. Anal.* 15 (1978) 152–161.
- [24] D.N. Arnold, An interior penalty finite element method with discontinuous elements, *SIAM J. Numer. Anal.* 19 (1982) 742–760.
- [25] B. Cockburn, G. Karniadakis, C.-W. Shu, *Discontinuous Galerkin methods: theory, computation and applications*, Lecture Notes in Computational Science and Engineering, vol. 11, Springer-Verlag, Berlin, 2000.
- [26] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, *J. Sci. Comput.* 16 (2001) 173–261.
- [27] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, *SIAM J. Numer. Anal.* 39 (2002) 1749.
- [28] Y. Maday, E. Rønquist, A reduced-basis element method, *J. Sci. Comput.* 17 (2002) 447–459.
- [29] Y. Maday, A. Patera, G. Turinici, Global a priori convergence theory for reduced-basis approximations of single-parameter symmetric coercive elliptic partial differential equations, *C.R. Acad. Sci. Paris. Ser. I* 335 (2002) 289–294.
- [30] E. Cancès, C. Le Bris, Y. Maday, N. Nguyen, A. Patera, G. Pau, Feasibility and competitiveness of a reduced basis approach for rapid electronic structure calculations in quantum chemistry, in: *High-Dimensional Partial Differential Equations in Science and Engineering*, CRM Proc. Lecture Notes, vol. 41, Amer. Math. Soc., Providence, RI, 2007, pp. 15–47.
- [31] Y. Chen, J. Hesthaven, Y. Maday, J. Rodríguez, Certified reduced basis methods and output bounds for the harmonic Maxwell's equations, *SIAM J. Sci. Comput.* 32 (2010) 970–996.
- [32] L. Yuan, C.-W. Shu, Discontinuous Galerkin method based on non-polynomial approximation spaces, *J. Comput. Phys.* 218 (2006) 295–323.
- [33] L. Yuan, C.-W. Shu, Discontinuous Galerkin method for a class of elliptic multi-scale problems, *Int. J. Numer. Methods Fluids* 56 (2007) 1017–1032.
- [34] W. Wang, J. Guzmán, C.-W. Shu, The multiscale discontinuous Galerkin method for solving a class of second order elliptic problems with rough coefficients, *Int. J. Numer. Anal. Model.* 8 (2011) 28–47.
- [35] L. Kleinman, D. Bylander, Efficacious form for model pseudopotentials, *Phys. Rev. Lett.* 48 (1982) 1425–1428.
- [36] D. Ceperley, B. Alder, Ground state of the electron gas by a stochastic method, *Phys. Rev. Lett.* 45 (1980) 566–569.
- [37] J. Perdew, A. Zunger, Self-interaction correction to density-functional approximations for many-electron systems, *Phys. Rev. B* 23 (1981) 5048–5079.
- [38] M. Payne, M. Teter, D. Allan, T. Arias, J. Joannopoulos, Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients, *Rev. Mod. Phys.* 64 (1992) 1045–1097.
- [39] D. Anderson, Iterative procedures for nonlinear integral equations, *J. Assoc. Comput. Mach.* 12 (1965) 547–560.
- [40] D.D. Johnson, Modified Broyden's method for accelerating convergence in self-consistent calculations, *Phys. Rev. B* 38 (1988) 12807–12813.
- [41] J. Fattebert, R. Hornung, A. Wissink, Finite element approach for density functional theory calculations on locally-refined meshes, *J. Comput. Phys.* 223 (2007) 759–773.
- [42] T. Arias, Multiresolution analysis of electronic structure: semicardinal and wavelet bases, *Rev. Mod. Phys.* 71 (1999) 267.
- [43] L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, R. Schneider, Daubechies wavelets as a basis set for density functional pseudopotential calculations, *J. Chem. Phys.* 129 (2008) 014109.
- [44] R. Harrison, G. Fann, T. Yanai, Z. Gan, G. Beylkin, Multiresolution quantum chemistry: basic theory and initial applications, *J. Chem. Phys.* 121 (2004) 11587.
- [45] A. Knyazev, Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method, *SIAM J. Sci. Comput.* 23 (2001) 517–541.
- [46] M. Teter, M. Payne, D. Allan, Solution of Schrödinger's equation for large systems, *Phys. Rev. B* 40 (1989) 12255–12263.
- [47] C. Hartwigsen, S. Goedecker, J. Hutter, Relativistic separable dual-space gaussian pseudopotentials from h to rn, *Phys. Rev. B* 58 (1998) 3641.
- [48] J. Pask, P. Sterne, Real-space formulation of the electrostatic potential and total energy of solids, *Phys. Rev. B* 71 (2005) 113101.
- [49] L. Lin, J. Lu, L. Ying, W. E, Pole-based approximation of the Fermi–Dirac function, *Chin. Ann. Math. Ser. B* 30 (2009) 729–742.
- [50] L. Lin, J. Lu, L. Ying, R. Car, W. E, Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems, *Commun. Math. Sci.* 7 (2009) 755–777.
- [51] L. Lin, C. Yang, J. Lu, L. Ying, W. E, A fast parallel algorithm for selected inversion of structured sparse matrices with application to 2D electronic structure calculations, *SIAM J. Sci. Comput.* 33 (2011) 1329.
- [52] L. Lin, C. Yang, J. Meza, J. Lu, L. Ying, W. E, Sellnv – an algorithm for selected inversion of a sparse symmetric matrix, *ACM. Trans. Math. Software* 37 (2010) 40.
- [53] P. Pulay, Ab initio calculation of force constants and equilibrium geometries in polyatomic molecules I. Theory, *Mol. Phys.* 17 (1969) 197–204.
- [54] L. Lin, J. Lu, L. Ying, W. E, Optimized local basis function for Kohn–Sham density functional theory, *J. Comput. Phys.* (2011), submitted for publication.