AN ORTHOGONALIZATION-FREE PARALLELIZABLE FRAMEWORK FOR ALL-ELECTRON CALCULATIONS IN DENSITY FUNCTIONAL THEORY[*]

BIN GAO[†], GUANGHUI HU[‡], YANG KUANG[§], AND XIN LIU[¶]

Abstract. All-electron calculations play an important role in density functional theory, in which improving computational efficiency is one of the most needed and challenging tasks. In the model formulations, both nonlinear eigenvalue problem and total energy minimization problem pursue orthogonal solutions. Most existing algorithms for solving these two models invoke orthogonalization process either explicitly or implicitly in each iteration. Their efficiency suffers from this process in view of its cubic complexity and low parallel scalability in terms of the number of electrons for large scale systems. To break through this bottleneck, we propose an orthogonalization-free algorithm framework based on the total energy minimization problem. It is shown that the desired orthogonality can be gradually achieved without invoking orthogonalization in each iteration. Moreover, this framework fully consists of Basic Linear Algebra Subprograms (BLAS) operations and thus can be naturally parallelized. The global convergence of the proposed algorithm is established. We also present a precondition technique which can dramatically accelerate the convergence of the algorithm. The numerical experiments on all-electron calculations show the efficiency and high scalability of the proposed algorithm.

Key words. density functional theory, all-electron calculations, orthogonalization-free, parallel algorithm

AMS subject classifications. 35Q55, 65N30, 90C06

1. Introduction. We aim to find the ground state solution of a molecular system from all-electron calculations. In view of Kohn–Sham density functional theory (KSDFT) [19], this can be achieved by solving the lowest p eigenpairs of the Kohn–Sham equation:

$$(1.1) \quad \begin{cases} \hat{H}\psi_l(\boldsymbol{r}) = \varepsilon_l \psi_l(\boldsymbol{r}), & l = 1, 2, \ldots, p, \\ \int_{\mathbb{R}^3} \psi_l \psi_{l'} d\boldsymbol{r} = \delta_{ll'}, & l, l' = 1, 2, \ldots, p, \end{cases}$$

where $\hat{H}$ is the Hamiltonian operator, $\psi_l(\boldsymbol{r})$ is the l-th wavefunction (eigenfunction), ε_l refers to the corresponding eigenenergy, $\delta_{ll'}$ is the Kronecker delta function, and p denotes the number of electrons. Alternatively, the ground state solution can be obtained by minimizing the total energy with orthogonality constraints [27]:

$$(1.2) \quad \begin{aligned} \min_{\Psi} \quad & E_{\mathrm{KS}}(\Psi) \\ \mathrm{s.\,t.} \quad & \langle \Psi, \Psi \rangle = I_p, \end{aligned}$$

where $\Psi = (\psi_1, \psi_2, \ldots, \psi_p)$, E_{KS} denotes the Kohn–Sham total energy, $\langle \cdot, \cdot \rangle$ stands for the inner product, and I_p denotes the $p \times p$ identity matrix. For notation brevity, we drop

[†]ICTEAM Institute, UCLouvain, Louvain-la-Neuve, Belgium (gaobin@lsec.cc.ac.cn).

[‡]Department of Mathematics, University of Macau, Macao SAR, China; Zhuhai UM Science & Technology Research Institute, Guangdong Province, China (garyhu@umac.mo).

[§]Corresponding author. Department of Mathematics, National University of Singapore, Singapore (matkuan@nus.edu.sg).

[¶]State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, and University of Chinese Academy of Sciences, China (liuxin@lsec.cc.ac.cn).

the subscript and let $I = I_p$. The detailed expressions of the Hamiltonian operator and the Kohn–Sham total energy are introduced in the next section.

1.1. Literature review and challenges. In electronic structure calculations, the pseudopotential approaches have proven to be successful in predicting electrical, magnetic and chemical properties for a wide range of materials [28]. However, the pseudopotentials can hardly construct the transition metals accurately [21] and tend to mispredict the material properties under extreme environment [31]. As a result, all-electron calculations which treat the Coulomb external potential exactly are in demand.

One of the most challenging aspects in all-electron calculations is the computational efficiency, which is usually dominated by two factors: the singularities arising from the Coulomb external potential and the orthogonality constraints of the wavefunctions.

To handle the singularities, the numerical discretization is generally required to be well designed in a manner such that it is able to capture the sharp variations of the orbitals and meanwhile describe the results on the regions where the orbitals vary slightly with the least effort. We focus on the finite element discretization (FEM) [30, 29, 3, 7] since it has local basis and allows a spatially adaptive resolution. Other discretizations handling the singularities can be found in [2, 8] and references therein.

When the quantum system is large, all-electron calculations turn into expensive [22]. In particular, to keep the orthogonality of the orbitals becomes the bottleneck in most existing algorithms. The self-consistent field (SCF) method and its variants [19, 17] are commonly used to solve the KS equation (1.1). However, the global convergence of the SCF-based algorithms can hardly be guaranteed [23, 24] and hence good initial guesses are often crucial for their performance. Since they lack robustness, the bad performance is often observed in numerical experiments [32]. This motivates the research on solving the total energy minimization problem (1.2) directly. Most of the first-order methods, such as QR retraction [33] and multipliers correction framework [11], carry out a feasible update. Namely, certain orthogonalization process is invoked in each iteration. Note that the orthogonalization process costs at least $\mathcal{O}(p^3)$ per iteration. Hence, these methods are not competent in solving large quantum systems due to this cubic complexity and the low scalability of any orthogonalization process.

Several algorithms have been exploited to avoid the orthogonalization. Linear scaling methods [6] build the solutions by direct minimization of unconstrained variational formulations. Note that most of them require to estimate the upper bound of the eigenvalue of the Hamiltonian [22], which is intractable in practice. Recently, an infeasible optimization algorithm based on the augmented Lagrangian method has been proposed in [12]. Here, "infeasible" indicates that the iterate is not required to satisfy the constraints in each iteration. The orthogonality can be guaranteed at any cluster point of the iteration sequence. Another favored property of this algorithm is that it is not sensitive with the choices of initial guess and parameters which makes it robust. Moreover, it is illustrated both theoretically and numerically that this algorithm does not highly rely on any priori knowledge of the studied system. All the calculations in which fully consist of BLAS operations. Thus it can be naturally parallelized. In view of these features, a parallelizable framework based on this infeasible minimization method for all-electron calculations is proposed.

1.2. Contribution. In this paper, we provide a competitive algorithm framework for all-electron calculations in the density functional theory. The framework consists of four parts shown in Figure 1, i.e., the pre-processing part for configuring the problem, the discretization part for numerically discretizing the continuous problem, the solving part for obtaining the solutions of the discretized system, and the post-processing part for transforming the numerical solutions for the further applications.
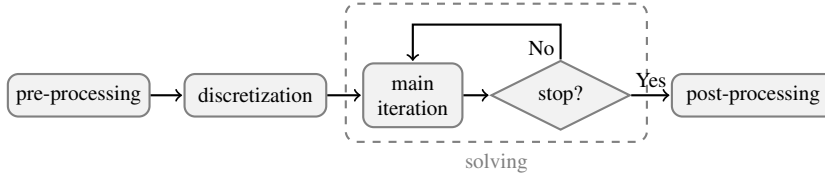
FIG. 1. *Flowchart of the framework for ground state calculations.*

The efficiency of all-electron calculations benefits from the following aspects of the proposed framework in Figure 1: i). a quality finite element space is designed for the given electronic structure based on the *a priori* analysis; ii). an orthogonalization-free method is proposed and analyzed for the discretized minimization problem; iii). high scalability is successfully demonstrated by numerical examples.

More specifically, in preparation of the tetrahedron mesh, the decay of the external potential is studied with the linear interpolation theory in [15, 29, 20], and a strategy on generating radial mesh for optimally capturing such decay is designed for a given electronic structure. It is noted that a quality finite element space would be built based on the radial mesh, and the efficiency of the algorithm would benefit from the sparsity of the discretized system and the mature and robust solvers for the sparse system such as the algebraic multigrid method.

The new method for the discretized optimization problem (1.2) is proposed by extending the parallelizable column-wise augmented Lagrangian (PCAL) [12] from the following two aspects. First, the PCAL is revised to handle the minimization problem with general orthogonality constraints $X^\top B X = I$ rather than the standard ones $X^\top X = I$. The global convergence of the new method is established theoretically. Second, a precondition strategy is proposed for the class of the PCAL methods, and a specific preconditioner is designed for all-electron calculations, which brings the dramatic acceleration for the convergence in the simulations.

As an attractive feature of the proposed algorithm, the robustness is successfully shown by a variety of numerical experiments, i.e., a random initial guess works for all numerical experiments in this paper, and the numerical convergence of the algorithm is not sensitive to the selection of the parameters. Finally, the high scalability of the algorithm is demonstrated by the numerical examples, which obviously indicates the potential of our algorithm for the large scale systems.

1.3. Notation and organization. $\mathbb{SR}^{p \times p} := \{S \in \mathbb{R}^{p \times p} \mid S^\top = S\}$ refers to the set of $p \times p$ real symmetric matrices. $\sigma_{\min}(A)$ denotes the smallest singular value of given real matrix A. $\mathrm{Diag}(v) \in \mathbb{SR}^{n \times n}$ denotes a diagonal matrix with all entries of $v \in \mathbb{R}^n$ in its diagonal, and $\mathrm{diag}(A) \in \mathbb{R}^n$ extracts the diagonal entries of matrix $A \in \mathbb{R}^{n \times n}$. For convenience, $\Theta(M) := \mathrm{Diag}(\mathrm{diag}(M))$ represents the diagonal matrix with the diagonal entries of square matrix M in its diagonal. $\mathrm{sym}(A) := \frac{1}{2}(A + A^\top)$ stands for the average of a square matrix and its transpose.

The organization of this paper is as follows. The KSDFT and its discretization are presented in section 2. In section 3, we present the algorithm and its convergence results. The implementation details of the proposed framework are introduced in section 4 and the numerical experiments are reported in section 5. In the end, we draw a brief conclusion and introduce the future works.

2. Finite Element Discretization for KSDFT. In this section, we introduce the detailed formulations for KSDFT and the discretization part as illustrated in Figure 1.

2.1. KSDFT. We consider a molecular system in $\mathbb{R}^3$ consisting of M nuclei of charges $Z_1, \ldots, Z_M$ locating at the positions $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_M$ and p electrons in the non-relativistic setting. The atomic unit is adopted in this work. Thus the Hamiltonian operator $\hat{H}$ in the Kohn–Sham equation (1.1) can be written as

$$(2.1) \qquad \hat{H} = -\frac{1}{2}\nabla^2 + V_{\text{ext}}(\boldsymbol{r}) + V_{\text{Har}}([\rho]; \boldsymbol{r}) + V_{\text{xc}}([\rho]; \boldsymbol{r}),$$

where the notation $V([\rho]; \boldsymbol{r})$ implies that V is a functional of the electron density $\rho(\boldsymbol{r}) = \sum_{l=1}^{p} |\psi_l(\boldsymbol{r})|^2$. The first term $-\nabla^2/2$ in $\hat{H}$ is the kinetic operator. The second term in $\hat{H}$ describes the Coulomb external potential due to the nuclei which takes the form

$$(2.2) \qquad V_{\text{ext}}(\boldsymbol{r}) = -\sum_{j=1}^{M} \frac{Z_j}{|\boldsymbol{r} - \boldsymbol{R}_j|}.$$

The third term is the Hartree potential describing the Coulomb repulsion among the electrons

$$(2.3) \qquad V_{\text{Har}}([\rho]; \boldsymbol{r}) = \int_{\mathbb{R}^3} \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r}'.$$

The last term V_{xc} stands for the exchange-correlation potential, which is caused by the Pauli exclusion principle and other non-classical Coulomb interactions. Note that the analytical expression for the exchange-correlation term is unknown and therefore an approximation is needed. Specifically, the local density approximation (LDA) from the library Libxc [25] is adopted in this work.

The total energy of the given quantum system consists of several parts:

$$(2.4) \qquad E_{\text{KS}} = E_{\text{kinetic}} + E_{\text{ext}} + E_{\text{Har}} + E_{\text{xc}} + E_{\text{nuc}},$$

where E_{kinetic} is the kinetic energy, and $E_{\text{ext}}, E_{\text{Har}}, E_{\text{xc}},$ and E_{nuc} are the potential energies induced by $V_{\text{ext}}, V_{\text{Har}}, V_{\text{xc}},$ and the nucleus-nucleus potential, respectively. Denoting the exchange-correlation energy per particle by $\epsilon_{\text{xc}}(\rho)$, then V_{xc} is the functional derivative of $\epsilon_{\text{xc}}(\rho)$ with respect to ρ, i.e., $V_{xc} = \delta\epsilon_{\text{xc}}(\rho)/\delta\rho$. As a result, it follows that

$$E_{\text{kinetic}} = \frac{1}{2}\sum_{l=1}^{p} \int_{\mathbb{R}^3} |\nabla\psi_l|^2 d\boldsymbol{r}, \quad E_{\text{ext}} = \int_{\mathbb{R}^3} V_{\text{ext}}\rho(\boldsymbol{r})d\boldsymbol{r}, \quad E_{\text{Har}} = \frac{1}{2}\int_{\mathbb{R}^3} V_{\text{Har}}\rho(\boldsymbol{r})d\boldsymbol{r},$$

$$E_{\text{xc}} = \int_{\mathbb{R}^3} \epsilon_{\text{xc}}\rho(\boldsymbol{r})d\boldsymbol{r}, \quad E_{\text{nuc}} = \sum_{j=1}^{M}\sum_{k=j+1}^{M} \frac{Z_j Z_k}{|\boldsymbol{R}_j - \boldsymbol{R}_k|}.$$

Note that E_{nuc} is a constant for the given system.

The ground state of the given system can be obtained from solving either the KS equation (1.1) or the total energy minimization problem (1.2). In order to numerically solve the continuous problem, we consider the finite element discretization.

2.2. Finite element discretization. In practical simulations, a bounded polyhedral domain $\Omega \subset \mathbb{R}^3$ is served as the computational domain. Thus the variational form of the Kohn–Sham equation (1.1) on Ω can be formulated as: Find $(\varepsilon_l, \psi_l) \in \mathbb{R} \times H_0^1(\Omega), l = 1, 2, \ldots, p,$ such that

$$(2.5) \qquad \begin{cases} \displaystyle\int_{\Omega} \varphi\hat{H}\psi_l d\boldsymbol{r} = \varepsilon_l \int_{\Omega} \psi_l\varphi d\boldsymbol{r}, \quad \forall\varphi \in H_0^1(\Omega), \\ \displaystyle\int_{\Omega} \psi_l\psi_{l'} d\boldsymbol{r} = \delta_{ll'}, \quad l' = 1, 2, \ldots, p, \end{cases}$$

where $H_0^1(\Omega) = \{\varphi \in H^1(\Omega) : \varphi|_\Omega = 0\}$ and $H^1(\Omega)$ is a standard Sobolev space.

To build a high quality finite element space to approximate the solution of (2.5) in all-electron calculations, the singularities stemming from the Coulomb potential in (2.2) should be prudently treated. In this work, we adopt a radial mesh generation strategy to resolve the difficulty brought by the singularities; see subsection 4.2 for details.

Assume that the linear finite element space $V_h \subset H_0^1(\Omega)$ is constructed on the bounded domain Ω partitioned by $\mathcal{T} = \{\mathcal{T}_K, K = 1, 2, \ldots, N_{\text{ele}}\}$, where N_{ele} represents the total number of elements of $\mathcal{T}$. Several commonly used notations in V_h are defined here. The basis functions are denoted by φ_i, $i = 1, \ldots, n$, where n is the dimension of V_h and the set of basis functions is denoted by $\mathcal{N} = (\varphi_1, \ldots, \varphi_n)^\top$. We construct the matrix of basis function $\mathcal{B}$ with $\mathcal{B}_{i,j} = \varphi_i\varphi_j$, then the symmetric mass matrix $B \in \mathbb{SR}^{n \times n}$ can be obtained from $B_{i,j} = \int_\Omega \mathcal{B}_{i,j} d\boldsymbol{r}$. Furthermore, a sequence of matrices $\{G_{(l)} \in \mathbb{SR}^{n \times n}, l = 1, \ldots, n\}$ with the entries $(G_{(l)})_{i,j} = \int_\Omega \mathcal{B}_{i,j}\varphi_l d\boldsymbol{r}$ are introduced. The discretized Laplacian $L \in \mathbb{SR}^{n \times n}$ on V_h is defined as $L_{i,j} = \int_\Omega \nabla\varphi_j \cdot \nabla\varphi_i d\boldsymbol{r}$.

On the finite element space V_h, the discretized variation form of (2.5) turns out: Find $(\varepsilon_l^h, \psi_l^h) \in \mathbb{R} \times V_h$, $l = 1, 2, \ldots, p$, such that

$$(2.6) \qquad \begin{cases} \displaystyle\int_\Omega \varphi \hat{H} \psi_l^h d\boldsymbol{r} = \varepsilon_l \int_\Omega \psi_l^h \varphi d\boldsymbol{r}, \quad \forall \varphi \in V_h, \\ \displaystyle\int_\Omega \psi_l^h \psi_{l'}^h d\boldsymbol{r} = \delta_{ll'}, \quad l' = 1, 2, \ldots, p. \end{cases}$$

We express the l-th wavefunction as $\psi_l^h = \sum_{i=1}^n X_{i,l}\varphi_i = X_l^\top \mathcal{N}$, where $X \in \mathbb{R}^{n \times p}$ and $X_{i,l}$ stands for the i-th degree of freedom of ψ_l^h. Then the electron density can be rewritten as

$$\rho(\boldsymbol{r}) = \sum_{l=1}^p (X_l^\top \mathcal{N})(X_l^\top \mathcal{N}) = \sum_{l=1}^p X_l^\top \mathcal{B} X_l = \text{tr}(X^\top \mathcal{B} X).$$

Note that the Hartree potential V_{Har} in (2.3) is also the solution to the Poisson equation $-\nabla^2 V_{\text{Har}} = 4\pi\rho(\boldsymbol{r})$. We denote the discretized Hartree potential by $U(X) \in \mathbb{R}^n$ such that $V_{\text{Har}} = U(X)^\top \mathcal{N}$. After the finite element discretization on the Poisson equation, U is calculated from the linear system $LU(X) = 4\pi \left(\text{tr}(X^\top G_{(1)}X), \ldots, \text{tr}(X^\top G_{(n)}X)\right)^\top$. In practical simulations, this linear system is solved by an efficient algebraic multigrid method [3].

Due to the arbitrary of φ in (2.6), we can choose $\varphi = \varphi_i$, $i = 1, \ldots, n$. In view of above expressions, finding the solution of the discretized variational form (2.6) turns out solving the generalized nonlinear eigenvalue problem:

$$(2.7) \qquad \begin{cases} H(X)X = BX\Xi, \\ X^\top B X = I_p, \end{cases}$$

where $\Xi = \text{Diag}(\varepsilon_1^h, \ldots, \varepsilon_p^h)$, $H(X) \in \mathbb{SR}^{n \times n}$ is the discretized Hamiltonian matrix which can be formulated from (2.1) as

$$(2.8) \qquad H(X) = \frac{1}{2}L + M_{\text{ext}} + M_{\text{Har}}(X) + M_{\text{xc}}(X).$$

The matrices $M_{\text{ext}}, M_{\text{Har}}(X), M_{\text{xc}}(X) \in \mathbb{R}^{n \times n}$ are defined as

$$(M_{\text{ext}})_{i,j} = \int_\Omega V_{\text{ext}}\mathcal{B}_{i,j}d\boldsymbol{r}, \quad (M_{\text{Har}})_{i,j} = \int_\Omega V_{\text{Har}}\mathcal{B}_{i,j}d\boldsymbol{r}, \quad (M_{\text{xc}})_{i,j} = \int_\Omega V_{\text{xc}}\mathcal{B}_{i,j}d\boldsymbol{r}.$$

We now represent the total energy (2.4) in the discretized form:

$$E_{\text{kinetic}}(X) = \frac{1}{2}\sum_{l=1}^{p}\int_{\Omega}\nabla\psi_l\cdot\nabla\psi_l dr = \frac{1}{2}\sum_{l=1}^{p}\int_{\Omega}X_l^{\top}\nabla\mathcal{N}\cdot X_l^{\top}\nabla\mathcal{N}dr = \frac{1}{2}\text{tr}(X^{\top}LX),$$

$$E_{\text{ext}}(X) = \int_{\Omega}V_{\text{ext}}\rho(\boldsymbol{r})dr = \int_{\Omega}V_{\text{ext}}\text{tr}(X^{\top}\mathcal{B}X)dr = \text{tr}(X^{\top}M_{\text{ext}}X),$$

$$E_{\text{Har}}(X) = \frac{1}{2}\int_{\Omega}V_{\text{Har}}\rho(\boldsymbol{r})dr = \frac{1}{2}\int_{\Omega}V_{\text{Har}}\text{tr}(X^{\top}\mathcal{B}X)dr = \frac{1}{2}\text{tr}(X^{\top}M_{\text{Har}}(X)X),$$

$$E_{\text{xc}}(X) = \int_{\Omega}\epsilon_{\text{xc}}\rho(\boldsymbol{r})dr = \int_{\Omega}\epsilon_{\text{xc}}\text{tr}(X^{\top}\mathcal{B}X)dr = \text{tr}(X^{\top}M_{\text{exc}}(X)X),$$

where the matrix $M_{\text{exc}}(X)$ in the last formula is defined as $(M_{\text{exc}})_{i,j} = \int_{\Omega}\varepsilon_{\text{xc}}\mathcal{B}_{i,j}dr$. Thus the discretized form of the minimization problem (1.2) is assembled as

$$\begin{array}{ll}
(2.9) & \min_{X\in\mathbb{R}^{n\times p}} \quad E_{\text{KS}}(X) = E_{\text{kinetic}}(X) + E_{\text{ext}}(X) + E_{\text{Har}}(X) + E_{\text{xc}}(X) + E_{\text{nuc}} \\
& \quad\text{s.t.} \quad X^{\top}BX = I_p.
\end{array}$$

The generalized orthogonality constraints in (2.9) are known as the generalized Stiefel manifold [1], denoted by $\mathcal{S}_{n,p}^{B} := \{X \in \mathbb{R}^{n\times p} : X^{\top}BX = I_p\}$. Note that the gradient of $E_{\text{KS}}(X)$ satisfies $\nabla E_{\text{KS}}(X) = 2H(X)X$, while we scale it as $\nabla E_{\text{KS}}(X) = H(X)X$ to be consistent with the convention.

3. Parallelizable Algorithms. In this section, we concentrate on the solving part in Figure 1. Namely, the discretized total energy minimization problem (2.9) is considered. We first state its optimality condition. Then a one step gradient-descent update is proposed for solving (2.9) and its global convergence result is established. We also develop an upgraded algorithm based on the column-wise block minimization with preconditioning.

The discretized total energy minimization problem (2.9) is a nonconvex constrained optimization problem due to the orthogonality constraints. We state its first-order optimality condition as follows.

DEFINITION 3.1. *Given $X \in \mathbb{R}^{n\times p}$, we call X a first-order stationary point of (2.9) if the following condition*

$$\begin{array}{ll}
(3.1) & \left\{\begin{array}{rcl}
\text{tr}(Z^{\top}\nabla E_{\text{KS}}(X)) & \geq & 0, \\
X^{\top}BX & = & I_p
\end{array}\right.
\end{array}$$

holds for any $Z \in \mathcal{T}_{\mathcal{S}_{n,p}^{B}}(X)$, where $\mathcal{T}_{\mathcal{S}_{n,p}^{B}}(X) := \{Z \in \mathbb{R}^{n\times p} : Z^{\top}BX + X^{\top}BZ = 0\}$ is the tangent space of $\mathcal{S}_{n,p}^{B}$ at X.

Following from [11, Lemma 2.2], it can be proved that the condition (3.1) is equivalent to

$$\begin{array}{ll}
(3.2) & \left\{\begin{array}{rcl}
(I_n - BXX^{\top})\nabla E_{\text{KS}}(X) & = & 0, \\
X^{\top}\nabla E_{\text{KS}}(X) & = & \nabla E_{\text{KS}}(X)^{\top}X, \\
X^{\top}BX & = & I_p.
\end{array}\right.
\end{array}$$

In fact, the second equality of (3.2) is automatically satisfied since $\nabla E_{\text{KS}}(X) = H(X)X$ and the Hamiltonian $H(X)$ is symmetric. Moreover, the condition (3.2) can be further reformulated as

$$\begin{array}{ll}
(3.3) & \left\{\begin{array}{l}
\nabla E_{\text{KS}}(X) = BX\Lambda, \\
X^{\top}BX = I_p,
\end{array}\right.
\end{array}$$

where the symmetric matrix $\Lambda \in \mathbb{SR}^{p \times p}$ can be regarded as the Lagrangian multipliers of the generalized orthogonality constraints. Multiplying the first equation from the left by $X^\top$, it follows that Λ reads the closed-form expression at any first-order stationary point,

$$(3.4) \qquad \Lambda = X^\top \nabla E_{\mathrm{KS}}(X) = X^\top H(X) X.$$

3.1. Main iteration: one step gradient-descent update. The infeasible method proposed in [12] has been proven to be efficient for solving the lager scale orthogonality constrained optimization problems. Briefly, the iterates are not required to be orthogonal. Meanwhile, the feasibility violation gradually decreases to zero until the method converges. This type of methods enables us to get rid of the unscalable computation for preserving constraints. In addition, it provides an opportunity to employ the multi-core machines and thus gain more scalability from parallel computing.

The algorithm in [12] originally aims to solve the problem with orthogonality constraints $(X^\top X = I)$, and in this subsection, we extend it to the general case $(X^\top B X = I)$ which is not a trivial task. The skeleton of this algorithm is based on the augmented Lagrangian method (ALM) [26]. Let X^k be the current iterate, the classical ALM has two major steps in each iteration:

1) Update the Lagrangian multipliers Λ^k;
2) Minimize the ALM subproblem to obtain X^{k+1},
   (3.5)
   $$\min_{X \in \mathbb{R}^{n \times p}} \mathcal{L}_\beta(X, \Lambda^k) := E_{\mathrm{KS}}(X) - \frac{1}{2} \left\langle \Lambda^k, X^\top B X - I_p \right\rangle + \frac{\beta}{4} \left\| X^\top B X - I_p \right\|_{\mathrm{F}}^2,$$

   where $\mathcal{L}_\beta(X, \Lambda^k)$ defines the augmented Lagrangian function of problem (2.9) and $\beta > 0$ is the penalty parameter.

This framework avoids being confronted with the generalized orthogonality constraints. Next, we discuss how to update these two steps efficiently.

For step 1), in view of the fact (3.4), we suggest the following update of Lagrangian multipliers

$$(3.6) \qquad \Lambda^k = X^{k^\top} H(X^k) X^k.$$

Due to the symmetry of the Hamiltonian $H(X^k)$, the above update provides symmetric multipliers Λ^k, which allows us to waive the symmetrization step, $\mathrm{sym}(X^{k^\top} H(X^k) X^k)$, in [12].

On the other side, the ALM subproblem in step 2) is an unconstrained optimization problem, and various methods can be applied to derive different updates. Instead of solving the subproblem to a certain preset precision, our strategy is to provide an approximate solution by an explicit formulation. We first introduce a proximal linearized approximation [5] to substitute the augmented Lagrangian function in (3.5). Specifically, we consider the subproblem

$$(3.7) \qquad \min_{X \in \mathbb{R}^{n \times p}} \left\langle \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k), X - X^k \right\rangle + \frac{\eta_k}{2} \left\| X - X^k \right\|_{\mathrm{F}}^2.$$

The parameter η_k measures the dominance of the proximal term. The solution of this quadratic subproblem reads an explicit form

$$X^{k+1} = X^k - \frac{1}{\eta_k} \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)$$

$$(3.8) \qquad = X^k - \frac{1}{\eta_k} \left(H(X^k) X^k - B X^k X^{k^\top} H(X^k) X^k + \beta B X^k (X^{k^\top} B X^k - I_p) \right),$$

where the last step is owing to the update formula (3.6). It implies that this modified ALM update is nothing but a vanilla gradient-descent step and $1/\eta_k$ specifies the stepsize.

Now we turn back to the solving part in Figure 1. By using the one step gradient-descent update (3.8) in the main iteration, we fulfill a solving part for KSDFT. The complete algorithm is described in Algorithm 1.

Algorithm 1: Proximal Linearized Augmented Lagrangian Algorithm (PLAM)

1 **Input:** discretization with $n \in \mathbb{N}$ and $B \in \mathbb{SR}^{n \times n}$; tolerance $\epsilon > 0$; initial guess $X^0 \in \mathbb{R}^{n \times p}$; Set $k := 0$.

2 **while** $\left\| (I_n - BX^k {X^k}^\top) H(X^k) X^k \right\|_{\mathrm{F}} + \left\| {X^k}^\top B X^k - I \right\|_{\mathrm{F}} > \epsilon$ **do**

3   $\quad$ Compute the Hamiltonian $H(X^k)$ by (2.8).

4   $\quad$ Update the variable X^{k+1} by (3.8).

5   $\quad$ Update the parameters η_k and β; Set $k := k + 1$.

6 **Output:** X^k.

Once the pre-processing and discretization are finished, the number of degrees of freedom n and the matrix B are fixed. Meanwhile, the initial guess X^0 can be generated by any popular strategy in KSDFT. In view of the condition (3.2), we notice that Line 2 (the stopping criteria) in Algorithm 1 is sufficient to check the first-order optimality. Line 3-5 are the main iterations in Figure 1. Indeed, those calculations in KSDFT can be well assembled in a parallel way. The gradient-descent update in Line 4 is the BLAS3 operation. The choices of parameters will be discussed in subsection 4.4. To sum up, the algorithm PLAM can be conveniently implemented since there is no matrix decomposition or eigen-solver. It completely consists of BLAS operations. Therefore, the algorithm PLAM is open to be parallelized. Note that SCF method can also be described by the framework Figure 1, and the only distinction between SCF and PLAM is the main iteration. Specifically, SCF replaces Line 3-5 with solving a linear eigenvalue problem from (2.7). By contrast, PLAM just carries out a one step gradient-descent update.

3.2. Convergence analysis. The global convergence of the plain PLAM for orthogonality constraints ($X^\top X = I$) has been studied in [12]. Next, we consider the generalized case, i.e., $X^\top B X = I$. It can be proved that the existing results are still applicable for Algorithm 1.

A natural idea to investigate the generalized orthogonality constraints is transforming it into the standard case. Since B is symmetric positive definite, there exists a symmetric positive definite matrix $G \in \mathbb{R}^{n \times n}$ satisfying $B = G^2$. By taking $Y = GX$, the problem (2.9) is equivalent to

$$(3.9) \qquad \begin{aligned} \min_{Y \in \mathbb{R}^{n \times p}} \quad & g(Y) := E_{\mathrm{KS}}(G^{-1}Y) \\ \text{s.t.} \quad & Y^\top Y = I_p. \end{aligned}$$

Thus the augmented Lagrangian function of (3.9) is defined as

$$\tilde{\mathcal{L}}_\beta(Y, \tilde{\Lambda}) = g(Y) - \frac{1}{2}\langle \tilde{\Lambda}, Y^\top Y - I_p \rangle + \frac{\beta}{4}\|Y^\top Y - I_p\|_{\mathrm{F}}^2.$$

The next lemma shows that the transform $Y = GX$ does not change the stationary points of problems.

LEMMA 3.2. *(i) X^* is a first-order stationary point of the problem* (2.9) *if and only if $Y^* = GX^*$ is also a first-order stationary point of the problem* (3.9).

(ii) X^* is a first-order stationary point of the ALM subprblem $\min_{X \in \mathbb{R}^{n \times p}} \mathcal{L}_\beta(X, \Lambda^*)$ *with $\Lambda^* = \mathrm{sym}(\nabla E_{\mathrm{KS}}(X^*)^\top X^*)$ if and only if $Y^* = GX^*$ is also a first-order stationary point of the ALM subprblem* $\min_{Y \in \mathbb{R}^{n \times p}} \tilde{\mathcal{L}}_\beta(Y, \tilde{\Lambda}^*)$ *with $\tilde{\Lambda}^* = \mathrm{sym}(\nabla g(Y^*)^\top Y^*)$.*

Proof. (i) Let $Y^* = GX^*$, it can be verified that

$$(I_n - Y^* Y^{*\top}) \nabla g(Y^*) = G^{-1}(I_n - BX^* X^{*\top}) \nabla E_{\mathrm{KS}}(X^*),$$
$$\nabla g(Y^*)^\top Y^* = \nabla E_{\mathrm{KS}}(X^*)^\top X^*,$$
$$Y^{*\top} Y^* - I_p = X^{*\top} BX^* - I_p.$$

Together with (3.2), we can conclude that problems (2.9) and (3.9) share the same first-order stationary points.

(ii) Let $Y^* = GX^*$. Similarly, it can be verified that

$$\tilde{\Lambda}^* = \mathrm{sym}(\nabla g(Y^*)^\top Y^*) = \mathrm{sym}(\nabla E_{\mathrm{KS}}(X^*)^\top X^*) = \Lambda^*,$$
$$\nabla_Y \tilde{\mathcal{L}}_\beta(Y^*, \tilde{\Lambda}^*) = G^{-1} \nabla_X \mathcal{L}_\beta(X^*, \Lambda^*).$$

These equalities lead to the desired equivalence. $\square$

In view of Lemma 3.2 and let $Y = GX$, the algorithm for problem (2.9) can be translated into an adaptation for (3.9). Next, we consider using PLAM to solve the problem (3.9). Recall that there are two major steps in the construction of PLAM:

1) For the multiplier update, we continue with the explicit update (3.6), i.e.,

$$\tilde{\Lambda}^k = \mathrm{sym}(\nabla g(Y^k)^\top Y^k).$$

2) We construct the subproblem with respect to Y,

$$(3.10) \qquad \min_{Y \in \mathbb{R}^{n \times p}} \left\langle \nabla_Y \tilde{\mathcal{L}}_\beta(Y, \tilde{\Lambda}^k), Y - Y^k \right\rangle_B + \frac{\eta_k}{2} \left\| Y - Y^k \right\|_{\mathrm{F}}^2.$$

where the inner product is defined as $\left\langle Y, \bar{Y} \right\rangle_B := \mathrm{tr}(Y^\top B \bar{Y})$.

Indeed, this subproblem has the closed-form solution

$$Y^{k+1} = Y^k - \frac{1}{\eta_k} B \nabla_Y \tilde{\mathcal{L}}_\beta(Y, \tilde{\Lambda}^k)$$
$$(3.11) \qquad = Y^k - \frac{1}{\eta_k} B \left(\nabla g(Y^k) - Y^k \Psi(\nabla g(Y^k)^\top Y^k) + \beta Y^k (Y^{k\top} Y^k - I_p) \right).$$

Using $Y = GX$ and the expression of $\tilde{\Lambda}^k$, it follows that the X-update (3.8) can be exactly recovered from (3.11). In other words, the algorithm PLAM for the X-problem (2.9) is proved to be equivalent to its adaptation for the Y-problem (3.9). Whereas the proximal linearized approximation in (3.10) differs from what we used in [12], the sketch of the convergence analysis is nearly the same. Therefore, the convergence results for PLAM can be accordingly migrated from [12].

Finally, we present the global convergence of PLAM without proofs. Interested readers are referred to [12] for a comprehensive understanding, such as the worst case complexity and local convergence rate.

ASSUMPTION 3.3. *$E_{\mathrm{KS}}(X)$ is twice differentiable.*

ASSUMPTION 3.4. *For a given $X^0 \in \mathbb{R}^{n \times p}$, we say it is a qualified initial guess, if there exists $\underline{\sigma} \in (0, 1)$ such that*

$$\sigma_{\min}(X^0) \geq \underline{\sigma}, \qquad 0 < ||X^{0\top} B X^0 - I_p||_{\mathrm{F}} \leq 1 - \underline{\sigma}^2.$$

THEOREM 3.5. *Let $\{X^k\}$ be the iterate sequence generated by Algorithm 1 initialized from X^0 satisfying Assumption 3.3 and Assumption 3.4. Suppose that the parameters β and η_k ($k = 1, \dots$) are sufficiently large, and in particular, the sequence $\{\eta_k\}$ is upper bounded. Then the sequence $\{X^k\}$ has at least one cluster point, and any which is a first-order stationary point of problem (2.9).*

3.3. An upgraded version of PLAM. According to the numerical reports in [12], the plain PLAM performs well in most problems, whereas its behavior is sensitive to the parameters β and η_k. In practice, it is always troublesome to tune these parameters as PLAM performs identically on different problems. Even worse, we cannot guarantee the boundedness of iterate sequences without restrictions on parameters.

Consequently, [12] suggests a column-wise block minimization for PLAM to overcome these limitations. In light of its motivation, we similarly impose the redundant column-wise constraints on the subproblem (3.7), and obtain the following subproblem.

$$(3.12) \qquad \min_{X \in \mathbb{R}^{n \times p}} \quad \left\langle \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k), X - X^k \right\rangle + \frac{\eta_k}{2} \left\| X - X^k \right\|_{\mathrm{F}}^2,$$
$$\text{s.t.} \quad \mathrm{Diag}(X^\top B X) = I.$$

Notice that the subproblem (3.12) is column-wisely separable. Thus, for the i-th column ($i = 1, \dots, p$), we can construct a subproblem with an extra constraint as follows,

$$(3.13) \qquad \min_{x \in \mathbb{R}^n} \quad \nabla_{X_i} \mathcal{L}_\beta(X^k, \Lambda^k)^\top (x - X_i^k) + \frac{\eta_k}{2} \|x - X_i^k\|_2^2,$$
$$\text{s.t.} \quad x^\top B x = 1,$$

where X_i denotes the i-th column of X. The redundant constraint is for restricting the iterate sequence to a compact set and hence make it bounded. The subproblem (3.13) has the closed-form solution

$$(3.14) \qquad X_i^{k+1} = \frac{X_i^k - \frac{1}{\eta_k} \nabla_{X_i} \mathcal{L}_\beta(X^k, \Lambda^k)}{\left\| X_i^k - \frac{1}{\eta_k} \nabla_{X_i} \mathcal{L}_\beta(X^k, \Lambda^k) \right\|_B},$$

where $\|x\|_B := \sqrt{x^\top B x}$ is a norm for any symmetric positive definite matrix B. Accordingly, the Lagrangian multipliers of X^k can be developed based on the new subproblem (3.12). In view of these formulations, an upgraded version of PLAM is listed in Algorithm 2 called PCAL.

Note that the update (3.15) for Lagrangian multipliers in PCAL is different from (3.6) in PLAM. When the redundant constraints, $\|X_i\|_B = 1$ ($i = 1, \dots, p$), are imposed, the corresponding optimality condition changes simultaneously. Specifically, the problem (2.9) with redundant constraints has the first-order optimality condition as follows,

$$(3.16) \qquad \begin{cases} \nabla E_{\mathrm{KS}}(X) = BX\Lambda + BXD, \\ X^\top B X = I_p. \end{cases}$$

The matrix $D \in \mathbb{R}^{p \times p}$ is diagonal and denotes the multipliers for extra constraints. Following a similar derivation of (3.3), it can be verified that Λ in (3.16) achieves the closed-form expression (3.15) at any first-order stationary point. Notice that the main calculation of PCAL

Algorithm 2: Parallelizable Column-wise Block Minimization for PLAM (PCAL)

---

1 **Input:** triangulation with $n \in \mathbb{N}$ and $B \in \mathbb{SR}^{n \times n}$; tolerance $\epsilon > 0$; initial guess $X^0 \in \mathcal{S}_{n,p}^B$; Set $k := 0$.

2 **while** $\left\| (I_n - BX^k X^{k^\top}) H(X^k) X^k \right\|_F + \left\| X^{k^\top} B X^k - I \right\|_F > \epsilon$ **do**

3 $\quad$ Compute the Hamiltonian $H(X^k)$ by (2.8).

4 $\quad$ Compute the Lagrangian multipliers by

$\quad$ (3.15)
$$\Lambda^k := X^{k^\top} H(X^k)^\top X^k + \Theta \left(X^{k^\top} \nabla_X L_\beta (X^k, X^{k^\top} H(X^k)^\top X^k) \right).$$

5 $\quad$ **for** $i = 1, \ldots, p$ **do**

6 $\quad\quad$ Update X_i^{k+1} by (3.14).

7 $\quad$ Update $X^{k+1} = [X_1^{k+1}, \ldots, X_p^{k+1}]$.

8 $\quad$ Update the parameters η_k and β; Set $k := k + 1$.

9 **Output:** X^k.

---

is a sequence of gradient-descent step with normalization. These for-loop computations are independent and hence can be executed in a parallel fashion. To sum up, the upgraded version of PLAM still enjoys the benefit of parallel computing.

In scientific computing, preconditioning is typically used to accelerate iterative algorithms. In [3], a preconditioner for the eigenvalue problem of SCF iteration has been proposed. It has the form of $T = \frac{1}{2}L - \lambda B$, where $\frac{1}{2}L$ is the discretized kinetic operator defined in (2.9) and λ is an approximated eigenvalue. Since $\frac{1}{2}L$ dominates the Hamiltonian, this preconditioner usually performs well in practical calculations. In view of the optimality condition (3.3), the update of Lagrangian multipliers (3.6) in PLAM can be viewed as the approximation of the eigenvalues. Thus, we choose $\Lambda_{ii}^k = \left(X^{k^\top} H(X^k) X^k \right)_{ii}$ to construct a preconditioner for the proposed algorithm:

$$(3.17) \qquad T_{(i)}^k = \begin{cases} \frac{1}{2}L - \Lambda_{ii}^k B, & \text{if } \Lambda_{ii}^k < 0, \\ I, & \text{otherwise,} \end{cases} \qquad \text{for } i = 1, \ldots, p.$$

Consequently, the one step gradient-descent update (3.8) in PLAM is preconditioned as

$$X_i^{k+1} = X_i^k - \frac{1}{\eta_k} \left(T_{(i)}^k \right)^{-1} \nabla_{X_i} \mathcal{L}_\beta (X^k, \Lambda^k), \quad \text{for } i = 1, \ldots, p,$$

where the preconditioned gradient can be assembled by solving p linear systems. Note that PCAL is compatible with this type of preconditioning providing that Λ^k is selected from (3.15). The parallelizable structure of PLAM and PCAL is still maintained as the preconditioning is conducted column-wisely. A test in Figure 2 verifies the effectiveness of the preconditioner (3.17) for both algorithms, where substationarity is computed by Line 2 in Algorithm 1.

4. Implementation Details. In this section, we introduce the implementation details of the framework (Figure 1) in solving the ground state. The quantum systems examined in this paper are introduced. In addition, several numerical issues in the simulations are discussed. In view of Figure 2, we observe that PCAL behaves more efficient and robust than PLAM, and thus we focus on PCAL in the following tests.
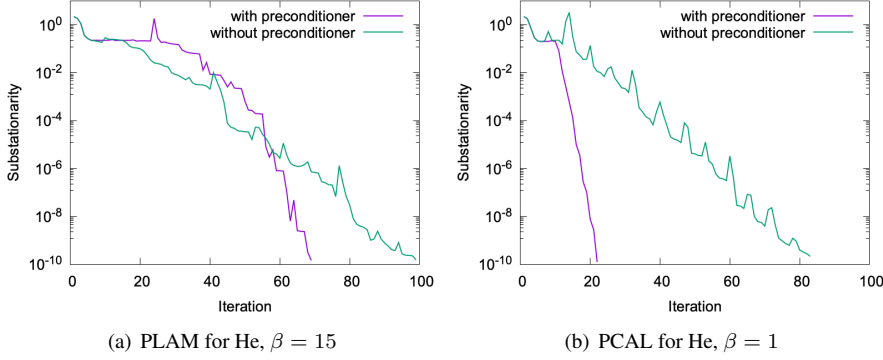
(a) PLAM for He, $\beta = 15$ (b) PCAL for He, $\beta = 1$

FIG. 2. *The performance of the preconditioner* (3.17) *for a helium (He) atom example with* $n = 1606, p = 1$.

All the simulations are performed on a workstation with two Intel(R) Xeon(R) Processors Silver 4110 (at 2.10GHz×8, 12M Cache) and 384GB of RAM, and the total number of cores is 16. The software is the C++ library AFEABIC [3] under Ubuntu 18.10.

4.1. Testing problems. A number of atom and molecules are simulated to illustrate the effectiveness and high scalability of the presented algorithm. It is noted that in practical simulations, p is regarded as the number of orbitals and each orbital is occupied by two electrons. The scale of testing systems, i.e., p, is ranging from 1 to 1152. In the formulation of problem (1.2), the exchange-correlation potential V_{xc} and exchange-correlation potential energy ϵ_{xc} per particle are obtained from the package Libxc [25]. The model equations for the various numerical examples are only different in the external potential term V_{ext} and precisely in the charge numbers and positions of the nuclei. The charge of a certain nucleus used in numerical experiments is listed in Table 1. The nuclei positions for small molecules are obtained from the calculated geometry part in CCCBDB [16] and for carbon nanotubes are from [10]. In summary, the following electronic structures He (1), LiH (2), CH$_4$ (5), H$_2$O (5), BF$_3$ (16), C$_6$H$_6$ (21), C$_{12}$H$_{10}$N$_2$ (48), C$_{60}$ (180), and carbon nanotubes C$_{96}$ (288), C$_{192}$ (576) and C$_{384}$ (1152) are tested, where the number in the bracket stands for the number of orbitals p in the associated system.

TABLE 1
Charge number Z_j of the nucleus.

	H	He	Li	B	C	N	O	F
Z_j	1	2	3	5	6	7	8	9

In practice, we evaluate the values for substationarity, feasibility violation and the total energy of each example during the simulations. Specifically, $kkt = \|H(X)X - BX\Lambda\|_F$, $fea = \|X^\top BX - I\|_F$, and the total energy E_{KS} is computed from (2.4). When the summation of kkt and fea is small enough, i.e., the following stopping criterion

$$\frac{kkt + fea}{kkt_0} < tol$$

is satisfied, we terminate the algorithm. Here, kkt_0 is the initial substationarity and tol denotes the tolerance and is chosen to be 1.0×10^{-8} in our simulations.

4.2. Pre-processing and discretization: mesh and initial guess generation. Once we determine the computational domain, a space discretization is generated for the ground state calculation. To resolve the singularities in the external potential term, a non-uniform mesh for the partition of the computational domain is introduced to obtain high accuracy with least effort. Specifically, a global mesh size function based on the external potential is adopted to generate the nonuniform mesh [20]. Within the linear finite element framework, to capture the $1/r$ decay in the external potential, the mesh size function locally behaves as $r^{6/5}$ for small r can be derived, where r represents the distance to the nucleus. Then we can construct the mesh size function $h(\boldsymbol{r})$ at the discretized point $\boldsymbol{r}$ as in [20]:

$$(4.1) \qquad h(\boldsymbol{r}) = \min\left\{ \gamma_1 Z_1^{-\frac{2}{5}} r_1^{\frac{6}{5}}, \cdots, \gamma_1 Z_M^{-\frac{2}{5}} r_M^{\frac{6}{5}}, \gamma_2 \right\},$$

where $r_j = |\boldsymbol{r} - \boldsymbol{R}_j|$ represents the distance to the j-th nucleus, γ_1 controls the resolution of the mesh, and γ_2 is the largest allowed mesh size. Note that (4.1) implies that the closer to the nucleus, the smaller the mesh size, i.e., the denser the mesh grid, which is as desired. Moreover, the distribution of the mesh grid around the nucleus with larger charge is also denser than that around the nucleus with a smaller charge. This can be verified from Figure 3 which shows an radial mesh example for the water molecule (H_2O).
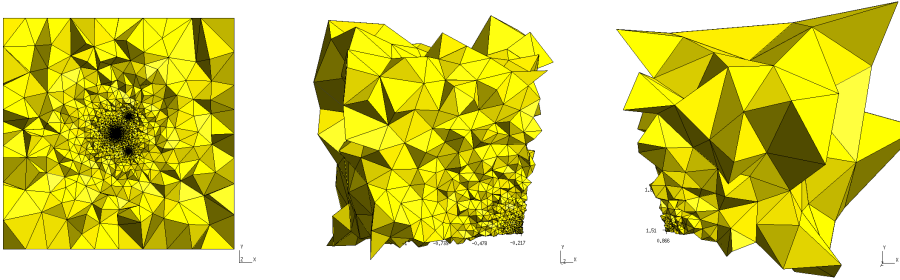

FIG. 3. *Left: the three dimensional mesh for molecule H_2O using the mesh size function (4.1) with $\gamma_1 = 0.15, \gamma_2 = 8$. Middle: the mesh around the oxygen nucleus $(-0.217, 0, 0)$ in X-Y plane $[-1.217, -0.217] \times [0, 1]$, on which the element shapes are kept. Right: the mesh around the hydrogen nucleus $(0.866, 1.509, 0)$ in X-Y plane $[0.866, 1.866] \times [1.509, 2.509]$. Generated by the software Gmsh v3.0.6 [14].*

In the following comparison, we choose a same randomly generated initial guess, $X^0 \in \mathbb{R}^{n \times p}$ satisfying $X^{0\top} B X^0 = I$, for different methods. Given a random matrix $V \in \mathbb{R}^{n \times p}$ from the pseudo-random number generator, X^0 is generated by the Cholesky-based Gram–Schmidt technique [13], i.e., $V = X^0 R$, where $R \in \mathbb{R}^{p \times p}$ is an upper triangular matrix. However, it is known that the SCF method may suffer a lot from divergence. For the sake of fairness, we stabilize SCF by improving the random initial guess with the imaginary time propagation (ITP) method [20] only when it diverges. By contrast, the numerical experiments in section 5 show that our algorithm behaves robust regardless of different initial guesses.

4.3. Post-processing: eigenvalue evaluation in the last step. In view of the presented infeasible methods, it is sufficient to output results such that X satisfies the orthogonality constraint $X^\top B X = I$. However, if we want to extract the desired wavefunctions from the eigenvectors of the generalized eigenvalue problem (2.7) or the other physical quantities based on the eigenvalues, we need to introduce a post-processing. This is due to the fact that X only provides an orthogonal basis of the desired eigenspace rather than the eigenvectors.

This can be implemented by solving a small $p \times p$ eigenvalue problem, $(X^\top H X)\tilde{X} = \lambda \tilde{X}$, with the Rayleigh-Ritz procedure to get the eigenvalues $\lambda_i, i = 1, \ldots, p$ and updating

X as $X = X\tilde{X}$ to get the wavefunctions. Note that this procedure is called only for once in the algorithm and it is of size $p \times p$. Consequently, its computational cost can be ignored compared to solving the optimization problem.

To verify the effectiveness of the post-procedure, we compute the eigenvalues of the Kohn–Sham equation of CH_4 system on the radial mesh with $n = 100127, p = 5$ for SCF and PCAL. The computational domain for this example is set as $[-20, 20]^3$, and the results are listed in Table 2. When the post-procedure is imposed, the eigenvalues from PCAL are well ordered and agree with eigenvalues from SCF. Moreover, it verifies that the post-procedure does not affect the energy value. In the practical simulations, the post-procedure step will be imposed as the final step of PCAL.

TABLE 2
Eigenvalue and energy evaluations for example CH_4.

	λ_0	λ_1	λ_2	λ_3	λ_4	E_{KS}
SCF	-9.75599	-0.66451	-0.38832	-0.38830	-0.38825	-40.24109
PCAL + post-processing	-9.75599	-0.66451	-0.38832	-0.38830	-0.38825	-40.24109

4.4. Choices of parameters. There are two major parameters in the algorithm PCAL. In view of Figure 2, the penalty parameter $\beta = 1$ works well for PCAL, and hence 1 is set as the default value of β in PCAL. Next, we investigate the proximal parameter η_k, whose reciprocal is the stepsize for the gradient-descent step in Algorithm 2. As suggested in [12], the Barzilai–Borwein (BB) strategy [4] is an efficient way to produce the stepsize,

$$\eta_k^{\mathrm{BB1}} := \frac{\left|\langle S^{k-1}, Y^{k-1}\rangle\right|}{\langle S^{k-1}, S^{k-1}\rangle}, \quad \text{or} \quad \eta_k^{\mathrm{BB2}} := \frac{\langle Y^{k-1}, Y^{k-1}\rangle}{\left|\langle S^{k-1}, Y^{k-1}\rangle\right|},$$

where $S^k = X^k - X^{k-1}$, $Y^k = \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) - \nabla_X \mathcal{L}_\beta(X^{k-1}, \Lambda^{k-1})$. It has other variations such as the Alternating BB strategy [9],

$$\eta_k^{\mathrm{ABB1}} := \begin{cases} \eta_k^{\mathrm{BB1}}, & \text{for odd } k, \\ \eta_k^{\mathrm{BB2}}, & \text{for even } k, \end{cases} \quad \text{or} \quad \eta_k^{\mathrm{ABB2}} := \begin{cases} \eta_k^{\mathrm{BB2}}, & \text{for odd } k, \\ \eta_k^{\mathrm{BB1}}, & \text{for even } k. \end{cases}$$

We test PCAL with four choices of the parameter η_k on several testing problems. The number of iterations to achieve convergence is recorded in Table 3. The notation "-" represents that the stopping criterion has not reached after 1000 iterations. This table reveals that PCAL with η_k^{BB2} behaves robust and has the best performance on number of iterations. As a result, we choose η_k^{BB2} as our default proximal parameter in the practical simulations.

TABLE 3
Number of iterations with different proximal parameters.

	He	LiH	CH_4	H_2O	C_6H_6
BB1	409	-	-	-	-
BB2	46	54	75	60	144
ABB1	86	90	129	180	291
ABB2	75	74	90	119	256

5. Numerical Examples. In this section, we numerically investigate the performance and parallel efficiency of the algorithm PCAL in all-electron calculations under the presented framework.

We test the classical SCF method and MOptQR for comparisons. MOptQR is a manifold-based optimization method which can be applied to the KSDFT [33]. All these methods are able to fulfill the solving part in the framework described in Figure 1. They are different in the main iteration: SCF solves a linear eigenvalue problem; PCAL produces a column-wise gradient-descent update; MOptQR searches along the Riemannian antigradient and projects the step onto the manifold by QR factorization. We choose the locally optimized block preconditioned conjugate gradient (LOBPCG) method [18] as the linear eigenvalue solver in SCF. Moreover, a simple mixing scheme is adopted for SCF, namely, $\rho^{(k+1)} = \alpha\tilde{\rho}^{(k+1)} + (1-\alpha)\rho^{(k)}$ where $\tilde{\rho}^{(k+1)}$ is the electron density obtained from solving the k-th step eigenvalue problem and α is the mixing parameter. In both SCF method and MOptQR, the orthogonalization process is implemented by the Cholesky-based Gram-Schmidt technique [13], which is shown to be more efficient than commonly-used Gram-Schmidt procedures.

In the serial setting, the leading order of computational costs is $\mathcal{O}(np^2)$ among these methods. The reason is that BLAS3 operations, such as $X^\top(BX)$, dominate the computing. While the function evaluation does not have a crucial impact on the cost due to the sparsity of discretized Hamiltonian H and mass matrix B. In the parallel setting, the total computation cost is divided into parallel and non-parallel parts. The above-mentioned leading cost $\mathcal{O}(np^2)$ (BLAS3) belongs to the parallel part. However, the orthogonalization process in SCF and MOptQR whose complexity is $\mathcal{O}(p^3)$ cannot be efficiently parallelized. When p is large, this cost is unaffordable in all-electron calculations. Conversely, PCAL is orthogonalization-free and completely consists of BLAS3 operations, and thus benefits a lot from parallel computing. These claims can be verified in the following experiments.

5.1. Ground state calculations. In this subsection, we test PCAL with SCF and MOptQR in all electron calculations of a list of atom and molecules under serial setting. For all the systems, the computational domain is set to be $[-20, 20]^3$. The mesh size function (4.1) is applied to generate the nonuniform mesh for each example. Note that the parameters in (4.1) are chosen as $\gamma_1 = 0.15, \gamma_2 = 8$ for C_6H_6 and $C_{12}H_{10}N_2$, and $\gamma_1 = 0.125, \gamma_2 = 8$ for the others. The preconditioner (3.17) is used in all the methods. For the system $C_{12}H_{10}N_2$, the initial guess is generated by ITP (see subsection 4.2), and we choose the mixing parameter $\alpha = 0.15$ for SCF to make it converge. For the other systems, we choose a random initial guess and the mixing parameter $\alpha = 0.3$.

The detailed numerical results are listed in Table 4, Figure 4 and Figure 5. We observe from Table 4 that: 1) the total energy E_{KS} obtained by PCAL agrees with SCF and MOptQR; 2) PCAL behaves more efficient than SCF and MOptQR in terms of the running time "CPU(s)"; 3) the number of iterations "N_{iter}" in PCAL is less than MOptQR. Note that the iteration numbers of SCF are always the smallest but conversely the CPU time. This is due to that the inner iterations, i.e., solving the linear eigenvalue problem, are required in each SCF iteration. The efficiency of PCAL can be also observed in Figure 4 for the example C_6H_6, from which we find that PCAL takes the least CPU time to converge at a given accuracy. In addition, the convergence results for PCAL are demonstrated in Figure 5. The first column displays the isosurface of the electron density, the last three columns present the convergence history of energy, substationarity and feasibility violation, respectively. We observe that the feasibility violation of PCAL gradually decreases until it converges. Note that the post-processing is not shown in this figure. In the He example, the feasibility violation is close to the machine accuracy since the normalization procedure is equivalent to the orthogonalization procedure in the case of $p = 1$.

5.2. Scalability. In this subsection, we investigate the parallel efficiency of PCAL. We first test all the algorithms on a single core and record the computational proportions of paral-

TABLE 4
The results in Kohn–Sham total energy minimization

| Solver	E_{KS}	kkt	N_{iter}	fea	CPU(s)	E_{KS}	kkt	N_{iter}	fea	CPU(s)
	He, $n = 34481, p = 1$					LiH, $n = 63725, p = 2$				
SCF	-2.86809	9.58_{-9}	36	2.88_{-15}	127	-7.98190	2.15_{-7}	39	2.09_{-14}	617
MOptQR	-2.86808	3.56_{-8}	36	3.55_{-15}	82	-7.98190	1.36_{-7}	70	2.70_{-15}	317
PCAL	-2.86808	5.99_{-9}	46	1.62_{-15}	99	-7.98190	1.25_{-7}	54	2.24_{-15}	264
	CH_4, $n = 141189, p = 5$					H_2O, $n = 149616, p = 5$				
SCF	-40.23775	4.83_{-7}	39	6.09_{-14}	3788	-75.83672	1.07_{-7}	44	3.64_{-14}	4246
MOptQR	-40.23775	1.24_{-7}	93	2.67_{-14}	1721	-75.83672	1.48_{-7}	74	2.45_{-14}	1413
PCAL	-40.23775	5.66_{-6}	75	1.59_{-14}	1283	-75.83672	1.33_{-7}	60	4.82_{-14}	1219
	C_6H_6, $n = 241939, p = 21$					$C_{12}H_{10}N_2$, $n = 522149, p = 48$				
SCF	-231.05824	5.11_{-7}	41	1.73_{-13}	43901	-571.60648	2.57_{-8}	94	1.64_{-13}	379482
MOptQR	-231.05824	3.60_{-7}	269	5.14_{-14}	21238	-571.60648	7.30_{-8}	501	1.63_{-13}	225856
PCAL	-231.05824	3.71_{-7}	144	7.35_{-14}	11013	-571.60648	5.37_{-8}	148	2.29_{-13}	89116

(a) Energy value

(b) Substationarity
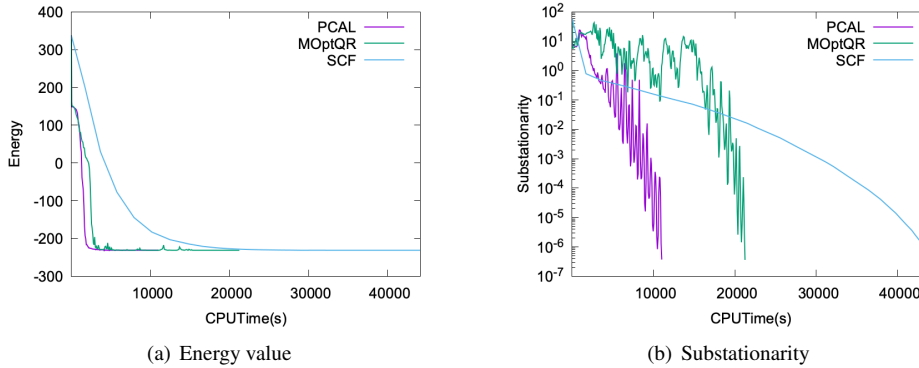
FIG. 4. *A comparison with different solvers for example C_6H_6.*

lel and non-parallel part in the total cost. We fix n to be around 100000 and choose different p, namely, the different molecules. By adjusting the parameters in subsection 4.2, we control the number of mesh grids n being as close as possible to 100000. The testing examples are BF_3 (16), $C_{12}H_{10}N_2$ (48), C_{60} (180), C_{96} (288), and C_{192} (576). The numerical results are displayed in Figure 6. We observe that the parallel part of PCAL will dominate the total cost when p becomes large. It is even higher than 99% When $p \geq 180$. This means that PCAL is suitable for parallel computing, especially when the scale of a system is very large. Meanwhile, it can be found that the non-parallel part of SCF and MOptQR becomes large when p increases. The main reason is the cubic complexity $\mathcal{O}(p^3)$ of orthogonalization process, which cannot be parallelized.

We next examine the scalability of PCAL in the parallel setting. The testing molecule is C_{384} which has 1152 occupied orbitals. The number of mesh grids n is set to be 380233. We run the code on different numbers of cores $\{4, 8, 12, 16\}$. The corresponding speedup factor is defined as

$$\text{speedup-factor} \ (m) = \frac{\text{wall-clock time for 4-core run}}{\text{wall-clock time for a } m\text{-core run}}.$$

The results are presented in Figure 7, from which we observe that the speedup factor of PCAL is close to the ideal one, and it achieves 3.76 for 16 cores. However, MOptQR has
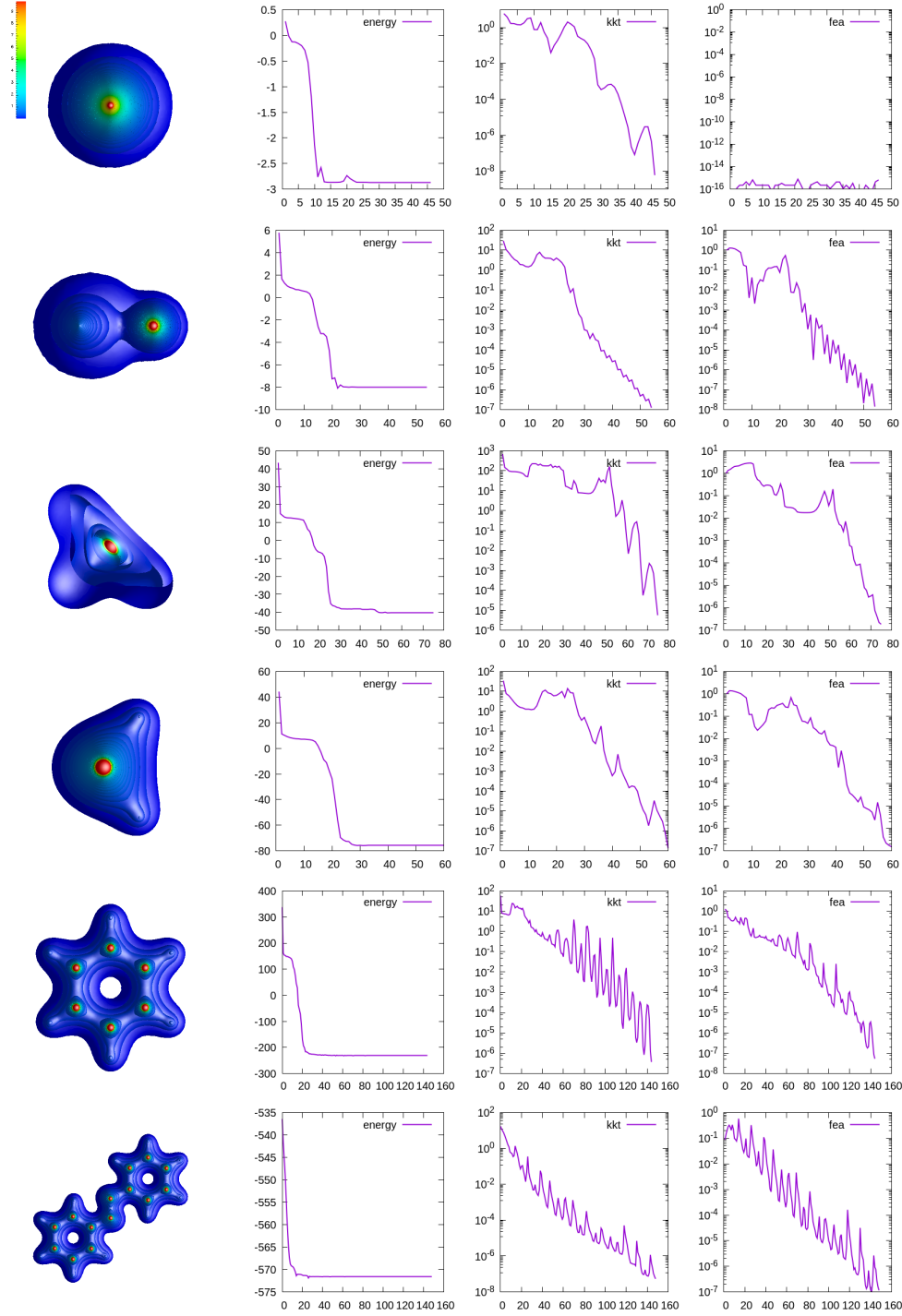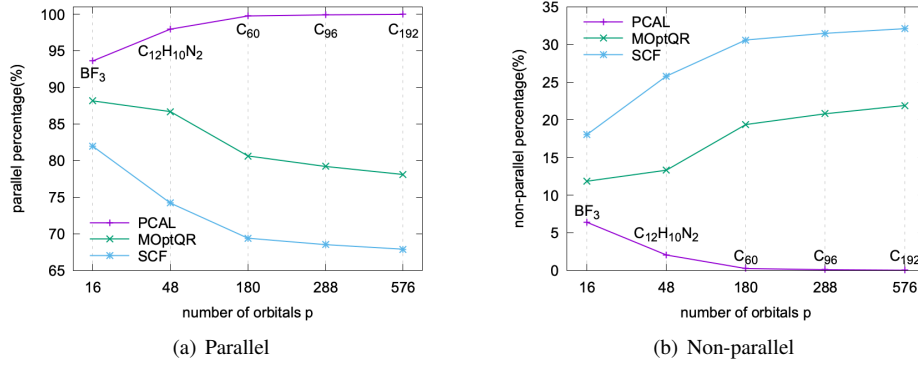
FIG. 5. *Convergence history of PCAL for He, LiH, CH_4, H_2O, C_6H_6, $C_{12}H_{10}N_2$ (from top to bottom). The left column displays the isosurface of each molecule. x-axis for the right 3 columns stands for the iteration step.*

(a) Parallel

(b) Non-parallel

FIG. 6. *Parallel proportion versus number of orbitals p.*

the low scalability and its speedup factor increases slowly. Note that the results of SCF are not recorded since the divergent phenomenon is observed. In view of Figure 6, even if we have the convergent results of SCF, it can be justifiably expected that the speedup factor of SCF will be smaller than that of MOptQR. In summary, the orthogonalization-free algorithm PCAL shows higher scalability and great potential than SCF and MOptQR.

(a) Structure

(b) Isosurface

(c) Speedup factor
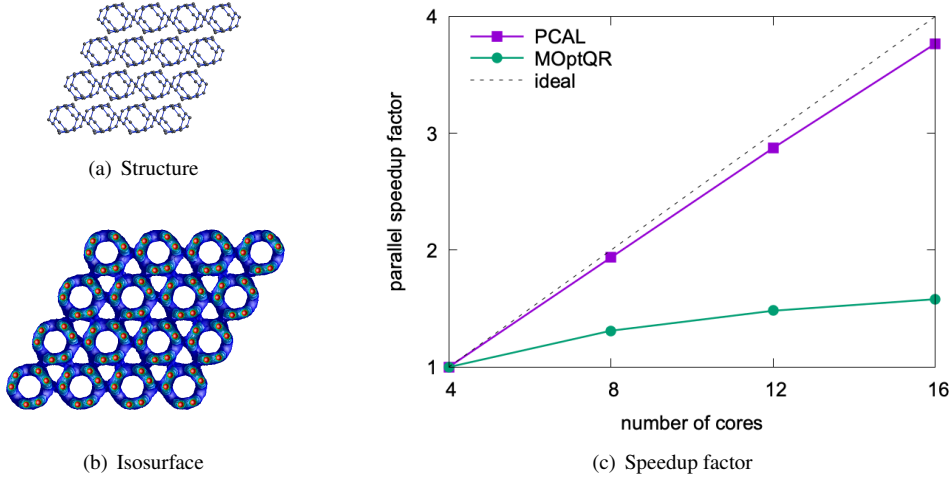
FIG. 7. *Example C_{384} with $n = 380233$, $p = 1152$.*

6. Conclusion. Based on the finite element method and PCAL algorithm, a scalable approach is proposed in this paper for the ground state solution of a given quantum system. To resolve the singularity introduced from the all-electron model, a radial mesh is generated according to the structure of the system, then the optimization problem is discretized in the associated finite element space. To avoid the efficiency bottleneck for large scale systems, i.e., the orthogonalization of those orbitals, the original PCAL method is extended and applied in this paper for solving the discretized optimization problem. A novel preconditioner is designed in the extended PCAL method, which generally accelerates the convergence in the simulations.

Comprehensive numerical experiments are implemented for different molecules. The effectiveness of the proposed method is well demonstrated by the comparison among the proposed method, the classical SCF method, and the MOptQR method in serial computing. Meanwhile, the robustness of the proposed method is fully demonstrated by its insensitivity to the initial guess and the algorithm parameters. The feature of the proposed method on improving the efficiency by avoiding the orthogonalization procedure is displayed clearly by the huge reduction of the CPU time in the comparison to the SCF method. More importantly, the excellent scalability of the proposed method is successfully shown in an experiment on a relatively large scale electronic system.

To improve the proposed method, the h-adaptive mesh method will be introduced for dynamically adjusting the finite element space according to the obtained numerical solutions. Furthermore, the preconditioner introduced in the PCAL method deserves more investigation in the following study, which has a chance to effectively accelerate the convergence of the numerical method towards the ground state. The improved method will be used for the numerical simulations of the Born-Oppenheimer molecular dynamics, to show the potential on the practical applications. The results will be reported in the forthcoming paper.

REFERENCES

[1] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2008, https://doi.org/10.1515/9781400830244.

[2] R. AHLRICHS, M. BÄR, M. HÄSER, H. HORN, AND C. KÖLMEL, *Electronic structure calculations on workstation computers: The program system turbomole*, Chem. Phys. Lett., 162 (1989), pp. 165–169, https://doi.org/10.1016/0009-2614(89)85118-8.

[3] G. BAO, G. HU, AND D. LIU, *An h-adaptive finite element solver for the calculations of the electronic structures*, J. Comput. Phys., 231 (2012), pp. 4967–4979, https://doi.org/10.1016/j.jcp.2012.04.002.

[4] J. BARZILAI AND J. M. BORWEIN, *Two-point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148, https://doi.org/10.1093/imanum/8.1.141.

[5] J. BOLTE, S. SABACH, AND M. TEBOULLE, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program., 146 (2014), pp. 459–494, https://doi.org/10.1007/s10107-013-0701-9.

[6] D. R. BOWLER AND T. MIYAZAKI, *$\mathcal{O}(N)$ methods in electronic structure calculations*, Rep. Prog. Phys., 75 (2012), p. 036503, https://doi.org/10.1088/0034-4885/75/3/036503.

[7] H. CHEN, X. DAI, X. GONG, L. HE, AND A. ZHOU, *Adaptive finite element approximations for Kohn–Sham models*, Multiscale Model. Sim., 12 (2014), pp. 1828–1869, https://doi.org/10.1137/130916096.

[8] O. COHEN, L. KRONIK, AND A. BRANDT, *Locally refined multigrid solution of the all-electron Kohn–Sham equation*, J. Chem. Theory Comput., 9 (2013), pp. 4744–4760, https://doi.org/10.1021/ct400479u.

[9] Y.-H. DAI AND R. FLETCHER, *Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming*, Numer. Math., 100 (2005), pp. 21–47, https://doi.org/10.1007/s00211-004-0569-y.

[10] J. T. FREY AND D. J. DOREN, *TubeGen 3.4 web-interface*, (2011), http://turin.nss.udel.edu/research/tubegenonline.html (accessed 2019-11-01).

[11] B. GAO, X. LIU, X. CHEN, AND Y.-X. YUAN, *A new first-order algorithmic framework for optimization problems with orthogonality constraints*, SIAM J. Optim., 28 (2018), pp. 302–332, https://doi.org/10.1137/16M1098759.

[12] B. GAO, X. LIU, AND Y.-X. YUAN, *Parallelizable algorithms for optimization problems with orthogonality constraints*, SIAM J. Sci. Comput., 41 (2019), pp. A1949–A1983, https://doi.org/10.1137/18M1221679.

[13] L. GENOVESE, A. NEELOV, S. GOEDECKER, T. DEUTSCH, S. A. GHASEMI, A. WILLAND, D. CALISTE, O. ZILBERBERG, M. RAYSON, A. BERGMAN, ET AL., *Daubechies wavelets as a basis set for density functional pseudopotential calculations*, J. Chem. Phys., 129 (2008), p. 014109, https://doi.org/10.1063/1.2949547.

[14] C. GEUZAINE AND J.-F. REMACLE, *Gmsh: A 3-D finite element mesh generator with built-in pre-and postprocessing facilities*, Int. J. Numer. Meth. Eng., 79 (2009), pp. 1309–1331, https://doi.org/10.1002/nme.2579.

[15] W. HUANG AND R. D. RUSSELL, *Adaptive moving mesh methods*, vol. 174, Springer Science & Business Media, 2010, https://doi.org/10.1007/978-1-4419-7916-2.

[16] R. D. J. III, *NIST Computational Chemistry Comparison and Benchmark Database*, (2006), http://cccbdb.nist.gov/ (accessed 2019-09-10).

[17] G. P. KERKER, *Efficient iteration scheme for self-consistent pseudopotential calculations*, Phys. Rev. B, 23 (1981), p. 3082, https://doi.org/10.1103/PhysRevB.23.3082.

[18] A. V. KNYAZEV, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541, https://doi.org/10.1137/S1064827500366124.

[19] W. KOHN AND L. J. SHAM, *Self-consistent equations including exchange and correlation effects*, Phys. Rev., 140 (1965), p. A1133, https://doi.org/10.1103/PhysRev.140.A1133.

[20] Y. KUANG AND G. HU, *On stabilizing and accelerating SCF using ITP in solving Kohn–Sham equation*, Commun. Comput. Phys., accepted.

[21] P. R. LEVASHOV, G. V. SIN'KO, N. A. SMIRNOV, D. V. MINAKOV, O. P. SHEMYAKIN, AND K. V. KHISHCHENKO, *Pseudopotential and full-electron DFT calculations of thermodynamic properties of electrons in metals and semiempirical equations of state*, J. Phys. Condens. Matter, 22 (2010), p. 505501, https://doi.org/10.1088/0953-8984/22/50/505501.

[22] L. LIN, J. LU, AND L. YING, *Numerical methods for Kohn–Sham density functional theory*, Acta Numer., 28 (2019), pp. 405–539, https://doi.org/10.1017/S0962492919000047.

[23] X. LIU, X. WANG, Z. WEN, AND Y. YUAN, *On the convergence of the self-consistent field iteration in Kohn–Sham density functional theory*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 546–558, https://doi.org/10.1137/130911032.

[24] X. LIU, Z. WEN, X. WANG, M. ULBRICH, AND Y. YUAN, *On the analysis of the discretized Kohn–Sham density functional theory*, SIAM J. Numer. Anal., 53 (2015), pp. 1758–1785, https://doi.org/10.1137/140957962.

[25] M. A. L. MARQUES, M. J. T. OLIVEIRA, AND T. BURNUS, *Libxc: A library of exchange and correlation functionals for density functional theory*, Comput. Phys. Commun., 183 (2012), pp. 2272–2281, https://doi.org/10.1016/j.cpc.2012.05.007.

[26] J. NOCEDAL AND S. J. WRIGHT, *Numerical optimization*, Springer Science & Business Media, 2006, https://doi.org/10.1007/978-0-387-40065-5.

[27] M. C. PAYNE, M. P. TETER, D. C. ALLAN, T. ARIAS, AND A. J. JOANNOPOULOS, *Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients*, Rev. Mod. Phys., 64 (1992), p. 1045, https://doi.org/10.1103/RevModPhys.64.1045.

[28] W. E. PICKETT, *Pseudopotential methods in condensed matter applications*, Comput. Phys. Rep., 9 (1989), pp. 115–197, https://doi.org/10.1016/0167-7977(89)90002-6.

[29] P. SURYANARAYANA, V. GAVINI, T. BLESGEN, K. BHATTACHARYA, AND M. ORTIZ, *Non-periodic finite-element formulation of Kohn–Sham density functional theory*, J. Mech. Phys. Solids, 58 (2010), pp. 256–280, https://doi.org/10.1016/j.jmps.2009.10.002.

[30] E. TSUCHIDA AND M. TSUKADA, *Adaptive finite-element method for electronic-structure calculations*, Phys. Rev. B, 54 (1996), p. 7602, https://doi.org/10.1103/PhysRevB.54.7602.

[31] H. Y. XIAO, X. D. JIANG, G. DUAN, F. GAO, X. T. ZU, AND W. J. WEBER, *First-principles calculations of pressure-induced phase transformation in AlN and GaN*, Comput. Mater. Sci., 48 (2010), pp. 768–772, https://doi.org/10.1016/j.commatsci.2010.03.028.

[32] C. YANG, J. C. MEZA, AND L.-W. WANG, *A trust region direct constrained minimization algorithm for the Kohn–Sham equation*, SIAM J. Sci. Comput., 29 (2007), pp. 1854–1875, https://doi.org/10.1137/060661442.

[33] X. ZHANG, J. ZHU, Z. WEN, AND A. ZHOU, *Gradient type optimization methods for electronic structure calculations*, SIAM J. Sci. Comput., 36 (2014), pp. C265–C289, https://doi.org/10.1137/130932934.