SIAM J. Sci. Comput.
Vol. 44, No. 3, pp. B723–B745

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 the nonlinear eigenvalue problem and the 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 BLAS operations and thus can be naturally parallelized. The global convergence of the proposed algorithm is established. We also present a preconditioning technique which can dramatically accelerate the convergence of the algorithm. The numerical experiments on all-electron calculations show the effectiveness and high scalability of the proposed algorithm.

Key words. density functional theory, all-electron calculations, total energy minimization, orthogonalization-free, scalability

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

DOI. 10.1137/20M1355884

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) [31], this can be achieved by solving the lowest p eigenpairs of the Kohn–Sham equation:

$$\hat{H}\psi_l(\boldsymbol{r}) = \varepsilon_l\psi_l(\boldsymbol{r}), \quad l = 1, 2, \ldots, p, \tag{1.1}$$

[*] Submitted to the journal's Computational Methods in Science and Engineering section July 28, 2020; accepted for publication (in revised form) January 6, 2022; published electronically June 7, 2022.

https://doi.org/10.1137/20M1355884

Funding: The work of the first author was supported by the Fonds de la Recherche Scientifique (FNRS) and the Fonds Wetenschappelijk Onderzoek - Vlaanderen under EOS project 30468160. The work of the second author was supported from National Natural Science Foundation of China (grants 11922120, 11871489), FDCT of Macao SAR (grant 0082/2020/A2), University of Macau (grants MYRG2020-00265-FST, MYRG2019-00154-FST), and Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications (grant 2020B1212030001). The work of the third author was supported by the Academic Research Fund of the Ministry of Education of Singapore under grant R-146-000-291-114. The work of the fourth author was partially supported by the National Natural Science Foundation of China (grants 12125108, 11971466, and 11991021), Key Research Program of Frontier Sciences, Chinese Academy of Sciences (grant ZDBSLY-7022).

[†] 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, Zhuhai, Guangdong, China; Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications, University of Macau, Macao SAR, China (garyhu@umac.mo).

[§] Corresponding author. School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou, Guangdong, China; Department of Mathematics, National University of Singapore, Singapore (ykuang@gdut.edu.cn).

[¶] 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, Beijing, China (liuxin@lsec.cc.ac.cn).

where $\hat{H}$ is the Hamiltonian operator, $\psi_l(\boldsymbol{r})$ is the lth wavefunction (eigenfunction), ε_l refers to the corresponding eigenenergy, and p denotes the number of occupied orbitals. In this paper, we consider closed-shell systems in which there are two electrons in each occupied orbital. Alternatively, the ground state solution can be obtained by minimizing the total energy with orthogonality constraints [43]:

$$\begin{array}{ll} \min\limits_{\Psi} & E_{\mathrm{KS}}(\Psi) \\ \mathrm{s.\,t.} & \langle \Psi, \Psi \rangle = I_p, \end{array} \tag{1.2}$$

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 the subscript and let $I = I_p$. The detailed expressions of the Hamiltonian operator and the Kohn–Sham total energy will be 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 [45]. However, the pseudopotentials can hardly construct the transition metals accurately [34] and tend to mispredict the material properties under extreme environment [54], though modern optimized norm-conserving Vanderbilt and projector augmented wave potential formulations have enabled accurate and efficient calculations for the whole of the periodic table over a wide range of external conditions, e.g., [25, 56]. 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 such a manner 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. Various discretization schemes have been proposed for all-electron calculations, for example, the finite difference method [12], the atomic basis set method [2], the finite element method [53, 51, 6], etc. Specifically, the finite element discretization, which has a local basis and allows a spatially adaptive resolution, has received a lot of attention in recent decades including the h-adaptive finite element method [9, 3, 4, 10, 14], the p-adaptive finite element method [47, 41, 38], the enriched finite element method [28, 46], and the FE method with a priori designed mesh [33, 50, 32]. Remark that the state-of-the-art all-electron calculations using a finite element basis are able to solve large systems containing more than 8000 electrons; see, e.g., [28, 40, 46]. In this work, we follow the finite element method presented in [32] to handle the singularities.

When the quantum system is large, all-electron calculations become expensive [35]. In particular, keeping the orthogonality of the orbitals becomes the bottleneck in most existing algorithms. The self-consistent field (SCF) method and its variants [31, 29] are commonly used to solve the Kohn–Sham equation (1.1). The SCF methods with different charge mixing schemes, for example, Pulay mixing and the Anderson mixing, have been shown to be robust and have been efficiently implemented in many widely used packages, e.g., [52, 40, 49, 15, 23, 22, 24, 17, 48, 36, 37, 55]. In addition to the SCF approaches, there are optimization methods based on solving the total energy minimization problem (1.2) directly. However, as the eigensolver in the SCF methods, most of the optimization methods, such as QR retraction [57] and

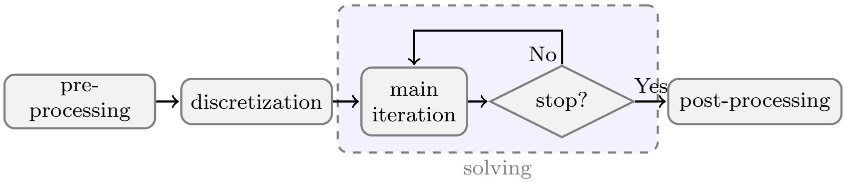


Fig. 1. *Flowchart of the framework for ground state calculations.*

multipliers correction framework [18], carry out an orthogonal update that preserves the orthogonality of the orbitals. Namely, a certain orthogonalization process such as the Cholesky-based Gram–Schmidt method [20] is invoked in each iteration. Note that the orthogonalization process costs at least $\mathcal{O}(p^3)$ per iteration. Hence, these methods may be 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 [8, 35] build the solutions by direct minimization of unconstrained variational formulations. Recently, an unconstrained optimization algorithm based on the augmented Lagrangian method has been proposed in [19]. Here, "unconstrained" 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 to 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 prior knowledge of the studied system. All the calculations in the algorithm fully consist of BLAS operations. Thus it can be naturally parallelized. In view of these features, a parallelizable framework based on this unconstrained minimization method for all-electron calculations is proposed.

1.2. Contribution.

In this paper, we provide an algorithm framework for all-electron calculations in the density functional theory. The framework consists of four parts shown in Figure 1, i.e., the preprocessing 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 postprocessing part for transforming the numerical solutions for further applications.

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 [26, 50, 32], 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 columnwise augmented Lagrangian (PCAL) method [19] from the following two aspects. First, the PCAL is revised to handle the minimization problem with general orthogonality constraints, $X^\top BX = I$, rather than the standard ones, $X^\top X = I$. The global convergence of the new method is established theoretically. Second, a preconditioning strategy is proposed for the class of the PCAL methods, and a specific preconditioner is designed for all-electron calculations, which dramatically accelerates 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 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 described in section 4, and the numerical experiments are reported in section 5. In the end, we draw a brief conclusion.

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 $2p$ electrons in the nonrelativistic 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

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

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

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

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

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

The last term V_{xc} stands for the exchange-correlation potential, which is caused by the Pauli exclusion principle and other nonclassical 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 from the library Libxc [39] is adopted in this work.

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

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

where E_{kinetic} is the kinetic energy and E_{ext}, E_{Har}, E_{xc}, and E_{nuc} are the potential energies induced by V_{ext}, V_{Har}, V_{xc}, and the nucleus-nucleus potential, respectively. Denoting the exchange-correlation energy per particle by $\epsilon_{\text{xc}}(\rho)$, 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 Kohn–Sher 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 $\Omega \subset \mathbb{R}^3$ serves as the computational domain. Thus the variational form of the Kohn–Sham equation (1.1) on Ω can be formulated as follows: Find $(\varepsilon_l, \psi_l) \in \mathbb{R} \times H_0^1(\Omega)$, $l = 1, 2, \ldots, p$, such that

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

where $H_0^1(\Omega) = \{\varphi \in H^1(\Omega) : \varphi|_{\partial\Omega} = 0\}$ 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) becomes the following: Find $(\varepsilon_l^h, \psi_l^h) \in \mathbb{R} \times V_h$, $l = 1, 2, \ldots, p$, such that

$$\left\{ \begin{aligned} &\int_\Omega \varphi \hat{H} \psi_l^h d\boldsymbol{r} = \varepsilon_l \int_\Omega \psi_l^h \varphi d\boldsymbol{r} \quad \forall \varphi \in V_h, \\ &\int_\Omega \psi_l^h \psi_{l'}^h d\boldsymbol{r} = \delta_{ll'}, \quad l' = 1, 2, \ldots, p. \end{aligned} \right. \tag{2.6}$$

We express the lth 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 ith degree of freedom of ψ_l^h. Then the electron density can be rewritten as

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

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 $LU(X) = 4\pi(\operatorname{tr}(X^\top G_{(1)} X), \ldots, \operatorname{tr}(X^\top G_{(n)} X))^\top$. In practical simulations, this linear system is solved by an efficient algebraic multigrid method [11].

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

$$\left\{ \begin{aligned} &H(X)X = BX\Xi, \\ &X^\top BX = I_p, \end{aligned} \right. \tag{2.7}$$

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

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

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:

$$\begin{aligned} E_{\text{kinetic}}(X) &= \frac{1}{2}\sum_{l=1}^p \int_\Omega \nabla\psi_l \cdot \nabla\psi_l d\boldsymbol{r} = \frac{1}{2}\sum_{l=1}^p \int_\Omega X_l^\top \nabla\mathcal{N} \cdot X_l^\top \nabla\mathcal{N} d\boldsymbol{r} = \frac{1}{2}\operatorname{tr}(X^\top LX), \\ E_{\text{ext}}(X) &= \int_\Omega V_{\text{ext}} \rho(\boldsymbol{r}) d\boldsymbol{r} = \int_\Omega V_{\text{ext}} \operatorname{tr}(X^\top \mathcal{B} X) d\boldsymbol{r} = \operatorname{tr}(X^\top M_{\text{ext}} X), \\ E_{\text{Har}}(X) &= \frac{1}{2}\int_\Omega V_{\text{Har}} \rho(\boldsymbol{r}) d\boldsymbol{r} = \frac{1}{2}\int_\Omega V_{\text{Har}} \operatorname{tr}(X^\top \mathcal{B} X) d\boldsymbol{r} = \frac{1}{2}\operatorname{tr}(X^\top M_{\text{Har}}(X) X), \\ E_{\text{xc}}(X) &= \int_\Omega \epsilon_{\text{xc}} \rho(\boldsymbol{r}) d\boldsymbol{r} = \int_\Omega \epsilon_{\text{xc}} \operatorname{tr}(X^\top \mathcal{B} X) d\boldsymbol{r} = \operatorname{tr}(X^\top M_{\text{exc}}(X) X), \end{aligned}$$

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

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

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

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

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

Following from [18, Lemma 2.2], it can be proved that the condition (3.1) is equivalent to

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

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

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

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 is given by the closed-form expression at any first-order stationary point:

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

3.1. Main iteration: One-step gradient-descent update.

The unconstrained method proposed in [19] has been proven to be efficient for solving large scale orthogonality constrained optimization problems. Briefly, the iterates are not required to be orthogonal. Meanwhile, the feasibility[1] violation, defined by $\|X^\top BX - I\|_{\mathrm{F}}$, gradually decreases to zero until the method converges. This type of method enables us to get rid of the unscalable computation for preserving constraints. In addition, it provides an opportunity to employ the multicore machines and thus gain better scalability from parallel computing.

The algorithm in [19] originally aims to solve the problem with orthogonality constraints ($X^\top X = I$), and in this subsection, we extend it to the general case ($X^\top BX = I$) which is not a trivial task. The skeleton of this algorithm is based on the augmented Lagrangian method (ALM) [42]. 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)
$$\min_{X\in\mathbb{R}^{n\times p}} \mathcal{L}_\beta(X,\Lambda^k) := E_{\mathrm{KS}}(X) - \frac{1}{2}\left\langle \Lambda^k, X^\top BX - I_p\right\rangle + \frac{\beta}{4}\left\|X^\top BX - I_p\right\|_{\mathrm{F}}^2,$$

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:

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

Because of the symmetry of the Hamiltonian $H(X^k)$, the above update provides symmetric multipliers Λ^k, which allows us to waive the symmetrization step in [19].

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 [7] to substitute the augmented Lagrangian function in (3.5). Specifically, we consider the subproblem

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

The parameter η_k measures the dominance of the proximal term. The solution of this quadratic subproblem is expressed by an explicit form:

$$\begin{aligned} X^{k+1} &= X^k - \frac{1}{\eta_k}\nabla_X\mathcal{L}_\beta(X^k,\Lambda^k)\\ &= X^k - \frac{1}{\eta_k}\left(H(X^k)X^k - BX^k{X^k}^\top H(X^k)X^k + \beta BX^k({X^k}^\top BX^k - I_p)\right), \qquad (3.8)\end{aligned}$$

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.

[1] The word "feasibility" is often used in constrained optimization; see [42]. Here, it means "orthogonality."

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 $\left\|(I_n - BX^k{X^k}^\top)H(X^k)X^k\right\|_{\mathrm{F}} + \left\|{X^k}^\top BX^k - I_p\right\|_{\mathrm{F}} > \epsilon$ do
3     Compute the Hamiltonian $H(X^k)$ by (2.8).
4     Update the variable $X^{k+1}$ by (3.8).
5     Update the parameters $\eta_k$ and $\beta$; Set $k := k + 1$.
6 Output: $X^k$.
```

Once the preprocessing 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. Lines 3–5 are the main iterations in Figure 1. Since the evaluation of the stopping criterion in line 2 is involved in (3.8), it is a byproduct of lines 3–4 and thus does not require additional costs except for F-norm calculations. In practice, the BLAS3 operations $X^\top H(X)X$ and $X^\top BX$ in (3.8) can be efficiently computed since B and H are sparse. The gradient-descent update in line 4 is also the BLAS3 operation. As a result, those calculations in KSDFT can be well assembled in a parallel way. 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 eigensolver. It completely consists of BLAS operations. Therefore, the algorithm PLAM is open to be parallelized. Note that the 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 lines 4–5 with solving a linear eigenvalue problem from (2.7), and in practice one or at most a few iterations in the eigensolver are enough to get to the next SCF iteration [40].

3.2. Convergence analysis.

The global convergence of the plain PLAM for orthogonality constraints ($X^\top X = I$) has been studied in [19]. Next, we consider the generalized case, i.e., $X^\top BX = I$. 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 $Y = GX$, problem (2.9) is equivalent to

$$\begin{array}{ll} \min\limits_{Y\in\mathbb{R}^{n\times p}} & g(Y) := E_{\mathrm{KS}}(G^{-1}Y) \\ \text{s.t.} & Y^\top Y = I_p. \end{array} \tag{3.9}$$

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

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

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

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

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

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

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

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

These equalities lead to the desired equivalence. □

In view of Lemma 3.2 and letting $Y = GX$, the algorithm for problem (2.9) can be translated into an adaptation for (3.9). Next, we consider using PLAM to solve 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,

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

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

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

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 [19], the sketch of the convergence analysis is nearly the same. Therefore, the convergence results for PLAM can be accordingly migrated from [19].

Finally, we present the global convergence of PLAM without proofs. Interested readers are referred to [19] 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, any of which is a first-order stationary point of problem* (2.9)*.*

3.3. An upgraded version of PLAM. According to the numerical reports in [19], 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, [19] suggests a columnwise block minimization for PLAM to overcome these limitations. In light of its motivation, we similarly impose the redundant columnwise constraints on the subproblem (3.7) and obtain the subproblem:

$$\begin{array}{ll} \min\limits_{X\in\mathbb{R}^{n\times p}} & \left\langle \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k), X - X^k \right\rangle + \frac{\eta_k}{2} \left\| X - X^k \right\|_{\mathrm{F}}^2 \\ \mathrm{s.\,t.} & \mathrm{Diag}(X^\top B X) = I. \end{array} \tag{3.12}$$

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

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^{n}} & \nabla_{X_i} \mathcal{L}_\beta(X^k, \Lambda^k)^\top (x - X_i^k) + \frac{\eta_k}{2} \left\| x - X_i^k \right\|_2^2 \\ \mathrm{s.\,t.} & x^\top B x = 1, \end{array} \tag{3.13}$$

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

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

where $\left\| x \right\|_B := \sqrt{x^\top B x}$ is a norm for any symmetric positive definite matrix B. Accordingly, the Lagrangian multipliers of X^k can be similarly developed based on the new subproblem (3.12). Specifically, we construct an update for the Lagrangian multipliers as follows:

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

where $\Theta(\cdot) = \mathrm{Diag}(\mathrm{diag}(\cdot))$. In view of these formulations, an upgraded version of PLAM is listed in Algorithm 2 and called PCAL.

Note that the update (3.15) for Lagrangian multipliers in PCAL is different from (3.6) in PLAM. When the redundant constraints, $\left\| X_i \right\|_B = 1$ ($i = 1, \dots, p$), are

Algorithm 2: Parallelizable columnwise block minimization for PLAM (PCAL)

```
Input: triangulation with $n \in \mathbb{N}$ and $B \in \mathbb{SR}^{n \times n}$; tolerance $\epsilon > 0$; initial
  guess $X^0 \in \mathcal{S}^B_{n,p}$; Set $k := 0$.
while $\left\|(I_n - BX^k {X^k}^\top) H(X^k) X^k\right\|_{\mathrm{F}} + \left\|{X^k}^\top B X^k - I_p\right\|_{\mathrm{F}} > \epsilon$ do
    Compute the Hamiltonian $H(X^k)$ by (2.8).
    Compute the Lagrangian multipliers by (3.15).
    for $i = 1, \ldots, p$ do
        Update $X_i^{k+1}$ by (3.14).
    Update $X^{k+1} = [X_1^{k+1}, \ldots, X_p^{k+1}]$.
    Update the parameters $\eta_k$ and $\beta$; Set $k := k + 1$.
Output: $X^k$.
```

imposed, the corresponding optimality condition changes simultaneously. Specifically, problem (2.9) with redundant constraints has the first-order optimality condition as follows:

$$\left\{ \begin{array}{rcl} \nabla E_{\mathrm{KS}}(X) & = & BX\Lambda + BXD, \\ X^\top BX & = & I_p. \end{array} \right. \tag{3.16}$$

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 is a sequence of gradient-descent steps 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.

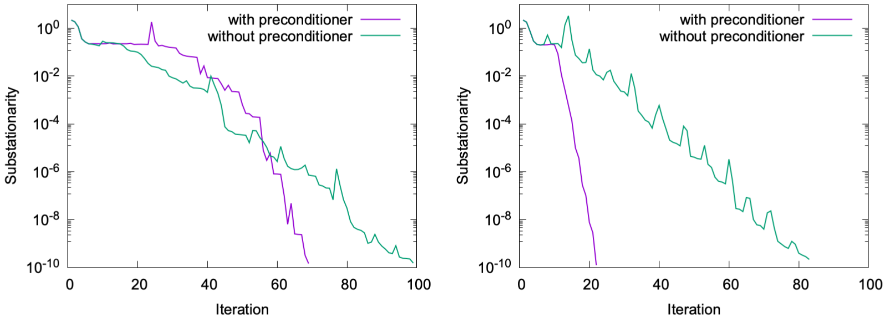
In scientific computing, preconditioning is typically used to accelerate iterative algorithms. In [4], a preconditioner for the eigenvalue problem of SCF iteration has been proposed. It has the form of $T = \frac{1}{2}L - \lambda B$, where $\frac{1}{2}L$ is the discretized kinetic operator defined in (2.9) and λ is an approximated eigenvalue. 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 $\Lambda^k_{ii} = ({X^k}^\top H(X^k) X^k)_{ii}$ to construct a preconditioner for the proposed algorithm:

$$T^k_{(i)} = \left\{ \begin{array}{ll} \frac{1}{2}L - \Lambda^k_{ii}B & \text{if } \Lambda^k_{ii} < 0, \\ I & \text{otherwise} \end{array} \right. \quad \text{for } i = 1, \ldots, p. \tag{3.17}$$

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

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

where the preconditioned gradient can be assembled by solving p linear systems. In practice, we can use the algebraic multigrid method [11] to solve the linear systems, and we find that 5 or fewer iteration steps are enough to obtain a quality preconditioner. As a result, the cost of the preconditioning process contributes only a small



(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* $n = 1606, p = 1$.

part to the total cost of the algorithm. Nevertheless, the parallel scalability of multigrid methods may be limited. In order to achieve efficient large scale calculations, it is crucial to implement this preconditioning in a sufficiently well scaling parallel way. 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 columnwise. 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 efficiently and robustly than PLAM; thus we focus on PCAL in the following tests.

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 [4] under Ubuntu 18.10.

4.1. Testing problems.

A number of atoms and molecules are simulated to illustrate the effectiveness and high scalability of the presented algorithm. The scale of testing systems, i.e., the number of orbitals 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 [39]. Specifically, the exchange energy functional E_{x} is the Slater $\rho^{1/3}$ functional

$$E_{\mathrm{x}} = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3}\rho^{1/3},$$

and the correlation functional E_{c} from [44] is used which takes the form

$$E_{\mathrm{c}}\left(r_s\right) = \begin{cases} -0.1423/\left(1+1.0529\sqrt{r_s}+0.3334r_s\right), & r_s \geqslant 1; \\ 0.0311\log r_s - 0.048 + 0.002r_s\log r_s - 0.0116r_s, & r_s < 1, \end{cases}$$

where $r_s = (3/(4\pi\rho))^{1/3}$ is defined as the radius of a sphere containing one electron.

Table 1
Charge number Z_j of the nucleus.

	H	He	Li	C	N	O
Z_j	1	2	3	6	7	8

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 of the Computational Chemistry Comparison and Benchmark Database [27] and for carbon nanotubes from [16]. In summary, the electronic structures He (1), LiH (2), CH_4 (5), H_2O (5), C_6H_6 (21), $C_{12}H_{10}N_2$ (48) and carbon nanotube C_{384} (1152) are tested, where the number in the bracket stands for the number of orbitals p in the associated system.

In practice, we evaluate the values for substationarity, feasibility violation, and the total energy of each example during the simulations. $kkt = \|H(X)X - BX\Lambda\|_{\text{F}}$, $fea = \left\|X^\top BX - I\right\|_{\text{F}}$, and the total energy E_{KS} is computed from (2.4). When the summation of kkt and fea is small enough, i.e., the 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 10^{-8} unless specifically stated. The relationship between the error of ground state energy and the tolerance is discussed in subsection 5.1.

4.2. Preprocessing 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 nonuniform 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 [32]. Within the linear finite element framework, to capture the $1/r$ decay in the external potential, the mesh size function locally behaving 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 [32]:

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

where $r_j = |\boldsymbol{r} - \boldsymbol{R}_j|$ represents the distance to the jth 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 a radial mesh example for the water molecule (H_2O). Remark that once the parameters γ_1 and γ_2 are fixed, the number of mesh grids will increase and the calculations become costly as the atomic number increases. Adaptive finite element methods or using higher order finite elements can mitigate the problem effectively, but this is out of the scope of this paper.

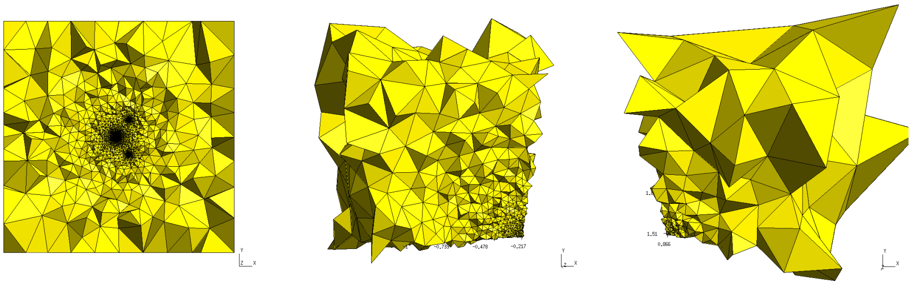


FIG. 3. *Left: the three dimensional mesh for molecule H_2O using the mesh size function* (4.1) *with* $\gamma_1 = 0.15, \gamma_2 = 8$. *Middle: the mesh around the oxygen nucleus* $(-0.217, 0, 0)$ *in X-Y plane* $[-1.217, -0.217] \times [0, 1]$*, on which the element shapes are kept. Right: the mesh 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 v*3.0.6 [21].

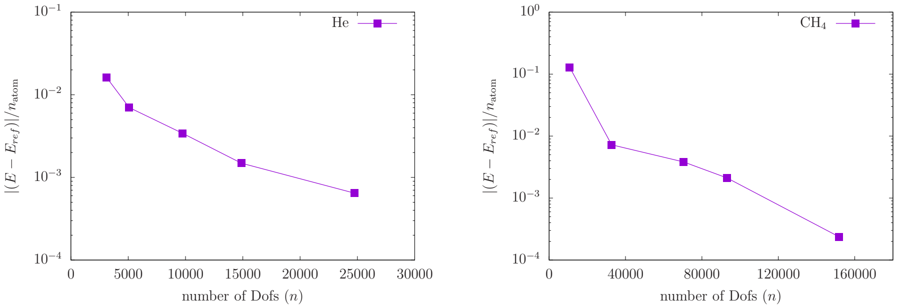


FIG. 4. *Energy difference with respect to the number of Dofs of He and CH_4 (in Hartree unit). n_{atom} is the number of atoms. The reference value $E_{ref} = -2.834826$ Ha for helium is calculated from* `NWChem` *with the* aug-cc-pv6z *basis set. The reference value $E_{ref} = -40.121849$ Ha for methane is calculated from* `NWChem` *with the* aug-cc-pv5z *basis set.*

In order to verify the accuracy of the finite element discretization, we calculate the ground state energy on different meshes generated by adjusting the parameters in (4.1). The reference value is generated from a widely used electronic structure calculation software, `NWChem` [52]. The results are listed in Figure 4. From the figure we can find that it takes around 2.5×10^4 degrees of freedom (Dof) for the helium example to achieve 10^{-3} Ha accuracy per atom (Ha is the energy unit Hartree), and it takes 1.5×10^5 Dofs for the methane example to achieve such accuracy. Moreover, we observe that most of the tested examples in this paper can be produced at such an accuracy. Hence, in the following experiments, we do not report the accuracy and mainly focus on the studies of convergence and scalability.

It is worth noting that the proposed algorithm is able to accept a random initial guess, which shows its robustness. In the following comparison, we choose the 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 pseudorandom number generator, X^0 is generated by the Cholesky-based Gram–Schmidt technique [20], i.e., $V = X^0 R$, where $R \in \mathbb{R}^{p\times p}$ is an upper triangular matrix. Nevertheless, one can always benefit from better initial approximations in practice (e.g., [40]), and hence, comparisons of timings and/or iterations to convergence using random initial guesses, as in the present work, may be different from the practice.

Table 2

Eigenvalues and energy for the example CH_4. The relative error is computed between PCAL and `NWChem`.

	λ_0	λ_1	λ_2	λ_3	λ_4	E_{KS}
SCF	-9.7582	-0.6493	-0.3731	-0.3731	-0.3731	-40.12066
PCAL + postprocessing	-9.7582	-0.6493	-0.3731	-0.3731	-0.3731	-40.12066
`NWChem`	-9.7609	-0.6221	-0.3474	-0.3474	-0.3474	-40.12185
Relative error	3×10^{-4}	4×10^{-2}	7×10^{-2}	7×10^{-2}	7×10^{-2}	3×10^{-5}

4.3. Postprocessing: Eigenvalue evaluation in the last step.

In view of the presented unconstrained methods, it is sufficient to output results such that X satisfies the orthogonality constraint $X^\top BX = I$. 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 postprocessing. 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, $(X^\top HX)\tilde{X} = \lambda\tilde{X}$, with the Rayleigh–Ritz procedure to get the eigenvalues $\lambda_i, i = 1, \dots, p$, and updating X as $X = X\tilde{X}$ to get the wavefunctions. Note that this procedure is called only for one time 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 postprocedure, we compute the eigenvalues of the Kohn–Sham equation of the CH_4 system on the radial mesh with $n = 151786, p = 5$ for SCF and PCAL. Moreover, we compare the converged results with the reference values obtained from the package `NWChem` [52] using the basis set $a$ug-cc-pv5z. The computational domain for this example is set as $[-20, 20]^3$ atomic unit (a.u.), and the results are listed in Table 2. When the postprocedure is imposed, the eigenvalues from PCAL are well ordered and agree with eigenvalues from SCF. In addition, we find that the smallest eigenvalue in PCAL matches `NWChem` at a 3×10^{-4} accuracy and the largest eigenvalue matches `NWChem` at a 7×10^{-2} accuracy.

It is worth noting that in the practical simulations, the postprocedure step will be imposed as the final step of PCAL.

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 [19], the Barzilai–Borwein (BB) strategy [5] is an efficient way to produce the stepsize:

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

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 Barzilai–Borwein (ABB) strategy [13]:

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

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 "-"

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

represents that the stopping criterion has not been reached after 1000 iterations. This table reveals that PCAL with η_k^{BB2} behaves robustly 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.

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 SCF method and a QR-based manifold optimization method (MOptQR) which can be applied to the KSDFT [57]. 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 columnwise 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 method [30] as the linear eigenvalue solver in SCF. The tolerance for the residual of each eigenfunction, i.e., $\|HX_i - \lambda_i BX_i\|_2$, is set as 10^{-8}, and the number of maximum allowed iterations is 10; the tolerance for solving linear systems in the Hartree potential is set as 10^{-8}; the tolerance for preconditioning is set as 10^{-6}. Moreover, an Anderson mixing scheme is adopted for SCF using at most 5 adjacent results. In both the SCF method and MOptQR, the orthogonalization process is implemented by the Cholesky-based Gram–Schmidt technique [20], 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, one challenge is to develop an efficiently parallelizable method for the orthogonalization process in SCF and MOptQR methods whose complexity is $\mathcal{O}(p^3)$. There are some success efforts on resolving this issue; e.g., see [40]. Instead, 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 compare PCAL with SCF and MOptQR in all-electron calculations of a list of atoms and molecules under the serial setting. In order to demonstrate the effectiveness and ability of PCAL, the following experiments are divided into two classes. For all the systems, the computational domain is set to be $[-20, 20]^3 a.u.$ 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.

Table 4
The results in Kohn–Sham total energy minimization.

Solver	E_{KS}	kkt	N_{iter}	fea	E_{KS}	kkt	N_{iter}	fea
	He, $n = 34481, p = 1$				LiH, $n = 63725, p = 2$			
SCF	-2.86809	2.17_{-8}	31	4.13_{-15}	-7.98190	5.48_{-8}	29	3.19_{-14}
MOptQR	-2.86808	3.56_{-8}	36	3.55_{-15}	-7.98190	1.36_{-7}	70	2.70_{-15}
PCAL	-2.86808	5.99_{-9}	46	1.62_{-15}	-7.98190	1.25_{-7}	54	2.24_{-15}
	CH_4, $n = 141189, p = 5$				H_2O, $n = 149616, p = 5$			
SCF	-40.23775	4.92_{-7}	25	1.91_{-14}	-75.83672	1.25_{-7}	27	5.41_{-14}
MOptQR	-40.23775	1.24_{-7}	93	2.67_{-14}	-75.83672	1.48_{-7}	74	2.45_{-14}
PCAL	-40.23775	5.66_{-6}	75	1.59_{-14}	-75.83672	1.33_{-7}	60	4.82_{-14}
	C_6H_6, $n = 241939, p = 21$				$C_{12}H_{10}N_2$, $n = 522149, p = 48$			
SCF	-231.05824	1.76_{-7}	28	7.69_{-14}	-571.60648	1.39_{-8}	34	2.27_{-14}
MOptQR	-231.05824	3.60_{-7}	269	5.14_{-14}	-571.60648	7.30_{-8}	501	1.63_{-13}
PCAL	-231.05824	3.71_{-7}	144	7.35_{-14}	-571.60648	5.37_{-8}	148	2.29_{-13}

The PCAL method is compared with SCF and MOptQR on different systems. The detailed numerical results are listed in Table 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) the number of iterations "N_{iter}" in PCAL is less than MOptQR. Note that the iteration numbers of SCF are always the smallest. This is due to the fact that the inner iterations, i.e., solving the linear eigenvalue problem, are required in each SCF iteration. 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 postprocessing is not shown in this figure. In the helium example, the feasibility violation is close to the machine accuracy since the normalization procedure is equivalent to the orthogonalization procedure in the case of $p = 1$.

In addition, in order to investigate the varying behavior of the error in the ground state energy corresponding to the choice of the stopping criterion and tolerance in PCAL, we produce the results in Figure 6 for the methane and benzene molecules. We can find that the smaller tolerance, the smaller error in the ground state energy, which meets our expectation.

5.2. Scalability. In this subsection, we investigate the parallel efficiency of PCAL. We examine the scalability of PCAL in the parallel setting on a workstation with two AMD EPYC 7H12 processors (at 2.42GHz×64, 64M cache) and 1024GB of RAM, and the total number of cores is 128. 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 $\{4, 8, 16, 32, 64, 128\}$. 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. However, MOptQR has low scalability, and its speedup factor increases slowly.

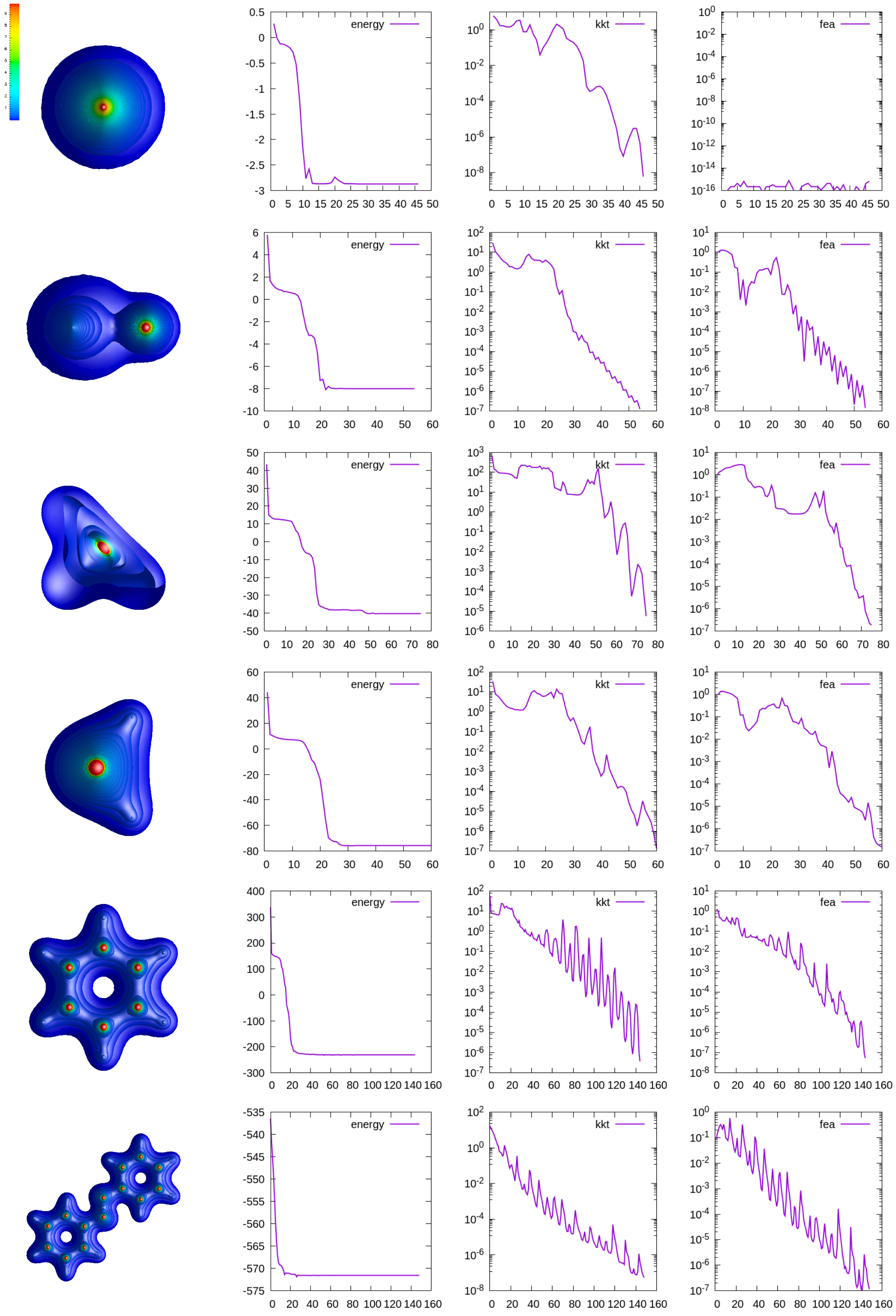


Fig. 5. *Convergence history of PCAL for He, LiH, CH_4, H_2O, C_6H_6, $C_{12}H_{10}N_2$ (from top to bottom). The left column displays the isosurface of each molecule. x-axis for the right 3 columns stands for the iteration step.*

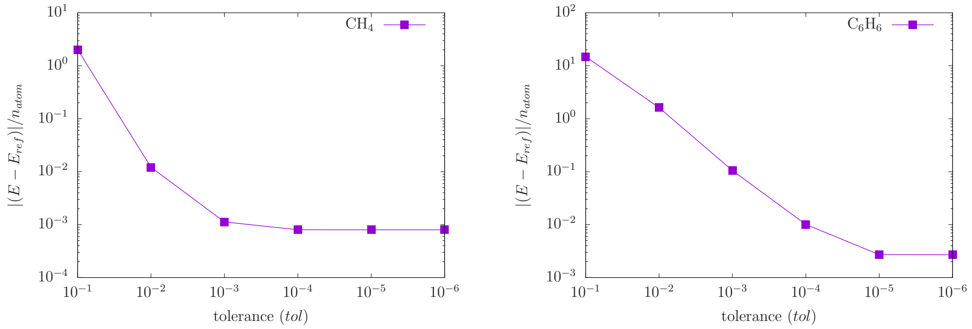
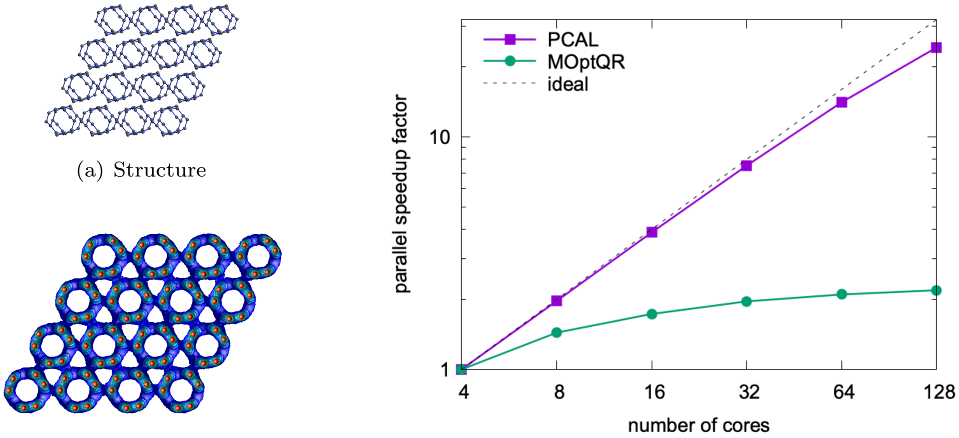


FIG. 6. *The error of the ground state energy with respect to the different tolerance in PCAL for CH_4 and C_6H_6 molecules. E_{ref} is obtained from* `NWChem`*.*



(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 the 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 converged results of the proposed method are verified by comparing with a state-of-the-art package, `NWChem`. The effectiveness of the proposed method is well demonstrated by the comparison among the proposed method, the 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.

One of the main features of the method proposed in this work is its high scalability, which is attractive for applying our method potentially on large systems. Although from the numerical experiments the effectiveness of the method has been observed successfully, there is room left for further improving the numerical efficiency to enhance its competitiveness relative to existing SCF codes: (i) methods such as higher order finite element methods, adaptive mesh methods, and efficient message passing interface parallel implementation would be considered in our method, and results towards numerical efficiency will be reported in our forthcoming works; (ii) given the limited parallel scalability of multigrid methods, a sufficiently well scaling parallel implementation of the multigrid preconditioner will be a key issue to efficient large-scale calculations in practice.

Acknowledgments. We are grateful to the referees for insightful comments and suggestions which helped to improve the readability and quality of this work.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 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] R. Alizadegan, K. J. Hsia, and T. Martinez, *A divide and conquer real space finite-element Hartree–Fock method*, J. Chem. Phys., 132 (2010), 034101, https://doi.org/10.1063/1.3290949.

[4] 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.

[5] 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.

[6] P. Batcho, *Computational method for general multicenter electronic structure calculations*, Phys. Rev. E, 61 (2000), p. 7169, https://doi.org/10.1103/PhysRevE.61.7169.

[7] 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.

[8] D. R. Bowler and T. Miyazaki, *$\mathcal{O}(N)$ methods in electronic structure calculations*, Rep. Progr. Phys., 75 (2012), p. 036503, https://doi.org/10.1088/0034-4885/75/3/036503.

[9] E. J. Bylaska, M. Holst, and J. H. Weare, *Adaptive finite element method for solving the exact Kohn–Sham equation of density functional theory*, J. Chem. Theory Comput., 5 (2009), pp. 937–948, https://doi.org/10.1021/ct800350j.

[10] H. Chen, X. Dai, X. Gong, L. He, and A. Zhou, *Adaptive finite element approximations for Kohn–Sham models*, Multiscale Model. Simul., 12 (2014), pp. 1828–1869, https://doi.org/10.1137/130916096.

[11] A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, G. N. Miranda, and J. W. Ruge, *Robustness and scalability of algebraic multigrid*, SIAM J. Sci. Comput., 21 (2000), pp. 1886–1908, https://doi.org/10.1137/S1064827598339402.

[12] 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.

[13] 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.

[14] D. Davydov, T. D. Young, and P. Steinmann, *On the adaptive finite element analysis of the Kohn–Sham equations: Methods, algorithms, and implementation*, Internat J. Numer. Methods Engrg., 106 (2016), pp. 863–888, https://doi.org/10.1002/nme.5140.

[15] *The Elk Code*, http://elk.sourceforge.net/.

[16] J. T. Frey and D. J. Doren, *TubeGen* 3.4, 2011, http://turin.nss.udel.edu/research/tubegenonline.html (accessed 2019-11-01).

[17] M. J. Frisch, G. W. Trucks, H. B. Schlegel, et al. *Gaussian* 16 *Revision* B.01, 2016. Gaussian Inc. Wallingford CT, https://gaussian.com/dft/.

[18] 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.

[19] 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.

[20] 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), 014109, https://doi.org/10.1063/1.2949547.

[21] C. Geuzaine and J.-F. Remacle, *Gmsh: A* 3*-D finite element mesh generator with built-in pre-and post-processing facilities*, Internat J. Numer. Methods Engrg., 79 (2009), pp. 1309–1331, https://doi.org/10.1002/nme.2579.

[22] P. Giannozzi, S. Baroni, N. Bonini, et al., *QUANTUM ESPRESSO: A modular and open-source software project for quantum simulations of materials*, J. phys. Condensed Matter, 21 (2009), 395502, https://doi.org/10.1088/0953-8984/21/39/395502.

[23] X. Gonze, J.-M. Beuken, R. Caracas, et al., *First-principles computation of material properties: The ABINIT software project*, Comput. Mater. Sci., 25 (2002), pp. 478–492, https://doi.org/10.1016/S0927-0256(02)00325-7

[24] J. Hafner, *Ab-initio simulations of materials using VASP: Density-functional theory and beyond*, J. comput. chem., 29 (2008), pp. 2044–2078, https://doi.org/10.1002/jcc.21057

[25] D. Hamann, *Optimized norm-conserving Vanderbilt pseudopotentials*, Phys. Rev. B, 88 (2013), 085117, https://doi.org/10.1103/PhysRevB.88.085117.

[26] W. Huang and R. D. Russell, *Adaptive Moving Mesh Methods*, Appl. Math. Sci. 174, Springer Science & Business Media, New York, 2010, https://doi.org/10.1007/978-1-4419-7916-2.

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

[28] B. Kanungo and V. Gavini, *Large-scale all-electron density functional theory calculations using an enriched finite-element basis*, Phys. Rev. B, 95 (2017), 035112, https://doi.org/10.1103/PhysRevB.95.035112.

[29] 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.

[30] 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.

[31] 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.

[32] Y. Kuang and G. Hu, *On stabilizing and accelerating SCF using ITP in solving Kohn–Sham equation*, Commun. Comput. Phys., 28 (2020), pp. 999–1018, https://doi.org/10.4208/cicp.OA-2019-0024.

[33] L. Lehtovaara, V. Havu, and M. Puska, *All-electron density functional theory and time-dependent density functional theory with high-order finite elements*, J. Chem. Phys, 131 (2009), 054103, https://doi.org/10.1063/1.3176508.

[34] 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. Condensed Matter, 22 (2010), 505501, https://doi.org/10.1088/0953-8984/22/50/505501.

[35] 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.

[36] 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.

[37] 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.

[38] Y. Maday, *h-P finite element approximation for full-potential electronic structure calculations*, Chin. Ann. Math. Ser. B, 35 (2014), pp. 1–24, https://doi.org/10.1007/s11401-013-0819-3.

[39] 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.

[40] P. Motamarri, S. Das, S. Rudraraju, K. Ghosh, D. Davydov, and V. Gavini, *DFT-FE–A massively parallel adaptive finite-element code for large-scale density functional theory calculations*, Comput. Phys. Commun., 246 (2020), 106853, https://doi.org/10.1016/j.cpc.2019.07.016.

[41] P. Motamarri, M. R. Nowak, K. Leiter, J. Knap, and V. Gavini, *Higher-order adaptive finite-element methods for Kohn–Sham density functional theory*, J. Comput. Phys., 253 (2013), pp. 308–343, https://doi.org/10.1016/j.jcp.2013.06.042.

[42] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Science & Business Media, New York, 2006, https://doi.org/10.1007/978-0-387-40065-5.

[43] 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. Modern Phys., 64 (1992), p. 1045, https://doi.org/10.1103/RevModPhys.64.1045.

[44] J. P. Perdew and A. Zunger, *Self-interaction correction to density-functional approximations for many-electron systems*, Phys. Rev. B, 23 (1981), p. 5048, https://doi.org/10.1103/PhysRevB.23.5048.

[45] 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.

[46] N. D. Rufus, B. Kanungo, and V. Gavini, *Fast and robust all-electron density functional theory calculations in solids using orthogonalized enriched finite elements*, Phys. Rev. B, 104 (2021), 085112.

[47] V. Schauer and C. Linder, *All-electron Kohn–Sham density functional theory on hierarchic finite element spaces*, J. Comput. Phys., 250 (2013), pp. 644–664, https://doi.org/10.1016/j.jcp.2013.04.020.

[48] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, et al., *General atomic and molecular electronic structure system*, J. comput. chem., 14 (1993), pp. 1347–1363, https://doi.org/10.1002/jcc.540141112

[49] K. Schwarz and P. Blaha, *Solid state calculations using WIEN2k*, Comput. Mater. Sci., 28 (2003), pp. 259–273, https://doi.org/10.1016/S0927-0256(03)00112-5

[50] 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.

[51] 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.

[52] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, and W.A. De Jong, *NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations*, Comput. Phys. Commun., 181 (2010), pp. 1477–1489, https://doi.org/10.1016/j.cpc.2010.04.018.

[53] S. R. White, J. W. Wilkins, and M. P. Teter, *Finite-element method for electronic structure*, Phys. Rev. B, 39 (1989), p. 5819, https://doi.org/10.1103/PhysRevB.39.5819.

[54] 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.

[55] 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.

[56] S. Zhang, A. Lazicki, B. Militzer, L. H. Yang, K. Caspersen, J. A. Gaffney, M. W. Däne, J. E. Pask, W. R. Johnson, A. Sharma, et al., *Equation of state of boron nitride combining computation, modeling, and experiment*, Phys. Rev. B, 99 (2019), 165103, https://doi.org/10.1103/PhysRevB.99.165103.

[57] 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.