

STRUCTURED QUASI-NEWTON METHODS FOR OPTIMIZATION WITH ORTHOGONALITY CONSTRAINTS

JIANG HU*, BO JIANG[†], LIN LIN[‡], ZAIWEN WEN[§], AND YAXIANG YUAN[¶]

Abstract. In this paper, we study structured quasi-Newton methods for optimization problems with orthogonality constraints. Note that the Riemannian Hessian of the objective function requires both the Euclidean Hessian and the Euclidean gradient. In particular, we are interested in applications that the Euclidean Hessian itself consists of a computational cheap part and a significantly expensive part. Our basic idea is to keep these parts of lower computational costs but substitute those parts of higher computational costs by the limited-memory quasi-Newton update. More specially, the part related to Euclidean gradient and the cheaper parts in the Euclidean Hessian are preserved. The initial quasi-Newton matrix is further constructed from a limited-memory Nyström approximation to the expensive part. Consequently, our subproblems approximate the original objective function in the Euclidean space and preserve the orthogonality constraints without performing the so-called vector transports. When the subproblems are solved to sufficient accuracy, both global and local q-superlinear convergence can be established under mild conditions. Preliminary numerical experiments on the linear eigenvalue problem and the electronic structure calculation show the effectiveness of our method compared with the state-of-art algorithms.

Key words. optimization with orthogonality constraints, structured quasi-Newton method, limited-memory Nyström approximation, Hartree-Fock total energy minimization, convergence.

AMS subject classifications. 15A18, 65K10, 65F15, 90C26, 90C30

1. Introduction. In this paper, we consider the optimization problem with orthogonality constraints:

$$(1.1) \quad \min_{X \in \mathbb{C}^{n \times p}} f(X) \quad \text{s.t.} \quad X^* X = I_p,$$

where $f(X) : \mathbb{C}^{n \times p} \rightarrow \mathbb{R}$ is a \mathbb{R} -differentiable function [26]. Although our proposed methods are applicable to a general function $f(X)$, we are in particular interested in the cases that the Euclidean Hessian $\nabla^2 f(X)$ takes a natural structure as

$$(1.2) \quad \nabla^2 f(X) = \mathcal{H}^c(X) + \mathcal{H}^e(X),$$

where the computational cost of $\mathcal{H}^e(X)$ is much more expensive than that of $\mathcal{H}^c(X)$. This situation occurs when f is a summation of functions whose full Hessian are

*Beijing International Center for Mathematical Research, Peking University, China (jianghu@pku.edu.cn).

[†]School of Mathematical Sciences, Key Laboratory for NSLSCS of Jiangsu Province, Nanjing Normal University, China (jiangbo@njnu.edu.cn). Research supported in part by NSFC grants 11501298 and 11671036, and by the NSF of Jiangsu Province (BK20150965).

[‡]Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720 and Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (linlin@math.berkeley.edu). Research supported in part by the National Science Foundation under Grant No. DMS-1652330, the Department of Energy under Grants No. DE-SC0017867 and No. DE-AC02-05CH11231, and the SciDAC project.

[§]Beijing International Center for Mathematical Research, Peking University, China (wenzw@pku.edu.cn). Research supported in part by NSFC grants 11831002, 11421101 and 91730302, and by the National Basic Research Project under grant 2015CB856002.

[¶]State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China (yyx@lsec.cc.ac.cn). Research supported in part by NSFC grants 11331012 and 11461161005.

expensive to be evaluated or even not accessible. A practical example is the Hartree-Fock-like total energy minimization problem in electronic structure theory [38, 31], where the computation cost associated with the Fock exchange matrix is significantly larger than the cost of the remaining components.

There are extensive methods for solving (1.1) in the literature. By exploring the geometry of the manifold (i.e., orthogonality constraints), the Riemannian gradient descent, conjugate gradient (CG), Newton and trust-region methods are proposed in [11, 10, 41, 36, 1, 2, 44]. Since the second-order information sometimes is not available, the quasi-Newton type method serves as an alternative method to guarantee the good convergence property. Different from the Euclidean quasi-Newton method, the vector transport operation [2] is used to compare tangent vectors in different tangent spaces. After obtaining a descent direction, the so-called retraction provides a curvilinear search along manifold. By adding some restrictions between differentiable retraction and vector transport, a Riemannian Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is presented in [33, 34, 35]. Due to the requirement of differentiable retraction, the computational cost associated with the vector transport operation may be costly. To avoid this disadvantage, authors in [18, 21, 23, 20] develop a new class of Riemannian BFGS methods, symmetric rank-one (SR1) and Broyden family methods. Moreover, a selection of Riemannian quasi-Newton methods has been implemented in the software package Manopt [6] and ROPTLIB [19].

1.1. Our contribution. Since the set of orthogonal matrices can be viewed as the Stiefel manifold, the existing quasi-Newton methods focus on the construction of an approximation to the Riemannian Hessian $\text{Hess } f(X)$:

$$(1.3) \quad \text{Hess } f(X)[\xi] = \text{Proj}_X(\nabla^2 f(X)[\xi] - \xi \text{sym}(X^* \nabla f(X))),$$

where ξ is any tangent vector in the tangent space $T_X := \{\xi \in \mathbb{C}^{n \times p} : X^* \xi + \xi^* X = 0\}$ and $\text{Proj}_X(Z) := Z - X \text{sym}(X^* Z)$ is the projection of Z onto the tangent space T_X and $\text{sym}(A) := (A + A^*)/2$. See [3] for details on the structure (1.3). We briefly summarize our contributions as follows.

- By taking the advantage of this structure (1.3), we construct an approximation to Euclidean Hessian $\nabla^2 f(X)$ instead of the full Riemannian Hessian $\text{Hess } f(X)$ directly, but keep the remaining parts $\xi \text{sym}(X^* \nabla f(X))$ and $\text{Proj}_X(\cdot)$. Then, we solve a subproblem with orthogonality constraints, whose objective function uses an approximate second-order Taylor expansion of f with an extra regularization term. Similar to [16], the trust-region-like strategy for the update of the regularization parameter and the modified CG method for solving the subproblem are utilized. The vector transport is not needed in since we are working in the ambient Euclidean space.
- By further taking advantage of the structure (1.2) of f , we develop a structured quasi-Newton approach to construct an approximation to the expensive part \mathcal{H}^e while preserving the cheap part \mathcal{H}^c . This kind of structured approximation usually yields a better property than the approximation constructed by the vanilla quasi-Newton method. For the construction of an initial approximation of \mathcal{H}^e , we also investigate a limited-memory Nyström approximation, which gives a subspace approximation of a known good but

still complicated approximation of \mathcal{H}^e .

- When the subproblems are solved to certain accuracy, both global and local q-superlinear convergence can be established under certain mild conditions.
- Applications to the linear eigenvalue problem and the electronic structure calculation are presented. The proposed algorithms perform comparably well with state-of-art methods in these two applications.

1.2. Applications to electronic structure calculation. Electronic structure theories, and particularly Kohn-Sham density functional theory (KSDFT), play an important role in quantum physics, quantum chemistry and materials science. This problem can be interpreted as a minimization problem for the electronic total energy over multiple electron wave functions which are orthogonal to each other. The mathematical structure of Kohn-Sham equations depends heavily on the choice of the exchange-correlation (XC) functional. With some abuse of terminology, throughout the paper we will use KSDFT to refer to Kohn-Sham equations with local or semi-local exchange-correlation functionals. Before discretization, the corresponding Kohn-Sham Hamiltonian is a differential operator. On the other hand, when hybrid exchange-correlation functionals [4, 15] are used, the Kohn-Sham Hamiltonian becomes an integro-differential operator, and the Kohn-Sham equations become Hartree-Fock-like equations. Again with some abuse of terminology, we will refer to such calculations as the HF calculation.

For KSDFT calculations, the most popular numerical scheme is the self-consistent field (SCF) iteration which can be efficient when combined with certain charge mixing techniques. Since the hybrid exchange-correlation functionals depend on all the elements of the density matrix, HF calculations are usually more difficult than KSDFT calculations. One commonly used algorithm is called the nested two-level SCF method [12]. In the inner SCF loop, by fixing the density matrix and the hybrid exchange operator, it only performs an update on the charge density ρ , which is solved by the SCF iteration. Once the stopping criterion of the inner iteration is satisfied, the density matrix is updated in the outer loop according to the Kohn-Sham orbitals computed in the inner loop. This method can also utilize the charge mixing schemes for the inner SCF loop to accelerate convergence. Recently, by combining with the adaptively compressed exchange operator (ACE) method [28], the convergence rate of the nested two-level SCF method is greatly improved. Another popular algorithm to solve HF calculations is the commutator direction inversion of the iterative subspace (C-DIIS) method. By storing the density matrix explicitly, it can often lead to accelerated convergence rate. However, when the size of the density matrix becomes large, the storage cost of the density matrix becomes prohibitively expensive. Thus Lin et al. [17] proposed the projected C-DIIS (PC-DIIS) method, which only requires storage of wave function type objects instead of the whole density matrix.

HF calculations can be also solved via using the aforementioned Riemannian optimization methods (e.g., a feasible gradient method on the Stiefel manifold [44]) without storing the density matrix or the wave function. However, these existing methods often do not use the structure of the Hessian in KSDFT or HF calculations. In this paper, by exploiting the structure of the Hessian, we apply our structured quasi-Newton method to solve these problems. Preliminary numerical experiments

show that our algorithm performs at least comparably well with state-of-art methods in their convergent case. In the case that state-of-art methods failed, our algorithm often returns high quality solutions.

1.3. Organization. This paper is organized as follows. In section 2, we introduce our structured quasi-Newton method and present our algorithm. In section 3, the global and local convergence is analyzed under certain inexact conditions. In sections 4 and 5, detailed applications to the linear eigenvalue problem and the electronic structure calculation are discussed. Finally, we demonstrate the efficiency of our proposed algorithm in section 6.

1.4. Notation. For a matrix $X \in \mathbb{C}^{n \times p}$, we use \bar{X} , X^* , $\Re X$ and $\Im X$ to denote its complex conjugate, complex conjugate transpose, real and imaginary parts, respectively. Let $\text{span}\{X_1, \dots, X_l\}$ be the space spanned by the matrices X_1, \dots, X_l . The vector denoted $\text{vec}(X)$ in \mathbb{C}^{np} is formulated by stacking each column of X one by one, from the first to the last column; the operator $\text{mat}(\cdot)$ is the inverse of $\text{vec}(\cdot)$, i.e., $\text{mat}(\text{vec}(X)) = X$. Given two matrices $A, B \in \mathbb{C}^{n \times p}$, the Frobenius inner product $\langle \cdot, \cdot \rangle$ is defined as $\langle A, B \rangle = \text{tr}(A^* B)$ and the corresponding Frobenius norm $\|\cdot\|_F$ is defined as $\|A\|_F = \sqrt{\text{tr}(A^* A)}$. The Hadamard product of A and B is $A \odot B$ with $(A \odot B)_{ij} = A_{ij} B_{ij}$. For a matrix $M \in \mathbb{C}^{n \times n}$, the operator $\text{diag}(M)$ is a vector in \mathbb{C}^n formulated by the main diagonal of M ; and for $c \in \mathbb{C}^n$, the operator $\text{Diag}(c)$ is an n -by- n diagonal matrix with the elements of c on the main diagonal. The notation I_p denotes the p -by- p identity matrix. Let $\text{St}(n, p) := \{X \in \mathbb{C}^{n \times p} : X^* X = I_p\}$ be the (complex) Stiefel manifold. The notation \mathbb{N} refers to the set of all natural numbers.

2. A structured quasi-Newton approach.

2.1. Structured quasi-Newton subproblem. In this subsection, we develop the structured quasi-Newton subproblem for solving (1.1). Based on the assumption (1.2), methods using the exact Hessian $\nabla^2 f(X)$ may not be the best choices. When the computational cost of the gradient $\nabla f(X)$ is significantly cheaper than that of the Hessian $\nabla^2 f(X)$, the quasi-Newton methods, which mainly use the gradients $\nabla f(X)$ to construct an approximation to $\nabla^2 f(X)$, may outperform other methods. Considering the form (1.2), we can construct a structured quasi-Newton approximation \mathcal{B}^k for $\nabla^2 f(X^k)$. The details will be presented in section 2.2. Note that a similar idea has been presented in [47] for the unconstrained nonlinear least square problems [24, 37]. Then our subproblem at the k -th iteration is constructed as

$$(2.1) \quad \min_{X \in \mathbb{C}^{n \times p}} m_k(X) \quad \text{s.t.} \quad X^* X = I,$$

where

$$m_k(X) := \Re \langle \nabla f(X^k), X - X^k \rangle + \frac{1}{2} \Re \langle \mathcal{B}^k [X - X^k], X - X^k \rangle + \frac{\tau_k}{2} d(X, X^k)$$

is an approximation to $f(X)$ in the Euclidean space. For the second-order Taylor expansion of $f(X)$ at a point X^k , we refer to [43, section 1.1] for details. Here, τ_k is a regularization parameter and $d(X, X^k)$ is a proximal term to guarantee the convergence.

The proximal term can be chosen as the quadratic regularization

$$(2.2) \quad d(X, X^k) = \|X - X^k\|_{\mathbb{F}}^2$$

or the cubic regularization

$$(2.3) \quad d(X, X^k) = \frac{2}{3} \|X - X^k\|_{\mathbb{F}}^3.$$

In the following, we will mainly focus on the quadratic regularization (2.2). Due to the Stiefel manifold constraint, the quadratic regularization (2.2) is actually equivalent to the linear term $-2\Re \langle X, X^k \rangle$. By using the Riemannian Hessian formulation (1.3) on the Stiefel manifold, we have

$$(2.4) \quad \text{Hess } m_k(X^k)[\xi] = \text{Proj}_{X^k} (\mathcal{B}^k[\xi] - \xi \text{sym}((X^k)^* \nabla f(X^k))) + \tau_k \xi, \quad \xi \in T_{X^k}.$$

Hence, the regularization term is to shift the spectrum of the corresponding Riemannian Hessian of the approximation \mathcal{B}^k with τ_k .

The Riemannian quasi-Newton methods for (1.1) in the literature [19, 21, 22, 23] focus on constructing an approximation to the Riemannian Hessian $\text{Hess } f(X^k)$ directly without using its special structure (1.3). Therefore, vector transport needs to be utilized to transport the tangent vectors from different tangent spaces to one common tangent space. If $p \ll n$, the second term $\text{sym}((X^k)^* \nabla f(X^k))$ is a small-scaled matrix and thus can be computed with low cost. In this case, after computing the approximation $\mathcal{B}^k[\xi]$ of $\nabla^2 f(X)[\xi]$, we obtain a structured Riemannian quasi-Newton approximation $\text{Proj}_{X^k} (\mathcal{B}^k[\xi] - \xi \text{sym}((X^k)^* \nabla f(X^k)))$ of $\text{Hess } f(X^k)[\xi]$ without using any vector transport.

2.2. Construction of \mathcal{B}^k . The classical quasi-Newton methods construct the approximation \mathcal{B}^k such that it satisfies the secant condition

$$(2.5) \quad \mathcal{B}^k[S^k] = \nabla f(X^k) - \nabla f(X^{k-1}),$$

where $S^k := X^k - X^{k-1}$. Noticing that $\nabla^2 f(X)$ takes the natural structure (1.2), it is reasonable to keep the cheaper part $\mathcal{H}^c(X)$ while only to approximate $\mathcal{H}^e(X)$. Specifically, we derive the approximation \mathcal{B}^k to the Hessian $\nabla^2 f(X^k)$ as

$$(2.6) \quad \mathcal{B}^k = \mathcal{H}^c(X^k) + \mathcal{E}^k,$$

where \mathcal{E}^k is an approximation to $\mathcal{H}^e(X^k)$. Substituting (2.6) into (2.5), we can see that the approximation \mathcal{E}^k should satisfy the following revised secant condition

$$(2.7) \quad \mathcal{E}^k[S^k] = Y^k,$$

where

$$(2.8) \quad Y^k := \nabla f(X^k) - \nabla f(X^{k-1}) - \mathcal{H}^c(X^k)[S^k].$$

For the large scale optimization problems, the limited-memory quasi-Newton methods are preferred since they often make simple but good approximations of the exact Hessian. Considering that the part $\mathcal{H}^e(X^k)$ itself may not be positive definite

even when X^k is optimal, we utilize the limited-memory symmetric rank-one (LSR1) scheme to approximate $\mathcal{H}^e(X^k)$ such that it satisfies the secant equation (2.7).

Let $l = \min\{k, m\}$. We define the $(np) \times l$ matrices $S^{k,m}$ and $Y^{k,m}$ by

$$S^{k,m} = [\text{vec}(S^{k-l}), \dots, \text{vec}(S^{k-1})], \quad Y^{k,m} = [\text{vec}(Y^{k-l}), \dots, \text{vec}(Y^{k-1})].$$

Let $\mathcal{E}_0^k : \mathbb{C}^{n \times p} \rightarrow \mathbb{C}^{n \times p}$ be the initial approximation of $\mathcal{H}^e(X^k)$ and define the $(np) \times l$ matrix $\Sigma^{k,m} := [\text{vec}(\mathcal{E}_0^k[S^{k-l}]), \dots, \text{vec}(\mathcal{E}_0^k[S^{k-1}])]$. Let $F^{k,m}$ be a matrix in $\mathbb{C}^{l \times l}$ with $(F^{k,m})_{i,j} = \langle S^{k-l+i-1}, Y^{k-l+j-1} \rangle$ for $1 \leq i, j \leq l$. Under the assumption that $\langle S^j, \mathcal{E}^j[S^j] - Y^j \rangle \neq 0$, $j = k-l, \dots, k-1$, it follows from [9, Theorem 5.1] that the matrix $F^{k,m} - (S^{k,m})^* \Sigma^{k,m}$ is invertible and the LSR1 gives

$$(2.9) \quad \mathcal{E}[U] = \mathcal{E}_0^k[U] + \text{mat} \left(N^{k,m} (F^{k,m} - (S^{k,m})^* \Sigma^{k,m})^{-1} (N^{k,m})^* \text{vec}(U) \right),$$

where $U \in \mathbb{C}^{n \times p}$ is any direction and $N^{k,m} = Y^{k,m} - \Sigma^{k,m}$. In the practical implementation, we skip the update if

$$|\langle S^j, \mathcal{E}^j[S^j] - Y^j \rangle| \leq r \|S^j\|_{\text{F}} \|\mathcal{E}^j[S^j] - Y^j\|_{\text{F}}$$

with small number r , say $r = 10^{-8}$. Similar idea can be found in [32].

2.3. Limited-memory Nyström approximation of \mathcal{E}_0^k . A good initial guess to the exact Hessian is also important to accelerate the convergence of the limited-memory quasi-Newton method. Here, we assume that a good initial approximation \mathcal{E}_0^k of the expensive part of the Hessian $\mathcal{H}^e(X^k)$ is known but its computational cost is still very high. We conduct how to use the limited-memory Nyström approximation to construct another approximation with lower computational cost based on \mathcal{E}_0^k .

Specially, let \mathfrak{S} be a matrix whose columns form an orthogonal basis of a well-chosen subspace \mathfrak{S} and denote $W = \mathcal{E}_0^k[\Omega]$. To reduce the computational cost and keep the good property of \mathcal{E}_0^k , we construct the following approximation

$$(2.10) \quad \hat{\mathcal{E}}_0^k[U] := W(W^* \Omega)^\dagger W^* U,$$

where $U \in \mathbb{C}^{n \times p}$ is any direction. This is called the limited-memory Nyström approximation; see [40] and references therein for more details. By choosing the dimension of the subspace \mathfrak{S} properly, the rank of $W(W^* \Omega)^\dagger W^*$ can be small enough such that the computational cost of $\hat{\mathcal{E}}_0^k[U]$ is significantly reduced. Furthermore, we still want $\hat{\mathcal{E}}_0^k$ to satisfy the secant condition (2.7) as \mathcal{E}_0^k does. More specifically, we need to seek the subspace \mathfrak{S} such that the secant condition

$$\hat{\mathcal{E}}_0^k[S^k] = Y^k$$

holds. To this aim, the subspace \mathfrak{S} can be chosen as

$$\text{span}\{X^{k-1}, X^k\},$$

which contains the element S^k . By assuming that $\mathcal{E}_0^k[UV] = \mathcal{E}_0^k[U]V$ for any matrices U, V with proper dimension (this condition is satisfied when \mathcal{E}_0^k is a matrix), we have

$\hat{\mathcal{E}}_0^k$ will satisfy the secant condition whenever \mathcal{E}_0^k does. From the methods for linear eigenvalue computation in [25] and [30], the subspace \mathfrak{S} can also be decided as

$$(2.11) \quad \text{span}\{X^{k-1}, X^k, \mathcal{E}_0^k[X^k]\} \quad \text{or} \quad \text{span}\{X^{k-h}, \dots, X^{k-1}, X^k\}$$

with small memory length h . Once the subspace is defined, we can obtain the limited-memory Nyström approximation by computing the $\mathcal{E}_0^k[\Omega]$ once and the pseudo inverse of a small scale matrix.

2.4. A structured quasi-Newton method with subspace refinement.

Based on the theory of quasi-Newton method for unconstrained optimization, we know that algorithms which set the solution of (2.1) as the next iteration point may not converge if no proper requirements on approximation \mathcal{B}^k or the regularization parameter τ_k . Hence, we update the regularization parameter here using a trust-region-like strategy. Referring to [16], we compute a trial point Z^k by utilizing a modified CG method to solve the subproblem inexactly, which is to solve the Newton equation of (2.1) at X^k as

$$(2.12) \quad \text{grad} m_k(X^k) + \text{Hess} m_k(X^k)[\xi] = 0, \quad \xi \in T_{X^k},$$

where $\text{grad} m_k(X^k) = \text{grad} f(X^k)$ and $\text{Hess} m_k(X^k)$ are given in (2.4). After obtaining the trial point Z^k of (2.1), we calculate the ratio between the predicted reduction and the actual reduction

$$(2.13) \quad r_k = \frac{f(Z^k) - f(X^k)}{m_k(Z^k)}.$$

If $r_k \geq \eta_1 > 0$, then the iteration is successful and we set $X^{k+1} = Z^k$; otherwise, the iteration is unsuccessful and we set $X^{k+1} = X^k$, that is,

$$(2.14) \quad X^{k+1} = \begin{cases} Z^k, & \text{if } r_k \geq \eta_1, \\ X^k, & \text{otherwise.} \end{cases}$$

The regularization parameter τ_{k+1} is updated as

$$(2.15) \quad \tau_{k+1} \in \begin{cases} (0, \gamma_0 \tau_k], & \text{if } r_k \geq \eta_2, \\ [\tau_k, \gamma_1 \tau_k], & \text{if } \eta_1 \leq r_k < \eta_2, \\ [\gamma_1 \tau_k, \gamma_2 \tau_k], & \text{otherwise,} \end{cases}$$

where $0 < \eta_1 \leq \eta_2 < 1$ and $0 < \gamma_0 < 1 < \gamma_1 \leq \gamma_2$. These parameters determine how aggressively the regularization parameter is decreased when an iteration is successful or it is increased when an iteration is unsuccessful. In practice, the performance of the regularized trust-region algorithm is not very sensitive to the values of the parameters.

Noticing that the Newton-type method may still be very slow when the Hessian is close to be singular [8]. Numerically, it may happen that the regularization parameter turns to be huge and the Riemannian Newton direction is nearly parallel to the negative gradient direction. Hence, it leads to an update X^{k+1} belonging to the subspace $\tilde{\mathfrak{S}}^k := \text{span}\{X^k, \text{grad} f(X^k)\}$, which is similar to the Riemannian gradient approach.

To overcome this issue, we propose an optional step of solving (1.1) restricted to a subspace. Specifically, at X^k , we construct a subspace \mathfrak{S}^k with an orthogonal basis $Q^k \in \mathbb{C}^{n \times q}$ ($p \leq q \leq n$), where q is the dimension of \mathfrak{S}^k . Then any point X in the subspace \mathfrak{S}^k can be represented by

$$X = Q^k M$$

for some $M \in \mathbb{C}^{q \times p}$. Similar to the constructions of linear eigenvalue problems in [25] and [30], the subspace \mathfrak{S}^k can be decided by using the history information $\{X^k, X^{k-1}, \dots\}$, $\{\text{grad} f(X^k), \text{grad} f(X^{k-1}), \dots\}$ and other useful information. Given the subspace \mathfrak{S}^k , the subspace method aims to find a solution of (1.1) with an extra constraint $X \in \mathfrak{S}^k$, namely,

$$(2.16) \quad \min_{M \in \mathbb{C}^{q \times p}} f(Q^k M) \quad \text{s.t.} \quad M^* M = I_p.$$

The problem (2.16) can be solved inexactly by existing methods for optimization with orthogonality constraints. Once a good approximate solution M^k of (2.16) is obtained, then we update $X^{k+1} = Q^k M^k$ which is an approximate minimizer in the subspace \mathfrak{S}^k instead of $\tilde{\mathfrak{S}}^k$. This completes one step of the subspace iteration. In fact, we compute the ratios between the norms of the Riemannian gradient of the last few iterations. If all of these ratios are almost 1, we infer that the current iterates stagnates and the subspace method is called. Consequently, our algorithm framework is outlined in Algorithm 1.

Algorithm 1: A structured quasi-Newton method with subspace refinement

Input initial guess $X^0 \in \mathbb{C}^{n \times p}$ with $(X^0)^* X^0 = I_p$ and the memory length m .

Choose $\tau_0 > 0$, $0 < \eta_1 \leq \eta_2 < 1$, $1 < \gamma_1 \leq \gamma_2$. Set $k = 0$.

while *stopping conditions not met* **do**

 Choose \mathcal{E}_0^k (use the limited-memory Nyström approximation if necessary).

 Construct the approximation \mathcal{B}^k via (2.6) and (2.9).

 Construct the subproblem (2.1) and use the modified CG method (Algorithm 2 in [16]) to compute a new trial point Z^k .

 Compute the ratio r_k via (2.13).

 Update X^{k+1} from the trial point Z^k based on (2.14).

 Update τ_k according to (2.15).

$k \leftarrow k + 1$.

if *stagnate conditions met* **then**

 └ Solve the subspace problem (2.16) to update X^{k+1} .

3. Convergence analysis. In this section, we present the convergence property of Algorithm 1. To guarantee the global convergence and fast local convergence rate, the inexact conditions for the subproblem (2.1) (with quadratic or cubic regularization) can be chosen as

$$(3.1) \quad m_k(Z^k) \leq -c \|\text{grad} f(X^k)\|_{\mathbb{F}}^2$$

$$(3.2) \quad \|\text{grad} m_k(Z^k)\|_{\mathbb{F}} \leq \theta^k \|\text{grad} f(X^k)\|_{\mathbb{F}}$$

with some positive constant c and $\theta^k := \min\{1, \|\text{grad}f(X^k)\|_{\mathbb{F}}\}$. Here, the inequality (3.1) is to guarantee the global convergence and the inequality (3.2) leads to fast local convergence. Throughout the analysis of convergence, we assume that the stagnate conditions are never met. (In fact, a sufficient decrease for the original problem in each iteration can be guaranteed from the description of subspace refinement. Hence, the global convergence still holds.)

3.1. Global convergence. Since the regularization term is used, the global convergence of our method can be obtained by assuming the boundedness on the constructed Hessian approximation. We first make the following assumptions.

ASSUMPTION 1. *Let $\{X^k\}$ be generated by Algorithm 1 without subspace refinement. We assume:*

(A1) *The gradient ∇f is Lipschitz continuous on the convex hull of $\text{St}(n, p)$, i.e., there exists $L_f > 0$ such that*

$$\|\nabla f(X) - \nabla f(Y)\|_{\mathbb{F}} \leq L_f \|X - Y\|_{\mathbb{F}}, \quad \forall X, Y \in \text{conv}(\text{St}(n, p)).$$

(A2) *There exists $\kappa_H > 0$ such that $\|\mathcal{B}^k\| \leq \kappa_H$ for all $k \in \mathbb{N}$, where $\|\cdot\|$ is the operator norm introduced by the Euclidean inner product.*

REMARK 2. *By Assumption (A1), $\nabla f(X)$ is uniformly bounded by some constant κ_g on the compact set $\text{conv}(\text{St}(n, p))$, i.e.,*

$$\|\nabla f(X)\|_{\mathbb{F}} \leq \kappa_g, \quad X \in \text{conv}(\text{St}(n, p)).$$

Assumption (A2) is often used in the traditional symmetric rank-1 method [7] which appears to be reasonable in practice.

Based on the similar proof in [16, 43], we have the following theorem for global convergence.

THEOREM 3. *Suppose that Assumptions (A1)-(A2) and the inexact conditions (3.1) hold. Then, either*

$$(3.3) \quad \text{grad}f(X^t) = 0 \text{ for some } t > 0 \quad \text{or} \quad \lim_{k \rightarrow \infty} \|\text{grad}f(X^k)\|_{\mathbb{F}} = 0.$$

Proof. For the quadratic regularization (2.2), let us note that the Riemannian Hessian $\text{Hess}m(X^k)$ can be guaranteed to be bounded from Assumption 1. In fact, from (2.4), we have

$$\|\text{Hess}m_k(X^k)\| \leq \|\mathcal{B}^k\| + \|X^k\| \|\nabla f(X^k)\|_{\mathbb{F}} + \tau_k \leq \kappa_H + \kappa_g + \tau_k,$$

where $\|X^k\| = 1$ because of its unitary property. Hence, we can guarantee that the direction obtained from the modified CG method is a descent direction via similar techniques in [16, Lemma 7]. Then the convergence of the iterates $\{X^k\}$ can be proved in a similar way by following the details in [16] for the quadratic regularization. As to the cubic regularization, we can refer [43, Theorem 4.9] for a similar proof. \square

3.2. Local convergence. We now focus on the local convergence with the inexact conditions (3.1) and (3.2). We make some necessary assumptions below.

ASSUMPTION 4. Let $\{X^k\}$ be the sequence generated by Algorithm 1 without sub-space refinement. We assume

- (B1) The sequence $\{X^k\}$ converges to X_* with $\text{grad} f(X_*) = 0$.
- (B2) The Euclidean Hessian $\nabla^2 f$ is continuous on $\text{conv}(\text{St}(n, p))$.
- (B3) The Riemannian Hessian $\text{Hess} f(X)$ is positive definite at X_* .
- (B4) The Hessian approximation \mathcal{B}^k satisfies

$$(3.4) \quad \frac{\|(\mathcal{B}^k - \nabla^2 f(X^k))[Z^k - X^k]\|_{\mathbb{F}}}{\|Z^k - X^k\|_{\mathbb{F}}} \rightarrow 0, \quad k \rightarrow \infty.$$

Following the proof in [16, Lemma 17], we show that all iterations are eventually very successful (i.e., $r_k \geq \eta_2$, for all sufficiently large k) when Assumptions (B1)-(B4) and the inexact conditions (3.1) and (3.2) hold.

LEMMA 5. Let Assumptions (B1)-(B4) be satisfied. Then, all iterations are eventually very successful.

Proof. From the second-order Taylor expansion, we have

$$f(Z^k) - f(X^k) - m_k(Z^k) \leq \frac{1}{2} \Re \langle (\nabla^2 f(X_\delta^k) - \mathcal{B}^k)[Z^k - X^k], Z^k - X^k \rangle,$$

for some suitable $\delta_k \in [0, 1]$ and $X_\delta^k := X^k + \delta_k(Z^k - X^k)$. Since the Stiefel manifold is compact, there exist some η^k such that $Z^k = \text{Exp}_{X^k}(\eta^k)$ where Exp_{X^k} is the exponential map from $T_{X^k}\text{St}(n, p)$ to $\text{St}(n, p)$. Following the proof in [5, Appendix B] and Assumption (B1) (Z^k can be sufficiently close to X^k for large k), we have

$$(3.5) \quad \|Z^k - X^k - \eta^k\|_{\mathbb{F}} \leq \kappa_1 \|\eta^k\|_{\mathbb{F}}^2$$

with a positive constant κ_1 for all sufficiently large k . Moreover, since the Hessian $\text{Hess} f(X_*)$ is positive definite and (B4) is satisfied, it holds for sufficiently large k :

$$\begin{aligned} \|\text{Hess} m_k(X^k)[\eta^k]\|_{\mathbb{F}} &= \|\text{Hess} m_k(X^k)[Z^k - X^k]\|_{\mathbb{F}} + O(\|\eta^k\|_{\mathbb{F}}^2) \\ &= \|\text{Hess} f(X^k)[Z^k - X^k] + (\text{Hess} m_k(X^k) - \text{Hess} f(X^k))[Z^k - X^k]\|_{\mathbb{F}} + O(\|\eta^k\|_{\mathbb{F}}^2) \\ &\geq \lambda_{\min}(\text{Hess} f(X^k))\|Z^k - X^k\|_{\mathbb{F}} + o(\|Z^k - X^k\|_{\mathbb{F}}) + O(\|\eta^k\|_{\mathbb{F}}^2) \\ &\geq \lambda_{\min}(\text{Hess} f(X^k))\|\eta^k\|_{\mathbb{F}} + o(\|\eta^k\|_{\mathbb{F}}), \end{aligned}$$

where $\lambda_{\min}(\text{Hess} f(X^k))$ is the minimal spectrum of $\text{Hess} f(X^k)$. From Assumption (B2)-(B3), [2, Proposition 5.5.4] and the Taylor expansion of $m_k \circ \text{Exp}_{X^k}$, we have

$$\|\text{grad}(m_k \circ \text{Exp}_{X^k})(\eta^k) - \text{grad} f(X^k)\|_{\mathbb{F}} = \|\text{Hess} f(X^k)[\eta^k]\|_{\mathbb{F}} + o(\|\eta^k\|_{\mathbb{F}}) \geq \frac{\kappa_2}{2} \|\eta^k\|_{\mathbb{F}},$$

where $\kappa_2 := \lambda_{\min}(\text{Hess} f(X_*))$. By [1, Lemma 7.4.9], we have

$$(3.6) \quad \|\eta^k\|_{\mathbb{F}} \leq \frac{2}{\kappa_2} (\|\text{grad} f(X^k)\|_{\mathbb{F}} + \tilde{c} \|\text{grad} m_k(Z^k)\|_{\mathbb{F}}) \leq \frac{2(1 + \tilde{c}\theta^k)}{\kappa_2} \|\text{grad} f(X^k)\|_{\mathbb{F}},$$

where $\tilde{c} > 0$ is a constant and the second inequality is from the inexact condition (3.2). It follows from the continuity of $\nabla^2 f$, (3.1), (3.5) and (3.6) that

$$\begin{aligned} 1 - r_k &\leq \frac{1}{2c} \left(\frac{\|(\nabla^2 f(X^k) - \mathcal{B}^k)[Z^k - X^k]\|_{\mathbb{F}} \|Z^k - X^k\|_{\mathbb{F}}}{\|\text{grad} f(X^k)\|_{\mathbb{F}}^2} \right. \\ &\quad \left. + \frac{\|\nabla^2 f(X_\delta^k) - \nabla^2 f(X^k)\| \|Z^k - X^k\|_{\mathbb{F}}^2}{\|\text{grad} f(X^k)\|_{\mathbb{F}}^2} \right) \rightarrow 0. \end{aligned}$$

Therefore, the iterations are eventually very successful. \square

As a result, the q-superlinear convergence can also be guaranteed.

THEOREM 6. *Suppose that Assumptions (B1)-(B4) and conditions (3.1) and (3.2) hold. Then the sequence $\{X^k\}$ converges q-superlinearly to X_* .*

Proof. We consider the cubic model here, while the local q-superlinear convergence of quadratic model can be showed by a similar fashion. Since the iterations are eventually very successful, we have $X^{k+1} = Z^k$ and τ_k converges to zero. From (3.2), we have

$$(3.7) \quad \begin{aligned} \|\text{grad} m_k(X^{k+1})\|_{\mathbb{F}} &= \|\text{Proj}_{X^{k+1}}(\nabla f(X^k) + \mathcal{B}^k[\Delta^k] + \tau_k \|\Delta^k\|_{\mathbb{F}} \Delta^k)\|_{\mathbb{F}} \\ &\leq \theta^k \|\text{grad} f(X^k)\|_{\mathbb{F}}, \end{aligned}$$

where $\Delta^k = Z^k - X^k$. Hence,

$$(3.8) \quad \begin{aligned} &\|\text{grad} f(X^{k+1})\|_{\mathbb{F}} \\ &= \|\text{Proj}_{X^{k+1}}(\nabla f(X^{k+1}))\|_{\mathbb{F}} \\ &= \|\text{Proj}_{X^{k+1}}(\nabla f(X^k) + \nabla^2 f(X^k)[\Delta^k] + o(\|\Delta^k\|_{\mathbb{F}}))\|_{\mathbb{F}} \\ &= \|\text{Proj}_{X^{k+1}}(\nabla f(X^k) + \mathcal{B}^k[\Delta^k] + o(\|\Delta^k\|_{\mathbb{F}}) + (\nabla^2 f(X^k) - \mathcal{B}^k)[\Delta^k])\|_{\mathbb{F}} \\ &\leq \theta^k \|\text{grad} f(X^k)\|_{\mathbb{F}} + o(\|\Delta^k\|_{\mathbb{F}}). \end{aligned}$$

It follows from a similar argument to (3.6) that there exists some constant c_1

$$\|\Delta^k\|_{\mathbb{F}} \leq c_1 \|\text{grad} f(X^k)\|_{\mathbb{F}},$$

for sufficiently large k . Therefore, from (3.8) and the definition of θ^k , we have

$$(3.9) \quad \frac{\|\text{grad} f(X^{k+1})\|_{\mathbb{F}}}{\|\text{grad} f(X^k)\|_{\mathbb{F}}} \rightarrow 0.$$

Combining (3.9), Assumption (B3) and [2, Lemma 7.4.8], it yields

$$\frac{\text{dist}(X^{k+1}, X_*)}{\text{dist}(X^k, X_*)} \rightarrow 0,$$

where $\text{dist}(X, Y)$ is the geodesic distance between X and Y which belong to $\text{St}(n, p)$. This completes the proof. \square

4. Linear eigenvalue problem. In this section, we apply the aforementioned strategy to the following linear eigenvalue problem

$$(4.1) \quad \min_{X \in \mathbb{R}^{n \times p}} f(X) := \frac{1}{2} \text{tr}(X^\top C X) \quad \text{s.t.} \quad X^\top X = I_p,$$

where $C := A + B$. Here, $A, B \in \mathbb{R}^{n \times n}$ are symmetric matrices and we assume that the multiplication of BX is much more expensive than that of AX . Motivated by the quasi-Newton methods and eliminating the linear term in subproblem (2.1), we investigate the multiseant conditions in [13]

$$(4.2) \quad \hat{B}^k X^k = B X^k, \quad \hat{B}^k S^k = B S^k$$

with $S^k = X^k - X^{k-1}$. By a brief induction, we have an equivalent form of (4.2)

$$(4.3) \quad \hat{B}^k[X^{k-1}, X^k] = B[X^{k-1}, X^k].$$

Then, using the limited-memory Nyström approximation, we obtain the approximated matrix \hat{B}^k as

$$(4.4) \quad \hat{B}^k = W^k((W^k)^\top O^k)^\dagger W_k^\top,$$

where

$$(4.5) \quad O^k = \mathbf{orth}(\text{span}\{X^{k-1}, X^k\}), \text{ and } W^k = BO^k.$$

Here, $\mathbf{orth}(Z)$ is to find the orthogonal basis of the space spanned by Z . Therefore, an approximation C^k to C can be set as

$$(4.6) \quad C^k = A + \hat{B}^k.$$

Since the objective function is invariant under rotation, i.e., $f(XQ) = f(X)$ for orthogonal matrix $Q \in \mathbb{R}^{p \times p}$, we also want to construct a subproblem whose objective function inherits the same property. Therefore, we use the distance function between X^k and X as

$$d_p(X, X^k) = \|XX^\top - X^k(X^k)^\top\|_{\mathbb{F}}^2,$$

which has been considered in [10, 39, 46] for the electronic structure calculation. Since X^k and X are orthonormal matrices, we have

$$(4.7) \quad \begin{aligned} d_p(X, X^k) &= \text{tr}((XX^\top - X^k(X^k)^\top)(XX^\top - X^k(X^k)^\top)) \\ &= 2p - 2\text{tr}(X^\top X^k(X^k)^\top X), \end{aligned}$$

which implies that $d_p(X, X^k)$ is a quadratic function on X . Consequently, the subproblem can be constructed as

$$(4.8) \quad \min_{X \in \mathbb{R}^{n \times p}} m_k(X) \quad \text{s.t.} \quad X^\top X = I_p,$$

where

$$m_k(X) := \frac{1}{2}\text{tr}(X^\top C^k X) + \frac{\tau_k}{4}d_p(X, X^k).$$

From the equivalent expression of $d_p(X, X^k)$ in (4.7), problem (4.8) is actually a linear eigenvalue problem

$$\begin{aligned} (A + \hat{B}^k - \tau_k X^k(X^k)^\top)X &= X\Lambda, \\ X^\top X &= I_p, \end{aligned}$$

where Λ is a diagonal matrix whose diagonal elements are the p smallest eigenvalues of $A + \hat{B}^k - \tau_k X^k(X^k)^\top$. Due to the low computational cost of $A + \hat{B}^k - \tau_k X^k(X^k)^\top$ compared to $A + B$, the subproblem (4.8) can be solved efficiently using existing eigensolvers. As in Algorithm 1, we first solve subproblem (4.8) to obtain a trial point and compute the ratio (2.13) between the actual reduction and predicted reduction based on this trial point. Then the iterate and regularization parameter are updated according to (2.13) and (2.15). Note that it is not necessary to solve the subproblems highly accurately in practice.

4.1. Convergence. Although the convergence analysis in section 3 is based on the regularization terms (2.2) and (2.3), similar results can be established with the specified regularization term $\frac{\tau_k}{4} d_p(X, X^k)$ using the sufficient descent condition (3.1). It follows from the construction of C^k in (4.6) that

$$\|C\|_2 \leq \|A\|_2 + \|B\|_2, \quad \|C^k\|_2 \leq \|A\|_2 + \|B\|_2$$

for any given matrices A and B . Hence, Assumptions (A1) and (A2) hold with $L_f = \kappa_H = \|A\|_2 + \|B\|_2$. We have the following theorem on the global convergence.

THEOREM 7. *Suppose that the inexact condition (3.1) holds. Then, for the Riemannian gradients, either*

$$(I_n - X^t(X^t)^\top)(CX^t) = 0 \text{ for some } t > 0 \text{ or } \lim_{k \rightarrow \infty} \|(I_n - X^k(X^k)^\top)(CX^k)\|_F = 0. \blacksquare$$

Proof. It can be guaranteed that the distance $d_p(X, X^k)$ is very small for a large enough regularization parameter τ_k by a similar argument to [16, Lemma 9]. Specifically, the reduction of the subproblem requires that

$$\langle Z^k, C^k Z^k \rangle + \frac{\tau_k}{4} \|Z^k(Z^k)^\top - X^k(X^k)^\top\|_F^2 - \langle X^k, C^k X^k \rangle \leq 0.$$

From the cyclic property of the trace operator, it holds that

$$\langle Z^k, Z^k(Z^k)^\top - X^k(X^k)^\top \rangle + \frac{\tau_k}{4} \|Z^k(Z^k)^\top - X^k(X^k)^\top\|_F^2 \leq 0.$$

Then

$$(4.9) \quad \|Z^k(Z^k)^\top - X^k(X^k)^\top\|_F \leq \frac{4\kappa_H}{\tau_k}.$$

From the descent condition (3.1) for the subproblem, there exists some positive constant ν such that

$$(4.10) \quad m_k(Z^k) - m_k(X^k) \geq -\frac{\nu}{\tau_k} \|\text{grad} f(X^k)\|_F^2.$$

Based on the properties of C^k and C , we have

$$(4.11) \quad \begin{aligned} & f(Z^k) - f(X^k) - (m_k(Z^k) - m_k(X^k)) \\ &= \langle Z^k, CZ^k \rangle - \langle Z^k, C^k Z^k \rangle - \frac{\tau_k}{4} \|Z^k(Z^k)^\top - X^k(X^k)^\top\|_F^2 \\ &\leq \langle C - C^k, Z^k(Z^k)^\top \rangle = \langle C - C^k, (Z^k(Z^k)^\top - X^k(X^k)^\top)^2 \rangle \\ &\leq (L_f + \kappa_H) \|Z^k(Z^k)^\top - X^k(X^k)^\top\|_F^2 \\ &\leq \frac{16\kappa_H^2(L_f + \kappa_H)}{\tau_k^2}, \end{aligned}$$

where the second equality is due to $CX^k = C^k X^k$, the unitary Z^k and X^k , as well as

$$\langle C - C^k, Z^k(Z^k)^\top X^k(X^k)^\top \rangle = \langle C - C^k, X^k(X^k)^\top Z^k(Z^k)^\top \rangle = 0.$$

Combining (4.10) and (4.11), we have that

$$1 - r_k = \frac{f(Z^k) - f(X^k) - (m_k(Z^k) - m_k(X^k))}{m_k(X^k) - m_k(Z^k)} \leq 1 - \eta_2$$

for sufficiently large τ_k as in [16, Lemma 8]. Since the subproblem is solved with some sufficient reduction, the reduction of the original objective f holds for large τ_k (i.e., r_k is close to 1). Then the convergence of the norm of the Riemannian gradient $(I_n - X^k(X^k)^\top)CX^k$ follows in a similar fashion as [16, Theorem 11]. \square

The ACE method in [29] needs an estimation β explicitly such that $B - \beta I_n$ is negative definite. By considering an equivalent matrix $(A + \beta I_n) + (B - \beta I_n)$, the convergence of ACE to a global minimizer is given. On the other hand, our algorithmic framework uses an adaptive strategy to choose τ_k to guarantee the convergence to a stationary point. By using similar proof techniques in [29], one may also establish the convergence to a global minimizer.

5. Electronic structure calculation.

5.1. Formulation. We now introduce the KS and HF total minimization models and present their gradient and Hessian of the objective functions in these two models. After some proper discretization, the wave functions of p occupied states can be approximated by a matrix $X = [x_1, \dots, x_p] \in \mathbb{C}^{n \times p}$ with $X^*X = I_p$, where n corresponds to the spatial degrees of freedom. The charge density associated with the occupied states is defined as

$$\rho(X) = \text{diag}(XX^*).$$

Unless otherwise specified, we use the abbreviation ρ for $\rho(X)$ in the following. The total energy functional is defined as

$$(5.1) \quad E_{\text{ks}}(X) := \frac{1}{4} \text{tr}(X^*LX) + \frac{1}{2} \text{tr}(X^*V_{\text{ion}}X) + \frac{1}{2} \sum_l \sum_i \zeta_l |x_i^* w_l|^2 + \frac{1}{4} \rho^\top L^\dagger \rho + \frac{1}{2} e^\top \epsilon_{\text{xc}}(\rho),$$

where L is a discretized Laplacian operator, V_{ion} is the constant ionic pseudopotentials, w_l represents a discretized pseudopotential reference projection function, ζ_l is a constant whose value is ± 1 , e is a vector of all ones in \mathbb{R}^n , and ϵ_{xc} is related to the exchange correlation energy. Therefore, the KS total energy minimization problem can be expressed as

$$(5.2) \quad \min_{X \in \mathbb{C}^{n \times p}} E_{\text{ks}}(X) \quad \text{s.t.} \quad X^*X = I_p.$$

Let $\mu_{\text{xc}}(\rho) = \frac{\partial \epsilon_{\text{xc}}(\rho)}{\partial \rho}$ and denote the Hamilton $H_{\text{ks}}(X)$ by

$$(5.3) \quad H_{\text{ks}}(X) := \frac{1}{2}L + V_{\text{ion}} + \sum_l \zeta_l w_l w_l^* + \text{Diag}((\Re L^\dagger)\rho) + \text{Diag}(\mu_{\text{xc}}(\rho)^* e).$$

Then the Euclidean gradient of $E_{\text{ks}}(X)$ is computed as

$$(5.4) \quad \nabla E_{\text{ks}}(X) = H_{\text{ks}}(X)X.$$

Under the assumption that $\epsilon_{\text{xc}}(\rho(X))$ is twice differentiable with respect to $\rho(X)$, Lemma 2.1 in [43] gives an explicit form of the Hessian of $E_{\text{ks}}(X)$ as

$$(5.5) \quad \nabla^2 E_{\text{ks}}(X)[U] = H_{\text{ks}}(X)U + \mathcal{R}(X)[U],$$

where $U \in \mathbb{C}^{n \times p}$ and $\mathcal{R}(X)[U] := \text{Diag} \left((\Re L^\dagger + \frac{\partial^2 \epsilon_{\text{xc}}}{\partial \rho^2} e)(\bar{X} \odot U + X \odot \bar{U})e \right) X$.

Compared with KSDFIT, the HF theory can provide a more accurate model to electronic structure calculations by involving the Fock exchange operator. After discretization, the exchange-correlation operator $\mathcal{V}(\cdot) : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$ is usually a fourth-order tensor, see equations (3.3) and (3.4) in [27] for details. Furthermore, it is easy to see from [27] that $\mathcal{V}(\cdot)$ satisfies the following properties: (i) For any $D_1, D_2 \in \mathbb{C}^{n \times n}$, there holds $\langle \mathcal{V}(D_1), D_2 \rangle = \langle \mathcal{V}(D_2), D_1 \rangle$, which further implies that

$$(5.6) \quad \langle \mathcal{V}(D_1 + D_2), D_1 + D_2 \rangle = \langle \mathcal{V}(D_1), D_1 \rangle + 2 \langle \mathcal{V}(D_1), D_2 \rangle + \langle \mathcal{V}(D_2), D_2 \rangle.$$

(ii) If D is Hermitian, $\mathcal{V}(D)$ is also Hermitian. Besides, it should be emphasized that computing $\mathcal{V}(U)$ is always very expensive since it needs to perform the multiplication between a $n \times n \times n \times n$ fourth-order tensor and a n -by- n matrix. The corresponding Fock energy is defined as

$$(5.7) \quad E_{\text{f}}(X) := \frac{1}{4} \langle \mathcal{V}(XX^*)X, X \rangle = \frac{1}{4} \langle \mathcal{V}(XX^*), XX^* \rangle.$$

Then the HF total energy minimization problem can be formulated as

$$(5.8) \quad \min_{X \in \mathbb{C}^{n \times p}} E_{\text{hf}}(X) := E_{\text{ks}}(X) + E_{\text{f}}(X) \quad \text{s.t.} \quad X^*X = I_p.$$

We now can explicitly compute the gradient and Hessian of $E_{\text{f}}(X)$ by using the properties of $\mathcal{V}(\cdot)$.

LEMMA 8. *Given $U \in \mathbb{C}^{n \times p}$, the gradient and the Hessian along U of $E_{\text{f}}(X)$ are, respectively,*

$$(5.9) \quad \nabla E_{\text{f}}(X) = \mathcal{V}(XX^*)X,$$

$$(5.10) \quad \nabla^2 E_{\text{f}}(X)[U] = \mathcal{V}(XX^*)U + \mathcal{V}(XU^* + UX^*)X.$$

Proof. We first compute the value $E_{\text{f}}(X+U)$. For simplicity, denote $D := XU^* + UX^*$. Using the property (5.6), by some easy calculations, we have

$$\begin{aligned} 4E_{\text{f}}(X+U) &= \langle \mathcal{V}((X+U)(X+U)^*), (X+U)(X+U)^* \rangle \\ &= 4E_{\text{f}}(X) + 2 \langle \mathcal{V}(XX^*), D + UU^* \rangle + \langle \mathcal{V}(D + UU^*), D + UU^* \rangle \\ &= 4E_{\text{f}}(X) + 2 \langle \mathcal{V}(XX^*), D \rangle + 2 \langle \mathcal{V}(XX^*), UU^* \rangle + \langle \mathcal{V}(D), D \rangle + \text{h.o.t.}, \end{aligned}$$

where h.o.t. denotes the higher-order terms. Noting that $\mathcal{V}(XX^*)$ and $\mathcal{V}(D)$ are both Hermitian, we have from the above assertions that

$$(5.11) \quad E_{\text{f}}(X+U) = E_{\text{f}}(X) + \Re \langle \mathcal{V}(XX^*)X, U \rangle + \frac{1}{2} \Re \langle \mathcal{V}(XX^*)U + \mathcal{V}(D)X, U \rangle + \text{h.o.t.}$$

Finally, it follows from expansion (1.2) in [43] that the second-order Taylor expression in X can be expressed as

$$E_{\text{f}}(X+U) = E_{\text{f}}(X) + \Re \langle \nabla E_{\text{f}}(X), U \rangle + \frac{1}{2} \Re \langle \nabla^2 E_{\text{f}}(X)[U], U \rangle + \text{h.o.t.},$$

which with (5.11) implies (5.9) and (5.10). The proof is completed. \square

Let $H_{\text{hf}}(X) := H_{\text{ks}}(X) + \mathcal{V}(XX^*)$ be the HF Hamilton. Recalling that $E_{\text{hf}}(X) = E_{\text{ks}}(X) + E_f(X)$, we have from (5.4) and (5.9) that

$$(5.12) \quad \nabla E_{\text{hf}}(X) = H_{\text{ks}}(X)X + \mathcal{V}(XX^*)X = H_{\text{hf}}(X)X$$

and have from (5.5) and (5.10) that

$$(5.13) \quad \nabla^2 E_{\text{hf}}(X)[U] = H_{\text{hf}}(X)U + \mathcal{R}(X)[U] + \mathcal{V}(XU^* + UX^*)X.$$

5.2. Review of Algorithms for the KSDFT and HF Models. We next briefly introduce the widely used methods for solving the KSDFT and HF models. For the KSDFT model (5.2), the most popular method is the SCF method [27]. At the k -th iteration, SCF first fixes $H_{\text{ks}}(X)$ to be $H(X^k)$ and then updates X^{k+1} via solving the linear eigenvalue problem

$$(5.14) \quad X^{k+1} := \arg \min_{X \in \mathbb{C}^{n \times p}} \frac{1}{2} \langle X, H(X^k)X \rangle \quad \text{s.t.} \quad X^*X = I_p.$$

Because the complexity of the HF model (5.8) is much higher than that of the KSDFT model, using SCF method directly may not obtain desired results. Since computing $\mathcal{V}(X^k(X^k)^*)U$ with some matrix U of proper dimension is still very expensive, we investigate the limited-memory Nyström approximation $\hat{\mathcal{V}}(X^k(X^k)^*)$ to approximate $\mathcal{V}(X^k(X^k)^*)$ to reduce the computational cost, i.e.,

$$(5.15) \quad \hat{\mathcal{V}}(X^k(X^k)^*) := Z(Z^*\Omega)^\dagger Z^*,$$

where $Z = \mathcal{V}(X^k(X^k)^*)\Omega$ and Ω is any orthogonal matrix whose columns form an orthogonal basis of the subspace such as

$$\text{span}\{X^k\}, \text{span}\{X^{k-1}, X^k\} \text{ or } \text{span}\{X^{k-1}, X^k, \mathcal{V}(X^k(X^k)^*)X^k\}.$$

We should note that a similar idea called adaptive compression method was proposed in [28], which only considers to compress the operator $\mathcal{V}(X^k(X^k)^*)$ on the subspace $\text{span}\{X^k\}$. Then a new subproblem is constructed as

$$(5.16) \quad \min_{X \in \mathbb{C}^{n \times p}} E_{\text{ks}}(X) + \frac{1}{4} \left\langle \hat{\mathcal{V}}(X^k(X^k)^*)X, X \right\rangle \quad \text{s.t.} \quad X^*X = I_p.$$

Here, the exact form of the easier parts E_{ks} is preserved while its second-order approximation is used in the construction of subproblem (2.1). As in the subproblem (2.1), we can utilize the Riemannian gradient method or the modified CG method based on the following linear equation

$$\text{Proj}_{X^k} \left(\nabla^2 E_{\text{ks}}(X^k)[\xi] + \frac{1}{2} \hat{\mathcal{V}}(X^k(X^k)^*)\xi - \xi_{\text{sym}}((X^k)^* \nabla f(X^k)) \right) = -\text{grad} E_{\text{hf}}(X^k)$$

to solve (5.16) inexactly. Since (5.16) is a KS-like problem, we can also use the SCF method. Here, we present the detailed algorithm in Algorithm 2.

Algorithm 2: Iterative method for (5.8) using Nyström approximation

Input initial guess $X^0 \in \mathbb{C}^{n \times p}$ with $(X^0)^* X^0 = I_p$. Set $k = 0$.

while *Stopping conditions not met* **do**

 Compute the limited-memory Nyström approximation $\hat{V}(X^k(X^k)^*)$.

 Construct the subproblem (5.16) and solve it inexactly via the Riemannian gradient method or the modified CG method or the SCF method to obtain X^{k+1} .

 Set $k \leftarrow k + 1$.

We note that Algorithm 2 is similar to the two-level nested SCF method with the ACE formulation [28] when the subspace in (5.15) and inner solver for (5.16) are chosen as $\text{span}\{X^k\}$ and SCF, respectively.

5.3. Construction of the structured approximation \mathcal{B}^k . Note that the Hessian of the KSDFT or HF total energy minimization takes the natural structure (1.2), we next give the specific choices of $\mathcal{H}^c(X^k)$ and $\mathcal{H}^e(X^k)$, which are key to formulate the the structured approximation \mathcal{B}^k .

For the KS problem (5.2), we have its exact Hessian in (5.5). Since the computational cost of the parts $\frac{1}{2}L + \sum_l \zeta_l w_l w_l^*$ are much cheaper than the remaining parts in $\nabla^2 E_{\text{ks}}$, we can choose

$$(5.17) \quad \mathcal{H}^c(X^k) = \frac{1}{2}L + \sum_l \zeta_l w_l w_l^*, \quad \mathcal{H}^e(X^k) = \nabla^2 E_{\text{ks}}(X^k) - \mathcal{H}^c(X^k).$$

The exact Hessian of $E_{\text{hf}}(X)$ in (5.8) can be separated naturally into two parts, i.e., $\nabla^2 E_{\text{ks}}(X) + \nabla^2 E_{\text{f}}(X)$. Usually the hybrid exchange operator $\mathcal{V}(XX^*)$ can take more than 95% of the overall time of the multiplication of $H_{\text{hf}}(X)[U]$ in many real applications [29]. Recalling (5.5), (5.10) and (5.13), we know that the computational cost of $\nabla^2 E_{\text{f}}(X)$ is much higher than that of $\nabla^2 E_{\text{ks}}(X)$. Hence, we obtain the decomposition as

$$(5.18) \quad \mathcal{H}^c(X^k) = \nabla^2 E_{\text{ks}}(X^k), \quad \mathcal{H}^e(X^k) = \nabla^2 E_{\text{f}}(X^k).$$

Moreover, we can split the Hessian of $\nabla^2 E_{\text{ks}}(X^k)$ as done in (5.17) and obtain an alternative decomposition as

$$(5.19) \quad \mathcal{H}^c(X^k) = H_{\text{ks}}(X^k), \quad \mathcal{H}^e(X^k) = \nabla^2 E_{\text{f}}(X^k) + (\nabla^2 E_{\text{ks}}(X^k) - \mathcal{H}^c(X^k)).$$

Finally, we emphasize that the limited-memory Nyström approximation (5.15) can serve as a good initial approximation for the part $\nabla^2 E_{\text{f}}(X^k)$.

5.4. Subspace construction for the KSDFT model. As presented in Algorithm 1, the subspace method plays an important role when the modified CG method does not perform well. The first-order optimality conditions for (5.2) and (5.8) are

$$H(X)X = X\Lambda, \quad X^*X = I_p,$$

where $X \in \mathbb{C}^{n \times p}$, Λ is a diagonal matrix and H represents H_{ks} for (5.2) and H_{hf} for (5.8). Then, problems (5.2) and (5.8) are actually a nonlinear eigenvalue problem which aims to find the p smallest eigenvalues of H . We should point out that

in principle X consists of the eigenvectors of $H(X)$ but not necessary the eigenvectors corresponding to the p smallest eigenvalues. Since the optima X is still the eigenvectors of $H(X)$, we can construct some subspace which contains these possible wanted eigenvectors. Specifically, at current iterate, we first compute the first γp smallest eigenvalues and their corresponding eigenvectors of $H(X^k)$, denoted by Γ^k , then construct the subspace as

$$(5.20) \quad \text{span}\{X^{k-1}, X^k, \text{grad} f(X^k), \Gamma^k\},$$

with some small integer γ . With this subspace construction, Algorithm 1 will more likely escape a stagnated point.

6. Numerical experiments. In this section, we present some experiment results to illustrate the efficiency of the limited-memory Nyström approximation and our Algorithm 1. All codes were run in a workstation with Intel Xenon E5-2680 v4 processors at 2.40GHz and 256GB memory running CentOS 7.3.1611 and MATLAB R2017b.

6.1. Linear eigenvalue problem. We first construct A and B by using the following MATLAB commands:

$$A = \text{randn}(n, n); A = (A + A^\top)/2;$$

$$B = 0.01\text{randn}(n, n); B = (B + B^\top)/2; B = B - T; B = -B,$$

where randn and rand are the built-in functions in MATLAB, $T = \lambda_{\min}(B)I_n$ and $\lambda_{\min}(B)$ is the smallest eigenvalue of B . Then B is negative definite and A is symmetric. In our implementation, we compute the multiplication BX using $\frac{1}{19} \sum_{i=1}^{19} BX$ such that BX consumes about 95% of the whole computational time. In the second example, we set A to be a sparse matrix as

$$A = \text{gallery}(\text{'wathen'}, 5s, 5s)$$

with parameter s and B is the same as the first example except that BX is computed directly. Since A is sufficiently sparse, its computational cost AX is much smaller than that of BX . We use the following stopping criterion

$$(6.1) \quad \text{err} := \max_{i=1, \dots, p} \left\{ \frac{\|(A+B)x_i - \mu_i x_i\|_2}{\max(1, |\mu_i|)} \right\} \leq 10^{-10},$$

where x_i is the i -th column of the current iterate X^k and μ_i is the corresponding approximated eigenvalue.

The numerical results of the first and second examples are summarized in Tables 1 and 2, respectively. In these tables, EIGS is the built-in function “eigs” in MATLAB. LOBPCG is the locally optimal block preconditioned conjugate gradient method [25]. ASQN is the algorithm described in section 4. The difference between ACE and ASQN is that we take O^k as $\text{orth}(\text{span}\{X^k\})$ but not $\text{orth}(\text{span}\{X^{k-1}, X^k\})$. Since a good initial guess X^k is known at the $(k+1)$ -th iteration, LOBPCG is utilized to solve the corresponding linear eigenvalue subproblem (4.8). Note that BX^{k-1} and BX^k are available from the computation of the residual, we then adopt the orthogonalization

TABLE 1
Numerical results on random matrices

	AV/BV	err	time	AV/BV	err	time
$p = 10$						
n	5000			6000		
EIGS	459/459	8.0e-11	45.1	730/730	6.9e-11	94.3
LOBPCG	1717/1717	9.9e-11	128.9	2105/2105	9.8e-11	249.9
ASQN	2323/150	9.2e-11	13.3	2798/160	9.5e-11	22.8
ACE	4056/460	9.7e-11	30.8	4721/460	9.4e-11	47.4
n	8000			10000		
EIGS	538/538	8.7e-11	131.9	981/981	8.8e-11	327.3
LOBPCG	1996/1996	9.9e-11	336.7	2440/2440	9.7e-11	763.8
ASQN	2706/150	8.9e-11	29.8	2920/150	9.7e-11	50.2
ACE	4537/450	9.8e-11	66.3	4554/400	9.6e-11	99.4
$n = 5000$						
p	10			20		
EIGS	459/459	8.0e-11	45.1	638/638	3.2e-11	62.7
LOBPCG	1717/1717	9.9e-11	128.9	2914/2914	9.8e-11	130.3
ASQN	2323/150	9.2e-11	13.3	3809/260	9.2e-11	8.9
ACE	4056/460	9.7e-11	30.8	5902/680	9.5e-11	16.5
p	30			50		
EIGS	660/660	3.0e-11	62.8	879/879	1.6e-12	83.6
LOBPCG	4458/4458	1.0e-10	217.6	5766/5766	9.5e-11	186.7
ASQN	5315/420	9.8e-11	11.4	7879/650	9.8e-11	17.8
ACE	9701/1530	9.4e-11	23.0	21664/4450	1.0e-10	50.9

technique in [30] to compute O^k and W^k in (4.5) without extra multiplication BO^k . The labels “AV” and “BV” denote the total number of matrix-vector multiplications (MV), counting each operation $AX, BX \in \mathbb{R}^{n \times p}$ as p MVs. The columns “err” and “time” are the maximal relative error of all p eigenvectors defined in (6.1), and the wall-clock time in seconds of each algorithm, respectively. The maximal number of iterations for ASQN and ACE is set to 200.

As shown in Table 1, with fixed $p = 10$ and different $n = 5000, 6000, 8000$ and 10000, we see that ASQN performs better than EIGS, LOBPCG and ACE in terms of both accuracy and time. ACE spends a relative long time to reach a solution with a similar accuracy. For the case $n = 5000$, ASQN can still give a accurate solution with less time than EIGS and LOBPCG, but ACE usually takes a long time to get a solution of high accuracy. Similar conclusions can also be seen from Table 2. In which, ACE and LOBPCG do not reach the given accuracy in the cases $n = 11041$ and $p = 30, 40, 50, 60$. From the calls of AV and BV , we see that the limited-memory Nyström method reduces the number of calls on the expensive part by doing more evaluations on the cheap part.

6.2. Kohn-Sham total energy minimization. We now test the electron structure calculation models in subsections 6.2 and 6.3 using the new version of the KS-SOLV package [45]. One of the main differences is that the new version uses the more recently developed optimized norm-conserving Vanderbilt pseudopotentials (ONCV) [14], which are compatible to those used in other community software packages such as Quantum ESPRESSO. The problem information is listed in Table 3. For fair com-

TABLE 2
Numerical results on sparse matrices

	AV/BV	err	time	AV/BV	err	time
$p = 10$						
s	7			8		
EIGS	1589/1589	8.9e-11	10.8	1097/1097	6.1e-11	13.4
LOBPCG	3346/3346	9.8e-11	24.6	4685/4685	4.6e-10	48.6
ASQN	5387/180	9.6e-11	7.1	4861/150	9.9e-11	5.9
ACE	14361/1600	9.6e-11	21.7	8810/600	9.6e-11	12.7
s	9			10		
EIGS	1326/1326	9.3e-11	21.2	1890/1890	6.8e-11	44.4
LOBPCG	4306/4306	1.7e-07	66.9	3895/3895	9.9e-11	91.9
ASQN	5303/190	8.5e-11	7.7	6198/200	8.9e-11	10.1
ACE	16253/1850	9.9e-11	34.6	10760/820	9.0e-11	22.2
s	11			12		
EIGS	1882/1882	1.5e-07	58.9	1463/1463	9.6e-11	65.4
LOBPCG	4282/4282	9.5e-11	136.0	4089/4089	9.9e-11	190.6
ASQN	8327/240	9.6e-11	16.7	6910/220	9.3e-11	17.5
ACE	15323/1060	9.7e-11	38.9	17907/2010	1.7e-08	65.5
$s = 12$						
p	10			20		
EIGS	1463/1463	9.6e-11	65.4	1148/1148	5.8e-11	50.2
LOBPCG	4089/4089	9.9e-11	190.6	5530/5530	9.8e-11	86.4
ASQN	6910/220	9.3e-11	17.5	9749/340	9.5e-11	16.3
ACE	17907/2010	1.7e-08	65.5	14108/960	9.8e-11	23.4
p	30			40		
EIGS	1784/1784	8.1e-11	74.8	1836/1836	4.8e-11	69.1
LOBPCG	9076/9076	5.3e-09	173.3	12192/12192	4.6e-10	207.2
ASQN	17056/870	9.6e-11	41.5	19967/960	9.9e-11	39.9
ACE	37162/6030	9.1e-09	78.4	48098/8040	4.6e-07	105.4
p	50			60		
EIGS	1743/1743	7.3e-11	69.1	2122/2122	1.6e-11	86.7
LOBPCG	12288/12288	1.4e-09	168.4	15716/15716	1.1e-08	199.5
ASQN	21330/1300	9.3e-11	53.6	26343/1620	9.7e-11	71.8
ACE	49165/10050	2.9e-06	110.1	62668/12060	2.3e-08	134.0

parisons, we stop all algorithms when the Frobenius norm of the Riemannian gradient is less than 10^{-6} or the maximal number of iterations is reached. In the following tables, the column “solver” denotes which specified solver is used. The columns “fval”, “nrmG”, “time” are the final objective function value, the final Frobenius norm of the Riemannian gradient and the wall-clock time in seconds of each algorithm, respectively.

In this test, we compare structured quasi-Newton method with the SCF in KS-SOLV [45], the Riemannian L-BFGS method (RQN) in Manopt [6], the Riemannian gradient method with BB step size (GGB) and the adaptive regularized Newton method (ARNT) [16]. The default parameters therein are used. Our Algorithm 1 with the approximation with (5.17) is denoted by ASQN. The parameters setting of ASQN is same to that of ARNT [16].

For each algorithm, we first use GGB to generate a good starting point with stopping criterion $\|\text{grad } f(X^k)\|_F \leq 10^{-1}$ and a maximum of 2000 iterations. The

maximal numbers of iterations for SCF, GBB, ARNT, ASQN and RQN are set as 1000, 10000, 500, 500, 500 and 1000, respectively. The numerical results are reported in Tables 4 and 5. The column “its” represents the total number of iterations in SCF, GBB and RQN, while the two numbers in ARNT, ASQN are the total number of outer iterations and the average numbers of inner iterations.

TABLE 3
Problem information.

name	(n_1, n_2, n_3)	n	p
alanine	(91,68,61)	35829	18
c12h26	(136,68,28)	16099	37
ctube661	(162,162,21)	35475	48
glutamine	(64,55,74)	16517	29
graphene16	(91,91,23)	12015	37
graphene30	(181,181,23)	48019	67
pentacene	(80,55,160)	44791	51
gaas	(49,49,49)	7153	36
si40	(129,129,129)	140089	80
si64	(93,93,93)	51627	128
al	(91,91,91)	47833	12
ptnio	(89,48,42)	11471	43
c	(46,46,46)	6031	2

From Tables 4 and 5, we can see that SCF failed in “graphene16”, “graphene30”, “al”, “ptnio” and “c”. We next explain why SCF fails by taking “c” and “graphene16” as examples. For the case “c”, we obtain the same solution by using GBB, ARNT and ASQN. The number of wanted wave functions are 2, i.e., $p = 2$. With some abuse of notation, we denote the final solution by $X = [x_1, x_2]$. Since X satisfies the first-order optimality condition, the columns of X are also eigenvectors of $H(X)$ and the corresponding eigenvalues of $H(X)$ are -1.8790, -0.6058. On the other hand, the smallest four eigenvalues of $H(X)$ are -1.8790, -0.6577, -0.6058, -0.6058 and the corresponding eigenvectors, denoted by $Y = [y_1, y_2, y_3, y_4]$. The energies and norms of Riemannian gradients of the different eigenvector pairs $[x_1, x_2]$, $[y_1, y_2]$, $[y_1, y_3]$ and $[y_1, y_4]$ are $(-5.3127, 9.96 \times 10^{-7})$, $(-5.2903, 3.07 \times 10^{-1})$, $(-5.2937, 1.82 \times 10^{-1})$ and $(-4.6759, 1.82 \times 10^{-1})$, respectively. Comparing the angles between X and Y shows that x_1 is nearly parallel to y_1 but x_2 lies in the subspace spanned by $[y_3, y_4]$ other than y_2 . Hence, when the SCF method is used around X , the next point will jump to the subspace spanned by $[y_1, y_2]$. This indicates the failure of the *aufbau* principle, and thus the failure of the SCF procedure. This is consistent with the observation in the chemistry literature [42], where sometimes the converged solution may have a “hole” (i.e., unoccupied states) below the highest occupied energy level.

In the case “graphene16”, we still obtain the same solution from GBB, ARNT and ASQN. The number of wave functions p is 37. Let X be the computed solution and the corresponding eigenvalues of $H(X)$ be d . The smallest 37 eigenvalues and their corresponding eigenvectors of $H(X)$ are g and Y . We find that the first 36 elements of d and g are almost the same up to a machine accuracy, but the 37th element of d and g is 0.5821 and 0.5783, respectively. The energies and norms of Riemannian gradients of X and Y are $(-94.2613, 8.65 \times 10^{-7})$ and $(-94.2030, 6.95 \times 10^{-1})$, respectively.

TABLE 4
 Numerical results on KS total energy minimization.

solver	fval	nrmG	its	time	fval	nrmG	its	time
	alanine				c12h26			
SCF	-6.27084e+1	6.3e-7	11	64.0	-8.23006e+1	6.5e-7	10	61.1
GBB	-6.27084e+1	8.2e-7	92	71.3	-8.23006e+1	9.5e-7	89	65.8
ARNT	-6.27084e+1	3.8e-7	3(13.3)	63.0	-8.23006e+1	7.5e-7	3(15.3)	60.9
ASQN	-6.27084e+1	9.3e-7	13(11.8)	81.9	-8.23006e+1	9.3e-7	10(13.3)	67.8
RQN	-6.27084e+1	1.5e-6	34	114.9	-8.23006e+1	1.7e-6	45	120.0
	ctube661				glutamine			
SCF	-1.35378e+2	5.7e-7	11	200.4	-9.90525e+1	4.9e-7	10	49.5
GBB	-1.35378e+2	6.3e-7	102	199.7	-9.90525e+1	4.9e-7	63	44.0
ARNT	-1.35378e+2	3.2e-7	3(18.3)	168.3	-9.90525e+1	3.6e-7	3(12.0)	42.6
ASQN	-1.35378e+2	7.6e-7	11(12.8)	201.7	-9.90525e+1	5.3e-7	12(9.8)	50.7
RQN	-1.35378e+2	3.4e-6	40	308.8	-9.90525e+1	1.8e-6	26	72.8
	graphene16				graphene30			
SCF	-9.57196e+1	8.7e-4	1000	3438.4	-1.76663e+2	3.5e-4	1000	31897.6
GBB	-9.57220e+1	9.4e-7	434	185.1	-1.76663e+2	9.0e-7	904	3383.9
ARNT	-9.57220e+1	1.8e-7	4(37.2)	164.1	-1.76663e+2	4.2e-7	5(74.2)	2386.1
ASQN	-9.57220e+1	8.8e-7	23(24.1)	221.2	-1.76663e+2	7.2e-7	74(31.1)	4388.1
RQN	-9.57220e+1	1.6e-6	213	287.8	-1.76663e+2	3.3e-5	373	4296.7
	pentacene				gaas			
SCF	-1.30846e+2	8.5e-7	12	279.8	-2.86349e+2	5.8e-7	15	41.1
GBB	-1.30846e+2	9.6e-7	101	236.1	-2.86349e+2	7.5e-7	296	77.7
ARNT	-1.30846e+2	2.1e-7	3(14.0)	213.6	-2.86349e+2	7.4e-7	3(46.3)	59.9
ASQN	-1.30846e+2	9.0e-7	23(14.5)	423.0	-2.86349e+2	6.0e-7	35(24.8)	127.2
RQN	-1.30846e+2	2.1e-6	34	437.9	-2.86349e+2	1.5e-6	111	116.0
	si40				si64			
SCF	-1.57698e+2	7.5e-7	19	3587.4	-2.53730e+2	3.4e-7	10	1100.0
GBB	-1.57698e+2	8.7e-7	289	3657.2	-2.53730e+2	7.3e-7	249	1534.2
ARNT	-1.57698e+2	3.7e-7	3(33.0)	3343.9	-2.53730e+2	7.9e-7	3(47.3)	1106.8
ASQN	-1.57698e+2	9.8e-7	33(23.3)	4968.7	-2.53730e+2	9.4e-7	23(25.0)	1563.9
RQN	-1.57698e+2	4.1e-6	62	4946.7	-2.53730e+2	9.7e-7	122	2789.4
	al				ptnio			
SCF	-3.52151e+2	7.4e+0	1000	4221.1	-9.25762e+2	1.9e-1	1000	4461.9
GBB	-3.53707e+2	9.7e-7	1129	219.3	-9.26927e+2	2.4e-6	10000	5627.2
ARNT	-3.53710e+2	5.9e-7	59(60.7)	947.7	-9.26927e+2	9.4e-7	104(129.6)	7558.3
ASQN	-3.53710e+2	7.1e-7	94(47.3)	1395.4	-9.26927e+2	9.2e-7	153(69.6)	12728.1
RQN	-3.53710e+2	1.8e-3	267	323.4	-9.26925e+2	2.3e-4	380	924.4

Hence, SCF does not converge around the point X .

In Tables 4 and 5, ARNT usually converges in a few iterations due to the usage of the second-order information. It is often the fastest one in terms of time since the computational cost of two parts of the Hessian $\nabla^2 E_{\text{ks}}$ has no significant difference. GBB also performs comparably well as ARNT. ASQN works reasonably well on most problems. It takes more iterations than ARNT since the limit-memory approximation often is not as good as the Hessian. Because the costs of solving the subproblems of ASQN and ARNT are more or less the same, ASQN is not competitive to ARNT. However, by taking advantage of the problem structures, ASQN is still better than RQN in terms of computational time and accuracy. Finally, we show the convergence behaviors of these five methods on the system “glutamine” in Figure 1. Specifically,

the error of the objective function values is defined as

$$\Delta E_{\text{ks}}(X^k) = E_{\text{ks}}(X^k) - E_{\text{min}},$$

where E_{min} be the minimum of the total energy attained by all methods.

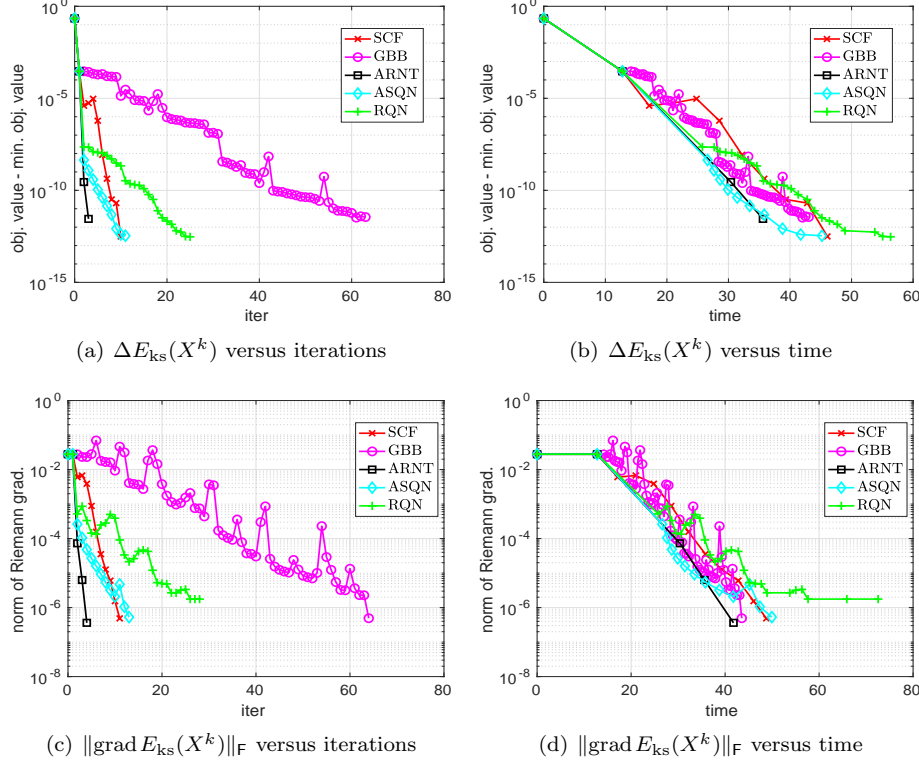


FIG. 1. Comparisons of different algorithms on “glutamine” of KS total energy minimization. The first two points are the input and output of the initial solver GBB, respectively.

TABLE 5
Numerical results on KS total energy minimization.

solver	fval	nrmG	its	time
c				
SCF	-5.29296e+0	7.3e-3	1000	168.3
GBB	-5.31268e+0	1.0e-6	3851	112.7
ARNT	-5.31268e+0	5.7e-7	96(49.1)	211.3
ASQN	-5.31268e+0	6.7e-7	104(38.5)	183.1
RQN	-5.31244e+0	1.4e-3	73	10.8

6.3. Hartree-Fock total energy minimization. In this subsection, we compare the performance of three variants of Algorithm 2 where the subproblem is solved by SCF (ACE), the modified CG method (ARN) and by GBB (GBBN), respectively, the Riemannian L-BFGS (RQN) method in Manopt [6], and two variants of Algorithm 1 with approximation (5.18) (ASQN) and approximation (5.19) (AKQN). Since the

computation of the exact Hessian $\nabla^2 E_{\text{hf}}$ is time-consuming, we do not present the results using the exact Hessian. The limited-memory Nyström approximation (5.15) serves as an initial Hessian approximation in both ASQN and AKQN. To compare the effectiveness of quasi-Newton approximation, we set $\mathcal{H}^e(X^k)$ to be the limited-memory Nyström approximation (5.15) in (5.19) and use the same framework as in Algorithm 1. We should mention that the subspace refinement is not used in ASQN and AKQN. Hence, only structured quasi-Newton iterations are performed in them. The default parameters in RQN and GBB are used. For ACE, GBBN, ASQN, AKQN and ARN, the subproblem is solved until the Frobenius-norm of the Riemannian gradient is less than $0.1 \min\{\|\text{grad } f(X^k)\|_{\text{F}}, 1\}$. We also use the adaptive strategy for choosing the maximal number of inner iterations of ARNT in [16] for GBBN, ASQN, AKQN and ARN. The settings of other parameters of ASQN, AKQN and ARN are the same to those in ARNT [16]. For all algorithms, we generate a good initial guess by using GBB to solve the corresponding KS total energy minimization problem (i.e., remove E_{f} part from E_{hf} in the objective function) until a maximal number of iterations 2000 is reached or the Frobenius-norm of the Riemannian gradient is smaller than 10^{-3} . The maximal number of iterations for ACE, GBBN, ASQN, ARN and AKQN is set to 200 while that of RQN is set to 1000.

A detailed summary of computational results is reported in Table 6. We see that ASQN performs best among all the algorithms in terms of both the number of iterations and time, especially in the systems: “alanine”, “graphene30”, “gaas” and “si40”. Usually, algorithms takes fewer iterations if more parts in the Hessian are preserved. Since the computational cost of the Fock energy dominates that of the KS part, algorithms using fewer outer iterations consumes less time to converge. Hence, ASQN is faster than AKQN. Comparing with ARN and RQN, we see that ASQN benefits from our quasi-Newton technique. Using a scaled identity matrix as the initial guess, RQN takes many more iterations than our algorithms which use the adaptive compressed form of the hybrid exchange operator. ASQN is two times faster than ACE in “graphene30” and “si40”. Finally, the convergence behaviors of these six methods on the system “glutamine” in Figure 2, where $\Delta E_{\text{hf}}(X^k)$ is defined similarly as the KS case. In summary, algorithms utilizing the quasi-Newton technique combining with the Nyström approximation is often able to give better performance.

7. Conclusion. We present a structured quasi-Newton method for optimization with orthogonality constraints. Instead of approximating the full Riemannian Hessian directly, we construct an approximation to the Euclidean Hessian and a regularized subproblem using this approximation while the orthogonality constraints are kept. By solving the subproblem inexactly, the global and local q-superlinear convergence can be guaranteed under certain assumptions. Our structured quasi-Newton method also takes advantage of the structure of the objective function if some parts are much more expensive to be evaluated than other parts. Our numerical experiments on the linear eigenvalue problems, KSDFT and HF total energy minimization demonstrate that our structured quasi-Newton algorithm is very competitive with the state-of-art algorithms.

The performance of the quasi-Newton methods can be further improved in several perspectives. For example, finding a better initial quasi-Newton matrix than the

TABLE 6
 Numerical results on HF total energy minimization.

solver	fval	nrmG	its	time	fval	nrmG	its	time
	alanine				c12h26			
ACE	-6.61821e+1	3.8e-7	11(3.0)	261.7	-8.83756e+1	3.9e-7	8(2.9)	259.7
GBBN	-6.61821e+1	1.0e-6	11(17.4)	268.8	-8.83756e+1	4.9e-4	200(68.7)	11839.8
ARN	-6.61821e+1	9.5e-7	10(13.7)	206.6	-8.83756e+1	4.9e-4	200(2.4)	4230.3
ASQN	-6.61821e+1	9.1e-7	7(14.1)	169.6	-8.83756e+1	2.1e-7	7(12.6)	234.1
AKQN	-6.61821e+1	4.8e-7	31(7.5)	530.2	-8.83756e+1	4.9e-7	29(7.6)	871.2
RQN	-6.61821e+1	1.9e-6	76	1428.5	-8.83756e+1	1.3e-3	45	3446.3
	ctube661				glutamine			
ACE	-1.43611e+2	9.2e-7	8(2.8)	795.0	-1.04525e+2	3.9e-7	10(3.0)	229.6
GBBN	-1.43611e+2	6.5e-7	10(26.3)	1399.2	-1.04525e+2	8.4e-7	11(13.3)	256.9
ARN	-1.43611e+2	6.0e-7	9(14.1)	832.7	-1.04525e+2	8.8e-7	10(9.5)	209.5
ASQN	-1.43611e+2	2.0e-7	8(13.2)	777.1	-1.04525e+2	1.5e-7	8(10.1)	182.9
AKQN	-1.43611e+2	6.1e-7	17(10.3)	1502.0	-1.04525e+2	9.1e-7	25(6.0)	515.7
RQN	-1.43611e+2	7.2e-6	59	6509.0	-1.04525e+2	2.9e-6	57	1532.8
	graphene16				graphene30			
ACE	-1.01716e+2	7.6e-7	13(3.4)	367.0	-1.87603e+2	8.6e-7	58(4.2)	14992.0
GBBN	-1.01716e+2	4.2e-7	14(42.1)	659.0	-1.87603e+2	8.9e-7	29(72.2)	19701.8
ARN	-1.01716e+2	4.5e-7	14(23.0)	403.6	-1.87603e+2	9.0e-7	45(35.6)	14860.6
ASQN	-1.01716e+2	4.9e-7	11(20.2)	357.5	-1.87603e+2	7.6e-7	15(26.5)	6183.0
AKQN	-1.01716e+2	7.9e-7	49(15.1)	1011.0	-1.87603e+2	8.0e-7	39(12.3)	9770.7
RQN	-1.01716e+2	1.0e-3	74	2978.9	-1.87603e+2	1.5e-5	110	39091.0
	pentacene				gaas			
ACE	-1.39290e+2	6.2e-7	13(3.0)	1569.5	-2.93496e+2	8.8e-7	29(2.9)	343.8
GBBN	-1.39290e+2	8.2e-7	16(23.0)	2620.2	-2.93496e+2	9.3e-7	34(35.3)	659.3
ARN	-1.39290e+2	7.2e-7	15(12.2)	1708.1	-2.93496e+2	9.6e-7	31(20.4)	468.7
ASQN	-1.39290e+2	1.9e-7	9(14.3)	1168.1	-2.93496e+2	3.3e-7	10(28.0)	199.5
AKQN	-1.39290e+2	5.4e-7	29(8.5)	3458.4	-2.93496e+2	4.6e-7	22(18.4)	347.1
RQN	-1.39290e+2	2.4e-6	73	11363.8	-2.93496e+2	1.0e-6	126	2154.1
	si40				si64			
ACE	-1.65698e+2	9.2e-7	29(4.5)	30256.4	-2.67284e+2	9.8e-7	9(2.9)	6974.3
GBBN	-1.65698e+2	8.6e-7	24(43.9)	34692.4	-2.67284e+2	5.3e-7	14(27.0)	11467.9
ARN	-1.65698e+2	8.0e-7	22(22.1)	21181.3	-2.67284e+2	7.7e-7	12(18.6)	9180.7
ASQN	-1.65698e+2	2.8e-7	12(37.8)	15369.5	-2.67284e+2	3.0e-7	8(21.9)	6764.7
AKQN	-1.65698e+2	9.2e-7	87(7.9)	89358.8	-2.67284e+2	7.1e-7	24(18.8)	33379.0
RQN	-1.65698e+2	6.1e-6	156	181976.8	-2.67284e+2	8.4e-7	112	115728.8

Nyström approximation and developing a better quasi-Newton approximation than the LSR1 technique. Our technique can also be extended to the general Riemannian optimization with similar structures.

REFERENCES

- [1] P.-A. ABSIL, C. G. BAKER, AND K. A. GALLIVAN, *Trust-region methods on Riemannian manifolds*, *Found. Comput. Math.*, 7 (2007), pp. 303–330.
- [2] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization algorithms on matrix manifolds*, Princeton University Press, Princeton, NJ, 2008.
- [3] P.-A. ABSIL, R. MAHONY, AND J. TRUMPF, *An extrinsic look at the Riemannian Hessian*, in *Geometric science of information*, Springer, 2013, pp. 361–368.
- [4] A. D. BECKE, *Density-functional thermochemistry. III. the role of exact exchange*, *J. Chem.*

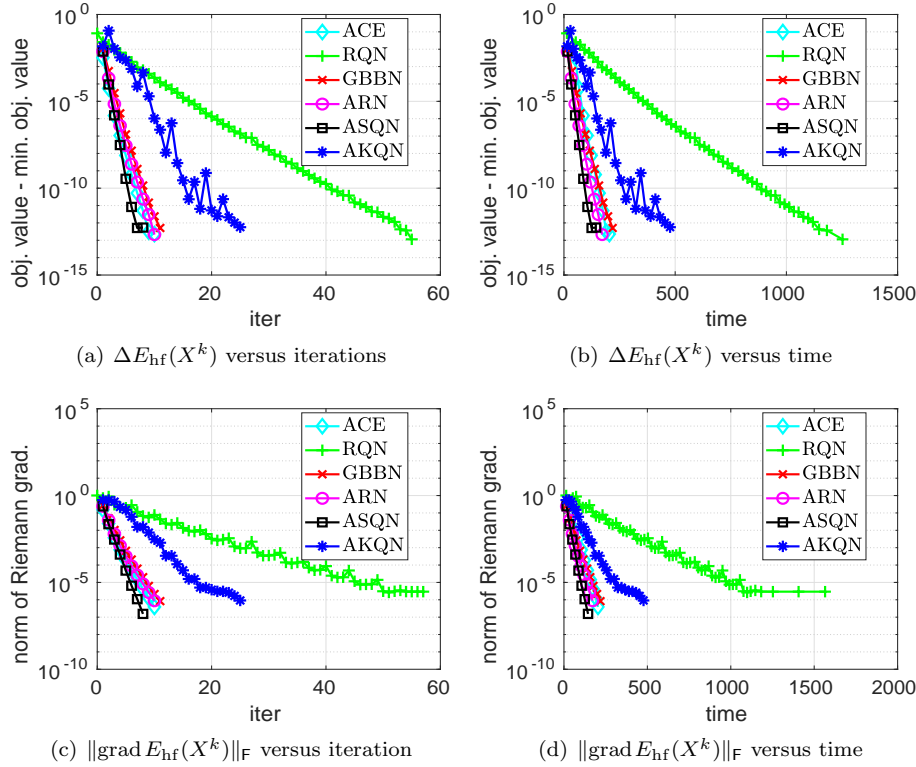


FIG. 2. Comparisons of different algorithms on “glutamine” of HF total energy minimization.

- Phys., 98 (1993), pp. 5648–5652.
- [5] N. BOUMAL, P.-A. ABSIL, AND C. CARTIS, *Global rates of convergence for nonconvex optimization on manifolds*, IMA J. Numer. Anal., (2016).
 - [6] N. BOUMAL, B. MISHRA, P.-A. ABSIL, AND R. SEPULCHRE, *Manopt, a Matlab toolbox for optimization on manifolds*, J. Mach. Learn. Res., 15 (2014), pp. 1455–1459, <http://www.manopt.org>.
 - [7] R. H. BYRD, H. F. KHALFAN, AND R. B. SCHNABEL, *Analysis of a symmetric rank-one trust region method*, SIAM J. Optim., 6 (1996), pp. 1025–1039.
 - [8] R. H. BYRD, M. MARAZZI, AND J. NOCEDAL, *On the convergence of Newton iterations to non-stationary points*, Math. Program., 99 (2004), pp. 127–148.
 - [9] R. H. BYRD, J. NOCEDAL, AND R. B. SCHNABEL, *Representations of quasi-Newton matrices and their use in limited memory methods*, Math. Program., 63 (1994), pp. 129–156.
 - [10] A. EDELMAN, T. A. ARIAS, AND S. T. SMITH, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 303–353.
 - [11] D. GABAY, *Minimizing a differentiable function over a differential manifold*, J. Optim. Theory Appl., 37 (1982), pp. 177–219.
 - [12] P. GIANNOZZI, S. BARONI, N. BONINI, M. CALANDRA, R. CAR, C. CAVAZZONI, D. CERESOLI, G. L. CHIAROTTI, M. COCOCIONI, I. DABO, ET AL., *Quantum espresso: a modular and open-source software project for quantum simulations of materials*, Journal of Physics: Condensed matter, 21 (2009), p. 395502.
 - [13] S. GRATTON AND P. L. TOINT, *Multi-secant equations, approximate invariant subspaces and multigrid optimization*, tech. report, Dept of Mathematics, FUNDP, Namur (B), 2007.
 - [14] D. HAMANN, *Optimized norm-conserving Vanderbilt pseudopotentials*, Phys. Rev. B, 88 (2013), p. 085117.
 - [15] J. HEYD, G. E. SCUSERIA, AND M. ERNZERHOF, *Hybrid functionals based on a screened Coulomb*

- potential*, J. Chem. Phys., 118 (2003), pp. 8207–8215.
- [16] J. HU, A. MILZAREK, Z. WEN, AND Y. YUAN, *Adaptive quadratically regularized Newton method for Riemannian optimization*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1181–1207.
- [17] W. HU, L. LIN, AND C. YANG, *Projected commutator DIIS method for accelerating hybrid functional electronic structure calculations*, J. Chem. Theory Comput., 13 (2017), pp. 5458–5467.
- [18] W. HUANG, *Optimization algorithms on Riemannian manifolds with applications*, PhD thesis, The Florida State University, 2013.
- [19] W. HUANG, P. ABSIL, K. GALLIVAN, AND P. HAND, *ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds*, tech. report, Technical Report FSU16-14, Florida State University, 2016.
- [20] W. HUANG, P.-A. ABSIL, AND K. GALLIVAN, *A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems*, SIAM J. Optim., 28 (2018), pp. 470–495.
- [21] W. HUANG, P.-A. ABSIL, AND K. A. GALLIVAN, *A Riemannian symmetric rank-one trust-region method*, Math. Program., 150 (2015), pp. 179–216.
- [22] W. HUANG, P.-A. ABSIL, AND K. A. GALLIVAN, *A Riemannian BFGS method for nonconvex optimization problems*, in Numerical Mathematics and Advanced Applications ENUMATH 2015, Springer, 2016, pp. 627–634.
- [23] W. HUANG, K. A. GALLIVAN, AND P.-A. ABSIL, *A Broyden class of quasi-Newton methods for Riemannian optimization*, SIAM J. Optim., 25 (2015), pp. 1660–1685.
- [24] R. E. KASS, *Nonlinear regression analysis and its applications*, J. Am. Stat. Assoc., 85 (1990), pp. 594–596.
- [25] A. V. KNYAZEV, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
- [26] K. KREUTZ-DELGADO, *The complex gradient operator and the CR-calculus*, 2009. <http://arxiv.org/abs/0906.4835>.
- [27] C. LE BRIS, *Computational chemistry from the perspective of numerical analysis*, Acta Numer., 14 (2005), pp. 363–444.
- [28] L. LIN, *Adaptively compressed exchange operator*, J. Chem. Theory Comput., 12 (2016), pp. 2242–2249.
- [29] L. LIN AND M. LINDSEY, *Convergence of adaptive compression methods for Hartree-Fock-like equations*, Commun. Pure Appl. Math., in press, (2017).
- [30] X. LIU, Z. WEN, AND Y. ZHANG, *Limited memory block Krylov subspace optimization for computing dominant singular value decompositions*, SIAM J. Sci. Comput., 35 (2013), pp. A1641–A1668.
- [31] R. M. MARTIN, *Electronic structure: basic theory and practical methods*, Cambridge university press, 2004.
- [32] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.
- [33] C. QI, *Numerical optimization methods on Riemannian manifolds*, PhD thesis, Florida State University, 2011.
- [34] W. RING AND B. WIRTH, *Optimization methods on Riemannian manifolds and their application to shape space*, SIAM J. Optim., 22 (2012), pp. 596–627.
- [35] M. SEIBERT, M. KLEINSTEUBER, AND K. HÜPER, *Properties of the BFGS method on Riemannian manifolds*, Mathematical System Theory C Festschrift in Honor of Uwe Helmke on the Occasion of his Sixtieth Birthday, (2013), pp. 395–412.
- [36] S. T. SMITH, *Optimization techniques on Riemannian manifolds*, Fields Institute Communications, 3 (1994).
- [37] W. SUN AND Y. YUAN, *Optimization theory and methods: nonlinear programming*, vol. 1, Springer Science & Business Media, 2006.
- [38] A. SZABO AND N. S. OSTLUND, *Modern quantum chemistry: introduction to advanced electronic structure theory*, Courier Corporation, 2012.
- [39] L. THOGERSEN, J. OLSEN, A. KOHN, P. JORGENSEN, P. SALEK, AND T. HELGAKER, *The trust-region self-consistent field method in Kohn–Sham density functional theory*, J. Chem. Phys., 123 (2005), p. 074103.
- [40] J. A. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Fixed-rank approximation of a*

- positive-semidefinite matrix from streaming data*, in Advances in Neural Information Processing Systems, 2017, pp. 1225–1234.
- [41] C. UDRISTE, *Convex functions and optimization methods on Riemannian manifolds*, vol. 297, Springer Science & Business Media, 1994.
- [42] R. VAN LEEUWEN, *Density functional approach to the many-body problem: key concepts and exact functionals*, Adv. Quantum Chem., 43 (2003), pp. 25–94.
- [43] Z. WEN, A. MILZAREK, M. ULBRICH, AND H. ZHANG, *Adaptive regularized self-consistent field iteration with exact Hessian for electronic structure calculation*, SIAM J. Sci. Comput., 35 (2013), pp. A1299–A1324.
- [44] Z. WEN AND W. YIN, *A feasible method for optimization with orthogonality constraints*, Math. Program., 142 (2013), pp. 397–434.
- [45] C. YANG, J. C. MEZA, B. LEE, AND L.-W. WANG, *KSSOLV—a MATLAB toolbox for solving the Kohn-Sham equations*, ACM Trans. Math. Softw., 36 (2009), pp. 1–35.
- [46] 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.
- [47] W. ZHOU AND X. CHEN, *Global convergence of a new hybrid Gauss–Newton structured BFGS method for nonlinear least squares problems*, SIAM J. Optim., 20 (2010), pp. 2422–2441.