SIAM J. SCI. COMPUT.
Vol. 41, No. 4, pp. A2239–A2269

STRUCTURED QUASI-NEWTON METHODS FOR OPTIMIZATION WITH ORTHOGONALITY CONSTRAINTS*

JIANG HU†, BO JIANG‡, LIN LIN§, ZAIWEN WEN†, AND YA-XIANG 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 specifically, the part related to the Euclidean gradient and the cheaper parts in the Euclidean Hessian are preserved. The initial quasi-Newton matrix is further constructed from a limited-memory Nystrom 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 Nystrom approximation, Hartree–Fock total energy minimization

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

DOI. 10.1137/18M121112X

1. Introduction.

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

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

where $f(X) : \mathbb{C}^{n\times p} \to \mathbb{R}$ is a $\mathbb{R}$-differentiable function [31]. Although our proposed methods are applicable to a general function $f(X)$, we are in particular interested in

*Submitted to the journal's Methods and Algorithms for Scientific Computing section September 4, 2018; accepted for publication (in revised form) April 16, 2019; published electronically July 23, 2019.

https://doi.org/10.1137/18M121112X

Funding: The work of the second author was supported by NSFC grants 11501298 and 11671036, the Young Elite Scientists Sponsorship Program by CAST (2017QNRC001), and by the NSF of Jiangsu Province (BK20150965). The work of the third author was supported by the National Science Foundation under grant DMS-1652330, the Department of Energy under grants DE-SC0017867 and DE-AC02-05CH11231, and the SciDAC project. The work of the fourth author was supported by NSFC grants 11831002, 11421101, and 91730302, and by the National Basic Research Project under grant 2015CB856002. The work of the fifth author was supported by NSFC grants 11331012 and 11461161005.

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

‡School of Mathematical Sciences, Key Laboratory for NSLSCS of Jiangsu Province, Nanjing Normal University, Nanjing 210023, China (jiangbo@njnu.edu.cn).

§Department of Mathematics, University of California, Berkeley, Berkeley, CA, 94720-3840 (linlin@math.berkeley.edu).

¶Institute of Computational Math and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, China (yyx@lsec.cc.ac.cn).

the cases that the Euclidean Hessian $\nabla^2 f(X)$ takes a natural structure as

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

where the computational cost of $\mathcal{H}^{\mathrm{e}}(X)$ is much more expensive than that of $\mathcal{H}^{\mathrm{c}}(X)$. This situation occurs when f is a summation of functions whose full Hessian are expensive to be evaluated or even not accessible. A practical example is the Hartree–Fock-like total energy minimization problem in the electronic structure theory [44, 36], 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 [13, 12, 47, 42, 1, 2, 50]. 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 the manifold. By adding some restrictions between differentiable retraction and vector transport, a Riemannian Broyden–Fletcher–Goldfarb–Shanno (BFGS) method is presented in [39, 40, 41]. 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 [22, 25, 27, 24] develop a new class of Riemannian BFGS methods and symmetric rank-one and Broyden family methods. Moreover, a selection of Riemannian quasi-Newton methods has been implemented in the software package Manopt [6] and ROPTLIB [23].

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. When using the Euclidean metric (it will be introduced in subsection 1.4) as the Riemannian metric, we can write the Riemannian Hessian $\mathrm{Hess}\, f(X)$ as

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

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 $\mathrm{Proj}_X(Z) := Z - X\mathrm{sym}(X^*Z)$ is the projection of Z onto the tangent space T_X and $\mathrm{sym}(A) := (A + A^*)/2$. See [12, 3] for details on the structure (1.3). We briefly summarize our contributions as follows.

- By taking advantage of this structure (1.3), we construct an approximation to the Euclidean Hessian $\nabla^2 f(X)$ instead of the full Riemannian Hessian $\mathrm{Hess}\, f(X)$ directly, but keep the remaining parts $\xi\mathrm{sym}(X^*\nabla f(X))$ and $\mathrm{Proj}_X(\cdot)$. 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 [20], 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 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 $\mathcal{H}^{\mathrm{e}}$ while preserving the cheap part $\mathcal{H}^{\mathrm{c}}$. 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 $\mathcal{H}^{\mathrm{e}}$, we also investigate a limited-memory Nyström approximation, which gives a subspace approximation of a known good but still complicated approximation of $\mathcal{H}^{\mathrm{e}}$.
- When the subproblems are solved to a 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) [19, 30], 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 an orthogonal set of electronic wave functions. The mathematical structure of Kohn–Sham equations depends heavily on the choice of the exchange-correlation functional. In particular, the Kohn–Sham Hamiltonian with a local or semilocal exchange-correlation functional is a differential operator. On the other hand, hybrid exchange-correlation functionals [4, 18] are known to provide a more accurate model to electronic structure calculations. When hybrid exchange-correlation functionals are used, the Kohn–Sham Hamiltonian becomes an integro-differential operator. The Kohn–Sham equations become Hartree–Fock-like equations. The computational cost of hybrid functional calculations is usually much more expensive than those using local and semilocal functionals. Existing optimization based methods often do not efficiently use the structure of the Hessian matrix in these calculations. In this paper, by exploiting the structure of the Hessian, we apply our structured quasi-Newton method to solve these problems. Numerical experiments show that our algorithm performs at least comparably well with state-of-art methods in their convergent case. In the case where 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, and real and imaginary parts, respectively. Let $\mathrm{span}\{X_1, \ldots, X_l\}$ be the space spanned by the matrices $X_1, \ldots, X_l$. Let $[X_1, \ldots, X_l] \in \mathbb{R}^{n\times(lp)}$ be a matrix with columns $X_1, \ldots, X_l$. The vector denoted $\mathrm{vec}(X)$ in $\mathbb{C}^{np}$ is formulated by stacking each column of X one by one, from the first

to the last column; the operator $\mathrm{mat}(\cdot)$ is the inverse of $\mathrm{vec}(\cdot)$, i.e., $\mathrm{mat}(\mathrm{vec}(X)) = X$. Given two matrices $A, B \in \mathbb{C}^{n\times p}$, the Frobenius inner product $\langle\cdot,\cdot\rangle$ is defined as $\langle A, B\rangle = \mathrm{tr}(A^*B)$, and the corresponding Frobenius norm $\|\cdot\|_{\mathsf{F}}$ is defined as $\|A\|_{\mathsf{F}} = \sqrt{\mathrm{tr}(A^*A)}$. The Euclidean metric is defined as the real part of the Frobenius inner product, i.e., $\Re\langle A, B\rangle$. For a matrix $M \in \mathbb{C}^{n\times n}$, the operator $\mathrm{diag}(M)$ is a vector in $\mathbb{C}^n$ formulated by the main diagonal of M; and for $c \in \mathbb{C}^n$, the operator $\mathrm{Diag}(c)$ 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 $\mathrm{St}(n,p) := \{X \in \mathbb{C}^{n\times p} : X^*X = I_p\}$ be the (complex) Stiefel manifold. With the Euclidean metric (i.e., the Riemannian metric used on $\mathrm{St}(n,p)$), $\nabla f(X)$ (resp., $\nabla^2 f(X)$) and $\mathrm{grad}\, f(X)$ (resp., $\mathrm{Hess}\, f(X)$) denote the Euclidean and Riemannian gradient (resp., Hessian) of f at X. 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 [38, Chapter 6] can be used to construct an approximation to $\nabla^2 f(X)$ via the gradients $\nabla f(X)$. Since the approximate Hessian is of low computational cost, it is possible that they 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 [53] for the unconstrained nonlinear least square problems [28], [43, Chapter 7]. Then our subproblem at the kth iteration is constructed as

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

where

$$m_k(X) := \Re\left\langle \nabla f(X^k), X - X^k\right\rangle + \frac{1}{2}\Re\left\langle \mathcal{B}^k[X - X^k], X - X^k\right\rangle + \frac{\tau_k}{2} d(X, X^k) \tag{2.2}$$

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 [49, section 1.1] for details. Here, τ_k is a regularization parameter and $d(X, X^k)$ is a proximal term. The choice of τ_k is crucial to control the distance between X and X^k. Solving subproblem (2.1) with a well chosen τ_k can lead to a sufficient decrease of the objective function of the original problem (1.1). Therefore, by developing a proper rule of updating τ_k, we can construct a class of algorithms with convergence guarantees.

The proximal term can be chosen as the quadratic regularization

$$d(X, X^k) = \|X - X^k\|_{\mathsf{F}}^2 \tag{2.3}$$

or the cubic regularization [37, 10, 11]

$$d(X, X^k) = \frac{2}{3}\|X - X^k\|_{\mathsf{F}}^3, \tag{2.4}$$

which is shown to be useful in the construction of subproblems. In the following, we will mainly focus on the quadratic regularization (2.3). Due to the Stiefel manifold constraint, the quadratic regularization (2.3) is actually equal 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

$$\text{Hess}\, m_k(X^k)[\xi] = \text{Proj}_{X^k}\left(\mathcal{B}^k[\xi] - \xi\text{sym}((X^k)^*\nabla f(X^k)\right) + \tau_k\xi,\ \xi \in T_{X^k}. \tag{2.5}$$

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 [23, 25, 26, 27] focus on constructing an approximation to the Riemannian Hessian 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 sym $\left((X^k)^*\nabla f(X^k)\right)$ 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 $\text{Proj}_{X^k}\left(\mathcal{B}^k[\xi] - \xi\text{sym}((X^k)^*\nabla f(X^k)\right)$ of 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

$$\mathcal{B}^k[S^k] = \nabla f(X^k) - \nabla f(X^{k-1}), \tag{2.6}$$

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 $\mathcal{H}^{\text{c}}(X)$ while only approximating $\mathcal{H}^{\text{e}}(X)$. Specifically, we derive the approximation $\mathcal{B}^k$ to the Hessian $\nabla^2 f(X^k)$ as

$$\mathcal{B}^k = \mathcal{H}^{\text{c}}(X^k) + \mathcal{E}^k, \tag{2.7}$$

where $\mathcal{E}^k$ is an approximation to $\mathcal{H}^{\text{e}}(X^k)$. Substituting (2.7) into (2.6), we can see that the approximation $\mathcal{E}^k$ should satisfy the following revised secant condition:

$$\mathcal{E}^k[S^k] = Y^k, \tag{2.8}$$

where

$$Y^k := \nabla f(X^k) - \nabla f(X^{k-1}) - \mathcal{H}^{\text{c}}(X^k)[S^k]. \tag{2.9}$$

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 $\mathcal{H}^{\text{e}}(X^k)$ itself may not be positive definite even when X^k is optimal, we utilize the limited-memory symmetric rank-one (LSR1) scheme to approximate $\mathcal{H}^{\text{e}}(X^k)$ such that it satisfies the secant equation (2.8).

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

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

Let $\mathcal{E}_0^k : \mathbb{C}^{n\times p} \to \mathbb{C}^{n\times p}$ be the initial approximation of $\mathcal{H}^{\text{e}}(X^k)$ and define the $(np)\times l$ matrix $\Sigma^{k,m} := \left[\text{vec}\left(\mathcal{E}_0^k[S^{k-l}]\right), \ldots, \text{vec}\left(\mathcal{E}_0^k[S^{k-1}]\right)\right]$. Let $F^{k,m}$ be a matrix in $\mathbb{C}^{l\times l}$

with $(F^{k,m})_{i,j} = \left\langle S^{k-l+i-1}, Y^{k-l+j-1} \right\rangle$ for $1 \le i, j \le l$. Under the assumption that $\left\langle S^j, \mathcal{E}^j[S^j] - Y^j \right\rangle \neq 0$, $j = k-l, \ldots, k-1$, 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

$$\mathcal{E}^k[U] = \mathcal{E}_0^k[U] + \operatorname{mat}\left(N^{k,m} \left(F^{k,m} - (S^{k,m})^* \Sigma^{k,m} \right)^{-1} (N^{k,m})^* \operatorname{vec}(U) \right), \tag{2.10}$$

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

$$\left| \left\langle S^j, \mathcal{E}^j[S^j] - Y^j \right\rangle \right| \le r \|S^j\|_{\mathsf{F}} \|\mathcal{E}^j[S^j] - Y^j\|_{\mathsf{F}}$$

with small number r, say, $r = 10^{-8}$. A similar idea can be found in [38, section 6.2].

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 $\mathcal{H}^{\mathrm{e}}(X^k)$ is known but its computational cost is still very high. We next explain how to use the limited-memory Nyström approximation to construct another approximation with lower computational cost based on $\mathcal{E}_0^k$.

Specially, let Ω 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:

$$\hat{\mathcal{E}}_0^k[U] \coloneqq W(W^*\Omega)^\dagger W^* U, \tag{2.11}$$

where $U \in \mathbb{C}^{n \times p}$ is any direction. This is called the limited-memory Nyström approximation; see [46] 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 (2.8) 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

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

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 that $\hat{\mathcal{E}}_0^k$ will satisfy the secant condition whenever $\mathcal{E}_0^k$ does. From the methods for linear eigenvalue computation in [29] and [35], the subspace $\mathfrak{S}$ can also be determined as

$$\operatorname{span}\{X^{k-1}, X^k, \mathcal{E}_0^k[X^k]\} \quad \text{or} \quad \operatorname{span}\{X^{k-h}, \ldots, X^{k-1}, X^k\} \tag{2.12}$$

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 pseudoinverse 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 there are 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 [20], we can compute a trial point Z^k by utilizing either the Riemannian gradient type method (see section 2.1 in [20]) or a modified CG method (Algorithm 1 in [20]) to solve the subproblem inexactly. Specifically, the Riemannian Newton equation of (2.1) at X^k is

$$\operatorname{grad} m_k(X^k) + \operatorname{Hess} m_k(X^k)[\xi] = 0, \quad \xi \in T_{X^k}, \tag{2.13}$$

where $\operatorname{grad} m_k(X^k) = \operatorname{grad} f(X^k)$ and $\operatorname{Hess} m_k(X^k)$ is given in (2.5). Based on (2.13), we compute a descent direction ξ^k and do an Armijo search along a curve introduced by ξ^k on the manifold. Hence, the trial point Z^k always stays on the manifold and leads to a sufficient decrease on m_k. After obtaining the trial point Z^k of (2.1), we calculate the ratio between the predicted reduction and the actual reduction

$$r_k = \frac{f(Z^k) - f(X^k)}{m_k(Z^k)}. \tag{2.14}$$

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,

$$X^{k+1} = \begin{cases} Z^k & \text{if } r_k \geq \eta_1, \\ X^k & \text{otherwise.} \end{cases} \tag{2.15}$$

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

$$\tau_{k+1} \in \begin{cases} (0, \gamma_0 \tau_k] & \text{if } r_k \geq \eta_2, \\ [\gamma_0 \tau_k, \gamma_1 \tau_k] & \text{if } \eta_1 \leq r_k < \eta_2, \\ [\gamma_1 \tau_k, \gamma_2 \tau_k] & \text{otherwise,} \end{cases} \tag{2.16}$$

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

From [8], the Newton-type method may still be very slow when the Hessian is close to being singular. Numerically, it may happen that the regularization parameter turns out 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 $\tilde{\mathfrak{S}}^k := \operatorname{span}\{X^k, \operatorname{grad} f(X^k)\}$, 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 [29] and [35], the subspace $\mathfrak{S}^k$ can be decided by using the history information $\{X^k, X^{k-1}, \ldots, \}$ $\{\operatorname{grad} f(X^k), \operatorname{grad} f(X^{k-1}), \ldots\}$ 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,

$$\min_{M\in\mathbb{C}^{q\times p}} \quad f(Q^k M) \quad \text{s.t.} \quad M^*M = I_p. \tag{2.17}$$

The problem (2.17) can be solved inexactly by existing methods for optimization with orthogonality constraints. Once a good approximate solution M^k of (2.17) 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 iterate 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 $\tau_0 > 0$, $0 < \eta_1 \le \eta_2 < 1$, $1 < \gamma_1 \le \gamma_2$. Set $k = 0$.
while *stopping conditions not met* **do**
  Choose $\mathcal{E}_0^k$ (by the limited-memory Nyström approximation if necessary).
  Construct the approximation $\mathcal{B}^k$ via (2.7) and (2.10).
  Construct and solve the subproblem (2.1) (by using the modified CG method (Algorithm 1 in [20]) or the Riemannian gradient type method (see section 2.1 in [20])) to obtain a new trial point Z^k.
  Compute the ratio r_k via (2.14).
  Update X^{k+1} from the trial point Z^k based on (2.15).
  Update τ_k according to (2.16).
  $k \leftarrow k + 1$.
  if *stagnate conditions met* **then**
    Solve the subspace problem (2.17) to update X^{k+1}.

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

$$m_k(Z^k) \le -\frac{a}{b+\tau_k}\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}^2, \tag{3.1}$$

$$\|\operatorname{grad} m_k(Z^k)\|_{\mathsf{F}} \le \theta^k \|\operatorname{grad} f(X^k)\|_{\mathsf{F}}, \tag{3.2}$$

where a, b are positive constants and $\theta^k := \min\{1, \|\operatorname{grad} f(X^k)\|_{\mathsf{F}}^c\}$ with $c > 0$. Here, the inexact condition (3.1) is to guarantee the decrease for each iteration and hence

the global convergence. The inequality (3.1) can be satisfied by one-step Riemannian gradient descent, in which the coefficient $\frac{a}{b+\tau_k}$ is from the bounds of the first- and second-order derivatives of m_k and the retraction operator [5, Lemma 2.10]. We will present the specific choices of a and b for the modified CG method in Lemma 3. The inequality (3.2) is to control how inexactly we solve the subproblem (2.1). With this choice of θ^k, we can guarantee fast local convergence. When the stagnate conditions in Algorithm 1 are met, we need to perform the subspace refinement procedure, namely, to solve the extra problem (2.17). Note that X^k and $\operatorname{grad} f(X^k)$ are always contained in the subspace used in (2.17), and a sufficient decrease for the original problem (i.e., a descent step) can be guaranteed, which is enough to ensure the global convergence to a stationary point. For the local convergence part, since we assume that the Riemannian Hessian is positive definite, the stagnate conditions will not be satisfied if the approximation $\mathcal{B}^k$ is properly constructed. Throughout the analysis of convergence, we assume that the stagnate conditions are never met.

3.1. Global convergence to a stationary point. Since the regularization term is used, the global convergence (i.e., convergence starting from any initial point) to a stationary point 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 the following:

(A1) The gradient ∇f is Lipschitz continuous on the convex hull of $\mathrm{St}(n,p)$ [14], i.e., there exists $L_f > 0$ such that

$$\|\nabla f(X) - \nabla f(Y)\|_{\mathsf{F}} \le L_f \|X - Y\|_{\mathsf{F}} \quad \forall\, X, Y \in \operatorname{conv}(\mathrm{St}(n,p)).$$

(A2) There exists $\kappa_H > 0$ such that $\|\mathcal{B}^k\| \le \kappa_H$ 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 $\kappa_g \ge 1$ on the compact set $\operatorname{conv}(\mathrm{St}(n,p))$, i.e.,

$$\|\nabla f(X)\|_{\mathsf{F}} \le \kappa_g,\ X \in \operatorname{conv}(\mathrm{St}(n,p)).$$

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

We first prove that the inexact condition (3.1) is satisfied by the modified CG method.

LEMMA 3. *Suppose that assumptions* (A1)–(A2) *hold. The modified CG method, always returns a trial point* Z^k *satisfying the inexact condition* (3.1).

Proof. From Assumption 1 and Remark 2, the Riemannian Hessian $\operatorname{Hess} m(X^k)$ can be bounded by

$$\|\operatorname{Hess} m_k(X^k)\| \le \|\mathcal{B}^k\| + \|X^k\|\|\nabla f(X^k)\|_{\mathsf{F}} + \tau_k \le \kappa_H + \kappa_g + \tau_k, \tag{3.3}$$

where $\|X^k\| = 1$ because of its unitary property. We note that the bound of the spectrum of $\operatorname{Hess} m(X^k)$ is obtained without requiring the boundedness of $\|\operatorname{Hess} f(X^k)\|_{\mathsf{F}}$

(assumed in [20]) since we are working on the compact Stiefel manifold. Similar to [20, Lemma 5], a descent direction ξ^k can be obtained via solving the Riemannian Newton equation (2.13) from the modified CG method, i.e.,

$$\frac{\Re\left\langle \operatorname{grad} f(X^k), \xi^k \right\rangle}{\|\operatorname{grad} f(X^k)\|_{\mathsf{F}} \|\xi^k\|_{\mathsf{F}}} \leq -\min\left\{\frac{\epsilon}{2}, 1\right\} \frac{1}{2np(\kappa_H + \kappa_g + 1)} =: -\kappa_0, \tag{3.4}$$

where (3.3) is used, $\epsilon > 0$ is a constant used in the modified CG method, and $2np$ is from the dimension of the complex Stiefel manifold $\mathrm{St}(n,p)$ (its dimension is $2np - p^2$). Following [5, Lemma 2.7] and [20, Lemma 6], we shall show how the inexact condition (3.1) is satisfied. Since $\mathrm{St}(n,p)$ is compact, there exist two positive constants α_1, α_2 such that (see [5, equations (B.3) and (B.4)]), for all $X \in \mathrm{St}(n,p)$ and for all $\xi \in T_X$,

$$\begin{aligned} \|R_X(\xi) - X\|_{\mathsf{F}} &\leq \alpha_1 \|\xi\|_{\mathsf{F}}, \\ \|R_X(\xi) - X - \xi\|_{\mathsf{F}} &\leq \alpha_2 \|\xi\|_{\mathsf{F}}^2, \end{aligned} \tag{3.5}$$

where R is a retraction [2, Definition 4.1.1]. Following the proofs in [5, Lemma 2.7], we know that $m_k(R_{X^k}(t\xi^k))$ is upper bounded by a quadratic function

$$m_k(R_{X^k}(t\xi^k)) \leq t\Re\left\langle \operatorname{grad} f(X^k), \xi^k \right\rangle + \left(\frac{\kappa_H + \tau_k}{2}\alpha_1^2 + \kappa_g \alpha_2\right) t^2 \|\xi^k\|_{\mathsf{F}}^2.$$

Then, for any constant $\rho \in (0,1)$,

$$\begin{aligned} & m_k(R_{X^k}(t\xi^k)) - \rho t \Re\left\langle \operatorname{grad} f(X^k), \xi^k \right\rangle \\ \leq & -(1-\rho)\kappa_0 t \|\operatorname{grad} f(X^k)\|_{\mathsf{F}} \|\xi^k\|_{\mathsf{F}} + \left(\frac{\kappa_H + \tau_k}{2}\alpha_1^2 + \kappa_g \alpha_2\right) t^2 \|\xi^k\|_{\mathsf{F}}^2 \\ = & \left[\left(\frac{\kappa_H + \tau_k}{2}\alpha_1^2 + \kappa_g \alpha_2\right) t - (1-\rho)\kappa_0 \frac{\|\operatorname{grad} m_k(X^k)\|_{\mathsf{F}}}{\|\xi^k\|_{\mathsf{F}}}\right] \cdot t \|\xi^k\|_{\mathsf{F}}^2, \end{aligned}$$

where (3.4) is used in the first inequality. We have

$$m_k(R_{X^k}(t\xi^k)) \leq \rho t \Re\left\langle \operatorname{grad} f(X^k), \xi^k \right\rangle \quad \forall t \in [0, \chi^k], \tag{3.6}$$

where

$$\chi^k := \frac{2(1-\rho)\kappa_0 \|\operatorname{grad} f(X^k)\|_{\mathsf{F}}}{\left((\kappa_H + \tau_k)\alpha_1^2 + 2\kappa_g \alpha_2\right) \|\xi^k\|_{\mathsf{F}}}.$$

Define $\alpha_0 := \|\operatorname{grad} f(X^k)\|_{\mathsf{F}}^2 / \left\langle \operatorname{Hess} m_k(X^k)[\operatorname{grad} f(X^k)], \operatorname{grad} f(X^k) \right\rangle$. From the construction of the modified CG method [20, Algorithm 1], we have $\xi_k = -\operatorname{grad} f(X^k)$ if $1/\alpha_0 \leq \epsilon$. When $1/\alpha_0 > \epsilon$, it follows from (3.3) and the monotonicity in [20, Lemma 4(iii)] that

$$\|\xi^k\|_{\mathsf{F}} \geq \frac{1}{\kappa_H + \kappa_g + \tau_k} \|\operatorname{grad} f(X^k)\|_{\mathsf{F}}. \tag{3.7}$$

Hence, the inequality (3.7) always holds whether $1/\alpha_0 \leq \epsilon$ or not. From the Armjio curvilinear search [20, equation (3.9)] (we use 1 and constant $\sigma \in (0,1)$ as the initial step size and the decreasing factor of the step size, respectively), when $1 \leq \chi^k$, the decrease induced by the trial point $Z^k = R_{X^k}(\xi^k)$ satisfying

$$m_k(R_{X^k}(Z^k)) \leq \rho \Re\left\langle \operatorname{grad} f(X^k), \xi^k \right\rangle \leq -\frac{\rho\kappa_0}{\kappa_H + \kappa_g + \tau_k} \|\operatorname{grad} f(X^k)\|_{\mathsf{F}}^2,$$

where (3.4) and (3.7) are used in the second inequality. Otherwise, the accepted step t^k must be larger than $\sigma\chi^k$ and the decrease on m_k yields

$$m_k(R_{X^k}(Z^k)) \leq -\frac{2\sigma\rho(1-\rho)\kappa_0^2}{((\kappa_H+\tau_k)\alpha_1^2+2\kappa_g\alpha_2)}\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}^2,$$

in which (3.4) and (3.6) are used. By setting

$$a=\min\left\{\rho\kappa_0,\ \frac{2\sigma\rho(1-\rho)\kappa_0^2}{\alpha_1^2}\right\},\ b=\max\left\{\kappa_H+\kappa_g,\ \kappa_H+\frac{2\kappa_g\alpha_2}{\alpha_1^2}\right\},$$

we conclude that the inexact condition (3.1) always holds by the modified CG method. ∎

Based on the similar proof in [20, 49], we have the following theorem for global convergence to a stationary point.

THEOREM 4. *Suppose that assumptions* (A1)–(A2) *and the inexact condition* (3.1) *hold. Then, either*

$$\operatorname{grad} f(X^t)=0 \quad \textit{for some } t>0 \quad \textit{or} \quad \lim_{k\to\infty}\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}=0. \tag{3.8}$$

Proof. We prove it by contradiction. Assume that $\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}\geq\varsigma>0$ for all k. Following the proof in [20, Lemmas 7–9] the inexact condition (3.1) is sufficient to guarantee the ratio $r_k\geq\eta_2$ in (2.14) when τ_k is larger than a finite number L_ς. Therefore, there are infinitely many iterations with $r_k\geq\eta_2$ which leads to $\lim_{k\to\infty}f(X^k)=-\infty$. This contradicts the lower boundedness of $\{f(X^k)\}$. It follows from [20, Theorem 11, Remark 12] that the convergence result (3.8) holds due to the compactness of the Stiefel manifold. ∎

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

Assumption 5. Let $\{X^k\}$ be the sequence generated by Algorithm 1 without subspace refinement. We assume the following:

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

$$\frac{\|(\mathcal{B}^k-\nabla^2 f(X^k))[Z^k-X^k]\|_{\mathsf{F}}}{\|Z^k-X^k\|_{\mathsf{F}}}\to 0,\ k\to\infty. \tag{3.9}$$

From [20], the trial point Z^k obtained by the modified CG method will locally satisfy the inexact condition (3.2) if $\|\mathcal{B}^k-\nabla^2 f(X^k)\|\to 0, k\to\infty$, as in the symmetric rank-one method [7]. Under the assumption (B4), the inexact condition (3.2) may not hold for a single Riemannian Newton step (2.13) solved by the modified CG method when Hess $m_k(X^k)$ is not positive definite. One may solve the subproblem (2.1) more accurately by applying multiple Riemannian Newton steps or the Riemannian gradient type methods until (3.2) is satisfied. In our numerical experiments, we found

that the inexact condition (3.2) is often satisfied with a single Riemannian Newton step and local linear convergence rate is observed for the LSR1 scheme.

Following the proof in [20, 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 6. *Let assumptions* (B1)–(B4) *and the inexact condition* (3.1) *be satisfied. Then, all iterations are eventually very successful (i.e., r_k defined in* (2.14) *satisfying $r_k \geq \eta_2$).*

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

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

for some suitable $\delta_k \in [0,1]$ and $X^k_\delta := X^k + \delta_k(Z^k - X^k)$. Since the Stiefel manifold is compact, there exist some ξ^k such that $Z^k = \mathrm{Exp}_{X^k}(\xi^k)$, where Exp_{X^k} is the exponential map from $T_{X^k}\mathrm{St}(n,p)$ to $\mathrm{St}(n,p)$. Following the proof in [5, Appendix B], we have

$$\begin{aligned} \|Z^k - X^k - \xi^k\|_{\mathsf{F}} &\leq \kappa_1 \|\xi^k\|_{\mathsf{F}}^2, \\ \|Z^k - X^k\|_{\mathsf{F}} &\leq \kappa_2 \|\xi^k\|_{\mathsf{F}} \end{aligned} \tag{3.10}$$

with positive constants κ_1 and κ_2. It follows from (2.5) that

$$\mathrm{Hess}\, m_k(X^k)[\xi^k] = \mathrm{Hess}\, f(X^k)[\xi_k] + \tau_k \xi_k + \mathrm{Proj}_{X^k}\left(\mathcal{B}^k - \nabla^2 f(X^k)[\xi_k]\right).$$

Moreover, since the Hessian $\mathrm{Hess}\, f(X_*)$ is positive definite and (B4) is satisfied, it holds for sufficiently large k that

$$\|\mathrm{Hess}\, m_k(X^k)[\xi^k]\|_{\mathsf{F}} \geq \left(\lambda_{\min}(\mathrm{Hess}\, f(X^k)) + \tau_k\right)\|\xi^k\|_{\mathsf{F}} + o(\|\xi^k\|_{\mathsf{F}}),$$

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

$$\begin{aligned} &\|\mathrm{grad}\,(m_k \circ \mathrm{Exp}_{X^k})(\xi^k) - \mathrm{grad}\, f(X^k)\|_{\mathsf{F}} \\ &= \|\mathrm{Hess}\, m(X^k)[\xi^k]\|_{\mathsf{F}} + o(\|\xi^k\|_{\mathsf{F}}) \geq \frac{\kappa_2 + \tau_k}{2}\|\xi^k\|_{\mathsf{F}}, \end{aligned} \tag{3.11}$$

where $\kappa_2 := \lambda_{\min}(\mathrm{Hess}\, f(X_*))$. By [2, Lemma 7.4.9], there exists a positive constant $\tilde{c}$ such that $\|\mathrm{grad}\,(m_k \circ \mathrm{Exp}_{X^k})\|_{\mathsf{F}} \leq \tilde{c}\|\mathrm{grad}\, m_k(Z^k)\|_{\mathsf{F}} \leq \tilde{c}\|\mathrm{grad}\, f(X^k)\|_{\mathsf{F}}$, where we use the inexact condition (3.2). Consequently, we have from (3.11) that

$$\|\xi^k\|_{\mathsf{F}} \leq \frac{2(1+\tilde{c})}{\kappa_2 + \tau_k}\|\mathrm{grad}\, f(X^k)\|_{\mathsf{F}}. \tag{3.12}$$

It follows from (3.10) and (3.12) that

$$\begin{aligned} \frac{\|Z^k - X^k\|_{\mathsf{F}}^2}{\frac{a}{\tau_k+b}\|\mathrm{grad}\, f(X^k)\|_{\mathsf{F}}^2} \leq \frac{\kappa_2^2(\tau_k+b)\|\xi^k\|_{\mathsf{F}}^2}{a\|\mathrm{grad}\, f(X^k)\|_{\mathsf{F}}^2} &\leq \frac{4\kappa_2^2(1+\tilde{c})^2(\tau_k+b)}{a(\kappa_2+\tau_k)^2} \\ &\leq \frac{4\kappa_2^2 b(1+\tilde{c})^2}{a\kappa_2^2} + \frac{\kappa_2^2(1+\tilde{c})^2}{a\kappa_2}. \end{aligned} \tag{3.13}$$

The continuity of $\nabla^2 f$, (3.1), (3.9), (3.10), (3.12), and (3.13) imply that

$$1 - r_k \leq \frac{\tau_k + b}{2a}\left(\frac{\|(\nabla^2 f(X^k) - \mathcal{B}^k)[Z^k - X^k]\|_{\mathsf{F}}\|Z^k - X^k\|_{\mathsf{F}}}{\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}^2} + \frac{\|\nabla^2 f(X_\delta^k) - \nabla^2 f(X^k)\|\|Z^k - X^k\|_{\mathsf{F}}^2}{\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}^2}\right) \to 0.$$

Therefore the iterations are eventually very successful. ∎

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

THEOREM 7. *Suppose that assumptions* (B1)–(B4) *and conditions* (3.1) *and* (3.2) *hold. Then the sequence* $\{X^k\}$ *converges q-superlinearly to* X_*.

Proof. Since the iterations are eventually very successful, we have $X^{k+1} = Z^k$ and τ_k converges to zero. Let $\Delta^k = Z^k - X^k$. Recalling the definition m_k in (2.2), we have $\operatorname{grad} m_k(X^{k+1}) = \operatorname{Proj}_{X^{k+1}}\left(\nabla f(X^k) + \mathcal{B}^k[\Delta^k] + \tau_k\Delta^k\right)$. Thus, we have from (3.2) that

$$\left\|\operatorname{Proj}_{X^{k+1}}\left(\nabla f(X^k) + \mathcal{B}^k[\Delta^k] + \tau_k\Delta^k\right)\right\|_{\mathsf{F}} \leq \theta^k\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}. \tag{3.14}$$

Hence, we have

$$\begin{aligned}\|\operatorname{grad} f(X^{k+1})\|_{\mathsf{F}} &= \left\|\operatorname{Proj}_{X^{k+1}}\left(\nabla f(X^{k+1})\right)\right\|_{\mathsf{F}}\\ &= \left\|\operatorname{Proj}_{X^{k+1}}\left(\nabla f(X^k) + \nabla^2 f(X^k)[\Delta^k] + o(\|\Delta^k\|_{\mathsf{F}})\right)\right\|_{\mathsf{F}}\\ &= \left\|\operatorname{Proj}_{X^{k+1}}\left(\nabla f(X^k) + \mathcal{B}^k[\Delta^k] + \tau_k\Delta^k + o(\|\Delta^k\|_{\mathsf{F}})\right.\right.\\ &\quad \left.\left. + (\nabla^2 f(X^k) - \mathcal{B}^k)[\Delta^k] - \tau_k\Delta^k\right)\right\|_{\mathsf{F}}\\ &\leq \theta^k\|\operatorname{grad} f(X^k)\|_{\mathsf{F}} + o(\|\Delta^k\|_{\mathsf{F}}),\end{aligned} \tag{3.15}$$

where the last inequality is due to (3.14) and the fact that τ_k converges to zero. It follows from a similar argument to (3.12) that there exists some constant c_1 such that

$$\|\Delta^k\|_{\mathsf{F}} \leq c_1\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}$$

for sufficiently large k. Therefore, from (3.15) and the definition of θ^k, we have

$$\frac{\|\operatorname{grad} f(X^{k+1})\|_{\mathsf{F}}}{\|\operatorname{grad} f(X^k)\|_{\mathsf{F}}} \to 0. \tag{3.16}$$

Combining (3.16), assumption (B3), and [2, Lemma 7.4.8], it yields

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

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

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

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

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. Since a usual quadratic approximation to the purely quadratic function $f(X)$ in (4.1) introduces a linear term, the resulting subproblem is not a linear eigenvalue problem and has no closed-form solution. We next investigate a specific construction of the subproblem. Motivated by the quasi-Newton methods and eliminating the linear term in subproblem (2.1), we investigate the multisecant conditions in [16]

$$\hat{B}^k X^k = BX^k, \quad \hat{B}^k S^k = BS^k \tag{4.2}$$

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

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

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

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

where

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

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

$$C^k = A + \hat{B}^k. \tag{4.6}$$

Since the objective function is invariant under rotation, i.e., $f(XQ) = f(X)$ for orthogonal matrix $Q \in \mathbb{R}^{p\times p}$, it is desirable to construct a subproblem whose objective function inherits the same property. Noticing that problem (4.1) is actually an optimization problem on the Grassmann manifold [2, Chapter 3], we use the distance between the projectors associated with X^k and X,

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

which has been considered in [12, 45, 52]. Since X^k and X are orthonormal matrices, we have

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

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

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

where

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

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

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

where Λ is a diagonal matrix whose diagonal elements are the p smallest eigenvalues of $A + \hat{B}^k - \tau_k X^k (X^k)^\top$. From the computation of the Riemannian Hessian on the Grassmann manifold [2, Chapter 5], the term $\frac{\tau_k}{4} d_p(X, X^k)$ shifts the spectrum of the Riemannian Hessian of $\frac{1}{2}\mathrm{tr}(X^\top C^k X)$ by τ_k. Problem (4.8) with an approximate Hessian still works on the Grassmann manifold. 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 (2.14) between the actual reduction and predicted reduction based on this trial point. Then the iterate and regularization parameter are updated according to (2.14) and (2.16). 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 (2.3) and (2.4), 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

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

for any given matrices A and B. Hence, assumptions (A1) and (A2) hold with $L_f = \kappa_H = \|A\|_2 + \|B\|_2$. The Riemannian gradient of f in (4.1) is $\mathrm{grad}\, f(X) = \left(I_n - XX^\top\right)(CX)$. Similar to Theorem 4, we have the following theorem on the global convergence.

THEOREM 8. *Suppose that the inexact condition* (3.1) *holds. Then, for the Riemannian gradients of f in* (4.1)*, either*

$$\left(I_n - X^t (X^t)^\top\right)(CX^t) = 0 \text{ for some } t > 0 \text{ or } \lim_{k\to\infty} \| \left(I_n - X^k (X^k)^\top\right)(CX^k)\|_{\mathsf{F}} = 0.$$

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 [20, Lemma 9]. Specifically, the reduction of the subproblem requires that

$$\left\langle Z^k, C^k Z^k \right\rangle + \frac{\tau_k}{4} \|Z^k (Z^k)^\top - X^k (X^k)^\top\|_{\mathsf{F}}^2 - \left\langle X^k, C^k X^k \right\rangle \le 0.$$

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

$$\left\langle C^k, Z^k (Z^k)^\top - X^k (X^k)^\top \right\rangle + \frac{\tau_k}{4} \|Z^k (Z^k)^\top - X^k (X^k)^\top\|_{\mathsf{F}}^2 \le 0.$$

Then

$$\|Z^k (Z^k)^\top - X^k (X^k)^\top\|_{\mathsf{F}} \le \frac{4\kappa_H}{\tau_k}. \tag{4.9}$$

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

$$m_k(Z^k) - m_k(X^k) \ge -\frac{\nu}{\tau_k} \|\mathrm{grad}\, f(X^k)\|_{\mathsf{F}}^2. \tag{4.10}$$

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

$$\begin{aligned} & f(Z^k) - f(X^k) - (m_k(Z^k) - m_k(X^k)) \\ & \quad = \langle Z^k, CZ^k \rangle - \langle Z^k, C^k Z^k \rangle - \frac{\tau_k}{4} \|Z^k (Z^k)^\top - X^k (X^k)^\top\|_{\mathsf{F}}^2 \\ & \quad \le \langle C - C^k, Z^k (Z^k)^\top \rangle = \left\langle C - C^k, \left(Z^k (Z^k)^\top - X^k (X^k)^\top\right)^2 \right\rangle \\ & \quad \le (L_f + \kappa_H) \|Z^k (Z^k)^\top - X^k (X^k)^\top\|_{\mathsf{F}}^2 \\ & \quad \le \frac{16 \kappa_H^2 (L_f + \kappa_H)}{\tau_k^2}, \end{aligned} \tag{4.11}$$

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

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

for sufficiently large τ_k as in [20, 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 $\operatorname{grad} f(X^k) = \left(I_n - X^k (X^k)^\top\right)(CX^k)$ follows in a similar fashion as [20, Theorem 11]. ∎

The ACE method in [34] 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 [34], one may also establish the convergence to a global minimizer.

5. Electronic structure calculation.

5.1. Formulation.

Electronic structure calculations with hybrid functionals involve the Fock exchange operator. With some abuse of terminology, we refer to Kohn–Sham equations with local or semilocal exchange-correlation functionals as KSDFT, and Kohn–Sham equations with hybrid functionals as Hartree–Fock (HF). We now introduce the KSDFT 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 $X = [x_1, \ldots, x_p] \in \mathbb{C}^{n \times p}$ 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

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

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

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

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

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

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

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

Note that $H_{\text{ks}}(X)$ only depends on X through the charge density ρ, and hence can also be written as $H_{\text{ks}}(\rho)$.

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

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

Under the assumption that $\epsilon_{\text{xc}}(\rho(X))$ is twice differentiable with respect to $\rho(X)$, Lemma 2.1 in [49] gives an explicit form of the Hessian of $E_{\text{ks}}(X)$ as

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

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

After discretization, the Fock exchange operator $\mathcal{V}(\cdot) : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is usually a fourth-order tensor; see equations (3.3) and (3.4) in [32] for details. Furthermore, it is easy to see from [32] 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

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

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

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

Then the HF total energy minimization problem can be formulated as

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

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

LEMMA 9. *Given $U \in \mathbb{C}^{n\times p}$, the gradient and the Hessian along U of $E_{\text{f}}(X)$ are, respectively,*

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

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

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

$$\begin{aligned} 4E_{\rm f}(X+U) =& \left\langle \mathcal{V}((X+U)(X+U)^*)\,, (X+U)(X+U)^* \right\rangle \\ =& 4E_{\rm f}(X) + 2\left\langle \mathcal{V}(XX^*), D + UU^* \right\rangle + \left\langle \mathcal{V}(D+UU^*), D+UU^* \right\rangle \\ =& 4E_{\rm f}(X) + 2\left\langle \mathcal{V}(XX^*), D \right\rangle + 2\left\langle \mathcal{V}(XX^*), UU^* \right\rangle + \left\langle \mathcal{V}(D), D \right\rangle + \varrho(U), \end{aligned}$$

where $\varrho(U)$ denotes the third and fourth-order terms about U. Noting that $\mathcal{V}(XX^*)$ and $\mathcal{V}(D)$ are both Hermitian, we have from the above assertions that

$$E_{\rm f}(X+U) = E_{\rm f}(X) + \Re\left\langle \mathcal{V}(XX^*)X, U\right\rangle + \frac{1}{2}\Re\left\langle \mathcal{V}(XX^*)U + \mathcal{V}(D)X, U\right\rangle + \varrho(U). \tag{5.11}$$

Finally, it follows from expansion (1.2) in [49] that the second-order Taylor expression in X can be expressed as

$$E_{\rm f}(X+U) = E_{\rm f}(X) + \Re\langle \nabla E_{\rm f}(X), U\rangle + \frac{1}{2}\Re\left\langle \nabla^2 E_{\rm f}(X)[U], U\right\rangle + \varrho(U),$$

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

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

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

and have from (5.5) and (5.10) that

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

5.2. Self-consistent field iteration methods. 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 self-consistent field iteration (SCF) method [32]. At the kth iteration, we first fix $H_{\rm ks}(X)$ to be $H_{\rm ks}(X^k)$ and solve the following linear eigenvalue problem:

$$\widetilde{X} := \mathop{\arg\min}_{X\in\mathbb{C}^{n\times p}} \ \frac{1}{2}\langle X, H_{\rm ks}(X^k)X\rangle \quad \text{s.t.} \quad X^*X = I_p. \tag{5.14}$$

Note that in KSDFT, $H_{\rm ks}(X^k) \equiv H_{\rm ks}(\rho^k)$ depends on the charge density ρ^k. The output eigenvectors $\widetilde{X}$ lead to a new charge density $\widetilde{\rho}$, which is then mixed with charge densities from previous steps to generate the new charge density ρ^{k+1} and hence $H_{ks}(\rho^{k+1})$. Hence this type of SCF method is also called the charge mixing method.

For the HF model, the Hamiltonian not only depends on ρ but also XX^*. Hence we cannot directly apply the charge mixing method. Computing $\mathcal{V}\left(X^k(X^k)^*\right)U$ with some matrix U of proper dimension is still very expensive, and we investigate the limited-memory Nyström approximation $\hat{\mathcal{V}}\left(X^k(X^k)^*\right)$ to approximate $\mathcal{V}\left(X^k(X^k)^*\right)$ to reduce the computational cost, i.e.,

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

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

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

We should note that a similar idea called the adaptive compression method was proposed in [33], which compresses the operator $\mathcal{V}(X^k(X^k)^*)$ on the subspace $\text{span}\{X^k\}$. Then a new subproblem is constructed as

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

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 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 condtions not met* **do**
  Compute the limited-memory Nyström approximation $\hat{\mathcal{V}}\left(X^k(X^k)^*\right)$.
  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 [15] with the ACE formulation [33] when the subspace in (5.15) and inner solver for (5.16) are chosen as $\text{span}\{X^k\}$ and SCF, respectively.

Another method to solve the HF model (5.8) is the commutator direct inversion of the iterative subspace (C-DIIS) method. By storing the density matrix explicitly, it can often lead to an accelerated convergence rate. However, when the size of the density matrix becomes large, the storage cost of the density matrix becomes prohibitively expensive. Hu, Lin, and Yang [21] proposed the projected C-DIIS (PC-DIIS) method, which only requires storage of wave function type objects instead of the whole density matrix. The ACE technique [33] was also used in PC-DIIS. In this paper, we focus on the comparisons of our Algorithms 1 and 2.

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), and we next give the specific choices of $\mathcal{H}^{\text{c}}(X^k)$ and $\mathcal{H}^{\text{e}}(X^k)$, which are key to formulating 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 $\nabla^2 E_{\rm ks}$, we can choose

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

The exact Hessian of $E_{\rm hf}(X)$ in (5.8) can be separated naturally into two parts, i.e., $\nabla^2 E_{\rm ks}(X) + \nabla^2 E_{\rm f}(X)$. Usually the hybrid exchange operator $\mathcal{V}(XX^*)$ can take more than 95% of the overall time of the multiplication of $H_{\rm hf}(X)[U]$ in many real applications [34]. Recalling (5.5), (5.10), and (5.13), we know that the computational cost of $\nabla^2 E_{\rm f}(X)$ is much higher than that of $\nabla^2 E_{\rm ks}(X)$. Hence, we obtain the decomposition as

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

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

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

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

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 $H_{\rm ks}$ for (5.2) and $H_{\rm hf}$ 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 necessarily the eigenvectors corresponding to the p smallest eigenvalues. Since the columns of an optimal solution X are still the eigenvectors of $H(X)$, we can construct some subspace which contains these possible wanted eigenvectors. Specifically, at the current iterate, we first compute the first γp smallest eigenvalues and their corresponding eigenvectors of $H(X^k)$, denoted by Γ^k, and then construct the subspace as

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

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 on 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:

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

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

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:

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

where x_i is the ith 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 [29]. ASQN is the algorithm described in section 4. The difference between ACE and ASQN is that we take O^k as $\mathbf{orth}(\text{span}\{X^k\})$ but not $\mathbf{orth}(\text{span}\{X^{k-1}, X^k\})$. 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). Noting that BX^{k-1} and BX^k are available from the computation of the residual, we then adopt the orthogonalization technique in [35] 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 $AV, BV \in \mathbb{R}^{n\times p}$ as p MVs. The labels "#A" and "#B" are the total number of calls of A and B. The columns "err," "time," and "B-time" are the maximal relative error of all p eigenvectors defined in (6.1), the wall-clock time (in seconds) of each algorithm, and the wall-clock time of BV (in seconds), 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 can see that ASQN performs better than EIGS, LOBPCG, and ACE in terms of both accuracy and time. ACE spends a relatively long time to reach a solution with a similar accuracy. For the case with a fixed $n = 5000$ but different values of p, ASQN can still provide an accurate solution using less time than the three other methods. We note that the matrix B is always multiplied by an n-by-p matrix in ASQN and ACE. However, the multiplication between the matrix B and n-dimensional vector often occurs in EIGS. Therefore, under similar counts of #Bv,

TABLE 1
Numerical results on random matrices.

	#Av/#A/#Bv/#B	err	Time	B-time	#Av/#A/#Bv/#B	err	Time	B-time
$p = 10$								
n	5000				6000			
EIGS	459/450/459/450	8.0e-11	19.9	18.1	730/721/730/721	6.9e-11	48.9	45.9
LOBPCG	1717/387/1717/387	9.9e-11	46.7	23.1	2105/382/2105/382	9.8e-11	97.3	42.6
ASQN	2323/530/150/15	9.2e-11	6.0	1.3	2798/610/160/16	9.5e-11	8.5	2.0
ACE	4056/1145/460/46	9.7e-11	13.0	3.8	4721/1103/460/46	9.4e-11	17.0	5.8
n	8000				10000			
EIGS	538/529/538/529	8.7e-11	70.6	66.6	981/972/981/972	8.8e-11	153.8	144.8
LOBPCG	1996/314/1996/314	9.9e-11	134.0	57.2	2440/387/2440/387	9.7e-11	287.4	122.5
ASQN	2706/567/150/15	8.9e-11	11.2	2.8	2920/581/150/15	9.7e-11	17.8	5.4
ACE	4537/1162/450/45	9.8e-11	26.1	9.8	4554/951/400/40	9.6e-11	35.3	14.1
$n = 5000$								
p	10				20			
EIGS	459/450/459/450	8.0e-11	19.9	18.1	638/619/638/619	3.2e-11	44.4	42.5
LOBPCG	1717/387/1717/387	9.9e-11	46.7	23.1	2914/308/2914/308	9.8e-11	70.4	22.2
ASQN	2323/530/150/15	9.2e-11	6.0	1.3	3809/429/260/13	9.2e-11	5.9	1.2
ACE	4056/1145/460/46	9.7e-11	13.0	3.8	5902/775/680/34	9.5e-11	10.9	3.2
p	30				50			
EIGS	660/631/660/631	3.0e-11	47.4	45.2	879/830/879/830	1.6e-12	47.7	44.6
LOBPCG	4412/707/4412/707	9.7e-11	111.2	56.1	5766/542/5766/542	9.5e-11	97.0	40.0
ASQN	5315/636/420/14	9.8e-11	7.9	1.3	7879/711/650/13	9.8e-11	12.6	1.8
ACE	9701/1173/1530/51	9.4e-11	15.8	4.6	21832/2270/4500/90	9.7e-11	41.4	13.2

EIGS usually takes more calls of B, i.e., more #B. Similar conclusions can also be seen from Table 2. From the numbers #Av, #Bv, #A, and #B, we can see that the limited-memory Nyström method reduces the computational cost on the expensive 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 KSSOLV package [51]. One of the main differences is that the new version uses the more recently developed optimized norm-conserving Vanderbilt pseudopotentials [17], 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 comparisons, 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," and "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 the structured quasi-Newton method with the SCF in KSSOLV [51], the Riemannian L-BFGS method (RQN) in Manopt [6], the Riemannian gradient method with BB step size (GBB) [20], and the adaptive regularized Newton method (ARNT) [20]. 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 the same as that of ARNT [20].

TABLE 2
Numerical results on sparse matrices.

	#Av/#A/#Bv/#B	err	Time	B-time	#Av/#A/#Bv/#B	err	Time	B-time
s	9				10			
EIGS	1752/1743/1752/1743	6.0e-08	13.2	11.1	1390/1381/1390/1381	9.1e-11	25.8	24.2
LOBPCG	4042/1003/4042/1003	3.5e-05	28.8	9.4	3304/689/3304/689	9.7e-11	40.5	10.0
ASQN	7865/3661/280/28	8.6e-11	14.9	0.4	5540/1424/210/21	8.5e-11	8.6	0.4
ACE	16459/6903/2010/201	8.5e-08	32.3	3.1	9657/2816/880/88	9.0e-11	18.5	1.9
s	11				12			
EIGS	1779/1770/1779/1770	5.3e-08	37.8	35.3	1759/1750/1759/1750	8.4e-11	40.1	37.8
LOBPCG	5091/1003/5091/1003	1.8e-08	71.3	24.4	4493/1003/4493/1003	1.4e-09	80.2	30.8
ASQN	8619/2662/240/24	8.7e-11	15.7	0.7	7622/2772/260/26	9.9e-11	17.8	1.0
ACE	13223/4222/970/97	9.9e-11	26.4	2.8	17113/6217/2010/201	1.4e-08	42.2	8.7
$s = 12$								
p	10				20			
EIGS	1759/1750/1759/1750	8.4e-11	40.1	37.8	1730/1711/1730/1711	8.2e-11	54.8	52.2
LOBPCG	4493/1003/4493/1003	1.4e-09	80.2	30.8	7488/1003/7488/1003	3.4e-04	110.2	40.9
ASQN	7622/2772/260/26	9.9e-11	17.8	1.0	15337/5290/680/34	9.8e-11	40.2	1.5
ACE	17113/6217/2010/201	1.4e-08	42.2	8.7	26087/7149/4020/201	3.2e-05	56.1	8.4
p	30				40			
EIGS	1561/1532/1561/1532	6.9e-11	50.9	48.5	1553/1514/1553/1514	4.4e-11	51.3	49.3
LOBPCG	8855/753/8855/753	9.7e-11	91.4	26.3	10522/616/10522/616	9.7e-11	89.7	22.2
ASQN	13646/1666/600/20	9.2e-11	24.3	1.1	15392/1032/680/17	9.6e-11	23.0	1.0
ACE	27099/3904/3780/126	9.7e-11	48.0	7.0	24310/2074/2640/66	9.6e-11	36.6	3.9

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

For each algorithm, we first use GBB to generate a good starting point with stopping criterion $\|\text{grad}\, f(X^k)\|_{\mathsf{F}} \leq 10^{-1}$ 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.

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"

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

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 are 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

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

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 [48], where sometimes the converged solution may have a "hole" (i.e., unoccupied states) below the highest occupied energy level.

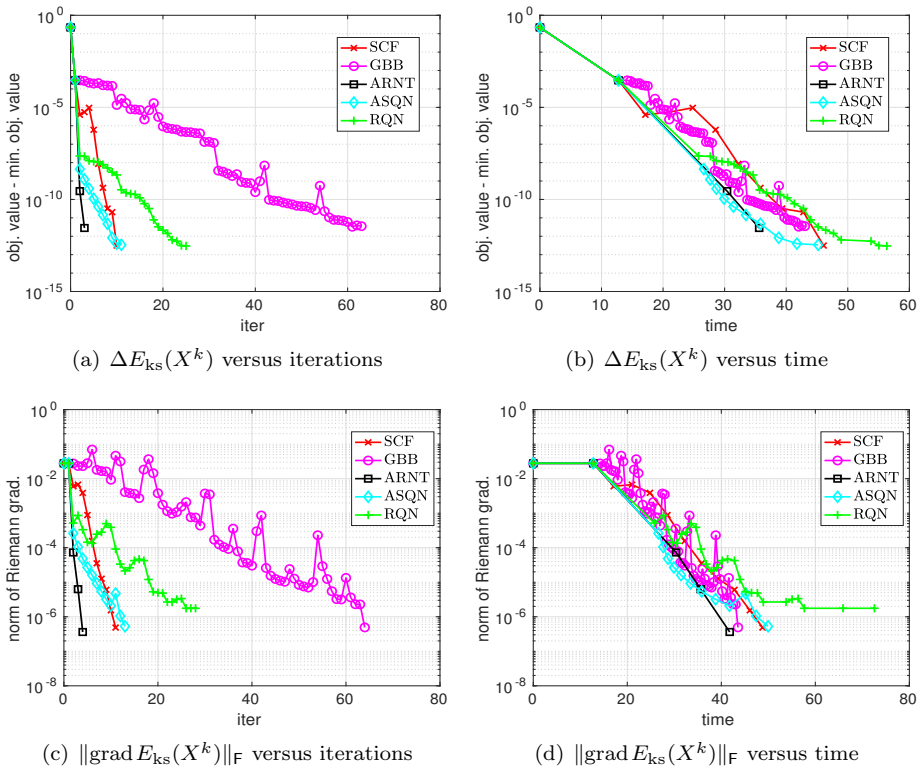
In the case "graphene16," we still obtain the same solution from GBB, ARNT, and ASQN. The number of wave functions p is 37. Let X be the computed solution and the corresponding eigenvalues of $H(X)$ be d. The smallest 37 eigenvalues and their corresponding eigenvectors of $H(X)$ are g and Y. We find 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. 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 $\nabla^2 E_{\rm ks}$ 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. To compare the computational cost of the cheap part $\mathcal{H}^{\rm c}$ and the remaining parts $\mathcal{H}^{\rm e}$ in $\nabla^2 E_{\rm ks}$, we repeat the calculations of $\mathcal{H}^{\rm e}(X)[U]$ and $\mathcal{H}^{\rm c}(X)[U]$ with fixed X and U 50 times; the ratios between the total time of $\mathcal{H}^{\rm e}(X)[U]$ and $\mathcal{H}^{\rm c}(X)[U]$ on "alanine," "c12h26," "ctube661," and "glutamine" are 22.2, 18.5, 10.6, and 22.0, respectively. 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

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

where $E_{\rm min}$ is the minimum of the total energy attained by all methods.

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 GBB (GBBN), respectively, the Riemannian L-BFGS (RQN) method in Manopt [6], and two variants of Algo-



(a) $\Delta E_{\rm ks}(X^k)$ versus iterations

(b) $\Delta E_{\rm ks}(X^k)$ versus time

(c) $\|\mathrm{grad}\, E_{\rm ks}(X^k)\|_{\mathsf{F}}$ versus iterations

(d) $\|\mathrm{grad}\, E_{\rm ks}(X^k)\|_{\mathsf{F}}$ versus time

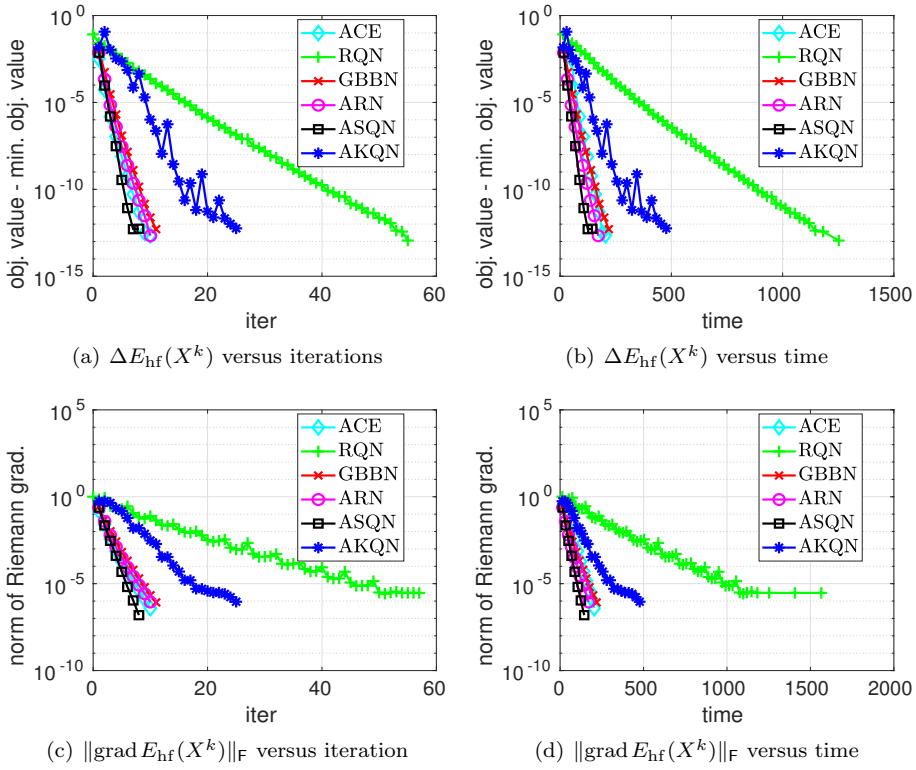
FIG. 1. *Comparisons of different algorithms on "glutamine" of KS total energy minimization. The first two points are the input and output of the initial solver GBB, respectively.*

rithm 1 with approximation (5.18) (ASQN) and approximation (5.19) (AKQN). Since the computation of the exact Hessian $\nabla^2 E_{\rm hf}$ 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 $\mathcal{H}^{\rm e}(X^k)$ 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 $0.1\min\{\|\mathrm{grad}\, f(X^k)\|_{\mathsf{F}}, 1\}$. We also use the adaptive strategy for choosing the maximal number of inner iterations of ARNT in [20] for GBBN, ASQN, AKQN, and ARN. The settings of other parameters of ASQN, AKQN, and ARN are the same to those in ARNT [20]. For all algorithms, we generate a good initial guess by using GBB to solve the corresponding KS total energy minimization problem (i.e., remove $E_{\rm f}$ part from $E_{\rm hf}$ 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.

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

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 take fewer iterations if more parts in the Hessian are preserved. Since the computational cost of the Fock exchange energy dominates that of the KS part, algorithms using fewer outer iterations consume 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." In fact, for a fixed X and U on "alanine," "c12h26," "ctube661," and "glutamine," the ratios between the total



(a) $\Delta E_{\rm hf}(X^k)$ versus iterations (b) $\Delta E_{\rm hf}(X^k)$ versus time

(c) $\|\mathrm{grad}\, E_{\rm hf}(X^k)\|_{\rm F}$ versus iteration (d) $\|\mathrm{grad}\, E_{\rm hf}(X^k)\|_{\rm F}$ versus time

FIG. 2. *Comparisons of different algorithms on "glutamine" of HF total energy minimization.*

time of $\mathcal{H}^{\rm e}(X)[U]$ and $\mathcal{H}^{\rm c}(X)[U]$ are 32.2, 70.4, 86.4, and 53.8, respectively. Finally, we show the convergence behaviors of these six methods on the system "glutamine" in Figure 2, where $\Delta E_{\rm hf}(X^k)$ is defined similarly as the KS case. In summary, algorithms utilizing the quasi-Newton technique combined with the Nyström approximation are 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 respects, for example, finding a better initial quasi-Newton matrix than the 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, New York, 2013, pp. 361–368.

[4] A. D. Becke, *Density-functional thermochemistry.* III. *The role of exact exchange*, J. Chem. 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., 39 (2019), pp. 1–33.

[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.

[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] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part* I*: Motivation, convergence and numerical results*, Math. Program., 127 (2011), pp. 245–295.

[11] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part* II*: Worst-case function- and derivative-evaluation complexity*, Math. Program., 130 (2011), pp. 295–319.

[12] 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.

[13] D. Gabay, *Minimizing a differentiable function over a differential manifold*, J. Optim. Theory Appl., 37 (1982), pp. 177–219.

[14] K. A. Gallivan and P.-A. Absil, *Note on the Convex Hull of the Stiefel Manifold*, Florida State University, 2010.

[15] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, et al., *Quantum espresso: A modular and open-source software project for quantum simulations of materials*, J. Phys. Condensed Matter, 21 (2009), 395502.

[16] S. Gratton and P. L. Toint, *Multi-Secant Equations, Approximate Invariant Subspaces and Multigrid Optimization*, Technical report, Department of Mathematics, FUNDP, Namur (B), 2007.

[17] D. Hamann, *Optimized norm-conserving Vanderbilt pseudopotentials*, Phys. Rev. B, 88 (2013), 085117.

[18] J. Heyd, G. E. Scuseria, and M. Ernzerhof, *Hybrid functionals based on a screened Coulomb potential*, J. Chem. Phys., 118 (2003), pp. 8207–8215.

[19] P. Hohenberg and W. Kohn, *Inhomogeneous electron gas*, Phys. Rev., 136 (1964), pp. B864–B871.

[20] 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.

[21] 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.

[22] W. Huang, *Optimization Algorithms on Riemannian Manifolds with Applications*, Ph.D. thesis, Florida State University, 2013.

[23] W. Huang, P. Absil, K. Gallivan, and P. Hand, *ROPTLIB: An object-oriented C++ library for optimization on Riemannian manifolds*, ACM Trans. Math. Software, 44 (2018), pp. 1–21.

[24] 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.

[25] W. Huang, P.-A. Absil, and K. A. Gallivan, *A Riemannian symmetric rank-one trust-region method*, Math. Program., 150 (2015), pp. 179–216.

[26] 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, New York, 2016, pp. 627–634.

[27] 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.

[28] R. E. Kass, *Nonlinear regression analysis and its applications*, J. Amer. Statist. Assoc., 85 (1990), pp. 594–596.

[29] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[30] W. Kohn and L. J. Sham, *Self-consistent equations including exchange and correlation effects*, Phys. Rev., 140 (1965), pp. A1133–A1138.

[31] K. Kreutz-Delgado, *The Complex Gradient Operator and the CR-Calculus*, arXiv:0906.4835, 2009.

[32] C. Le Bris, *Computational chemistry from the perspective of numerical analysis*, Acta Numer., 14 (2005), pp. 363–444.

[33] L. Lin, *Adaptively compressed exchange operator*, J. Chem. Theory Comput., 12 (2016), pp. 2242–2249.

[34] L. Lin and M. Lindsey, *Convergence of adaptive compression methods for Hartree-Fock-like equations*, Comm. Pure Appl. Math., 72 (2019), pp. 451–499.

[35] 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.

[36] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, Cambridge, 2004.

[37] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.

[38] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.

[39] C. Qi, *Numerical Optimization Methods on Riemannian Manifolds*, Ph.D. thesis, Florida State University, 2011.

[40] W. Ring and B. Wirth, *Optimization methods on Riemannian manifolds and their application to shape space*, SIAM J. Optim., 22 (2012), pp. 596–627.

[41] M. Seibert, M. Kleinsteuber, and K. Hüper, *Properties of the BFGS method on Riemannian manifolds*, in Mathematical System Theory, 2013, pp. 395–412.

[42] S. T. Smith, *Optimization techniques on Riemannian manifolds*, Fields Inst. Commun., 3 (1994).

[43] W. Sun and Y. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Vol. 1, Springer, New York, 2006.

[44] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover, New York, 2012.

[45] 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), 074103.

[46] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher, *Fixed-rank approximation of a positive-semidefinite matrix from streaming data*, in Proceedings of Advances in Neural Information Processing Systems, 2017, pp. 1225–1234.

[47] C. Udriste, *Convex Functions and Optimization Methods on Riemannian manifolds*, Math. Appl. 297, Springer, New York, 1994.

[48] R. van Leeuwen, *Density functional approach to the many-body problem: Key concepts and exact functionals*, Adv. Quantum Chem., 43 (2003), pp. 25–94.

[49] 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.

[50] Z. Wen and W. Yin, *A feasible method for optimization with orthogonality constraints*, Math. Program., 142 (2013), pp. 397–434.

[51] C. Yang, J. C. Meza, B. Lee, and L.-W. Wang, KSSOLV—*A MATLAB toolbox for solving the Kohn-Sham equations*, ACM Trans. Math. Software, 36 (2009), pp. 1–35.

[52] 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.

[53] 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.