

PARALLELIZABLE ALGORITHMS FOR OPTIMIZATION PROBLEMS WITH ORTHOGONALITY CONSTRAINTS

BIN GAO*, XIN LIU[†], AND YA-XIANG YUAN[‡]

Abstract. To construct a parallel approach for solving optimization problems with orthogonality constraints is usually regarded as an extremely difficult mission, due to the low scalability of the orthonormalization procedure. However, such demand is particularly huge in some application areas such as materials computation. In this paper, we propose a proximal linearized augmented Lagrangian algorithm (PLAM) for solving optimization problems with orthogonality constraints. Unlike the classical augmented Lagrangian methods, in our algorithm, the prime variables are updated by minimizing a proximal linearized approximation of the augmented Lagrangian function, meanwhile the dual variables are updated by a closed-form expression which holds at any first-order stationary point. The orthonormalization procedure is only invoked once at the last step of the above mentioned algorithm if high-precision feasibility is needed. Consequently, the main parts of the proposed algorithm can be parallelized naturally. We establish global subsequence convergence, worst-case complexity and local convergence rate for PLAM under some mild assumptions. To reduce the sensitivity of the penalty parameter, we put forward a modification of PLAM, which is called parallelizable column-wise block minimization of PLAM (PCAL). Numerical experiments in serial illustrate that the novel updating rule for the Lagrangian multipliers significantly accelerates the convergence of PLAM and makes it comparable with the existent feasible solvers for optimization problems with orthogonality constraints, and the performance of PCAL does not highly rely on the choice of the penalty parameter. Numerical experiments under parallel environment demonstrate that PCAL attains good performance and high scalability in solving discretized Kohn-Sham total energy minimization problems.

Key words. orthogonality constraint, Stiefel manifold, augmented Lagrangian method, parallel computing

AMS subject classifications. 15A18, 65F15, 65K05, 90C06

1. Introduction. In this paper, we consider the following matrix variable optimization problem with orthogonality constraints.

$$(1.1) \quad \begin{aligned} \min_{X \in \mathbb{R}^{n \times p}} \quad & f(X) \\ \text{s. t.} \quad & X^\top X = I_p, \end{aligned}$$

where I_p is the p -by- p identity matrix with $2p \leq n$, and $f : \mathbb{R}^{n \times p} \mapsto \mathbb{R}$ is a continuously differentiable function. The feasible set of the orthogonality constraints is also known as Stiefel manifold, $\mathcal{S}_{n,p} = \{Y \in \mathbb{R}^{n \times p} \mid Y^\top Y = I_p\}$.

Throughout this paper, we assume

ASSUMPTION 1.1. (Blanket Assumption) f is continuously differentiable.

The twice differentiability of f will be particularly mentioned once it is required in some theoretical analyses.

1.1. Literature Survey. Kohn-Sham density functional theory (KSDFT) is known to be an important topic in materials science [16]. The last step of KSDFT is to minimize a discretized Kohn-Sham total energy function

$$(1.2) \quad E(X) := \frac{1}{4} \text{tr}(X^\top L X) + \frac{1}{2} \text{tr}(X^\top V_{ion} X) + \frac{1}{4} \rho^\top L^\dagger \rho + \frac{1}{2} \rho^\top \epsilon_{xc}(\rho),$$

subject to orthogonality constraints. Here $\rho(X) := \text{diag}(X X^\top)$ denotes the charge density, and $L \in \mathbb{R}^{n \times n}$ is a finite-dimensional representation of the Laplace operator in the planewave basis. The discretized local ionic potential

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

[†]State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, and University of Chinese Academy of Sciences, China (liuxin@lsec.cc.ac.cn). Research supported in part by NSFC grants 11622112, 11471325, 91530204 and 11688101, the National Center for Mathematics and Interdisciplinary Sciences, CAS, and Key Research Program of Frontier Sciences QYZDJ-SSW-SYS010, CAS.

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

can be represented by a diagonal matrix V_{ion} . And the matrix L^\dagger which is the discrete form of the Hartree potential corresponds to the pseudo-inverse of L . The exchange correlation function ϵ_{xc} is used to model the non-classical and quantum interaction between electrons. The discretized energy minimization is exactly a special case of (1.1). The variable scale of such problems is often very large, and hence the demand on efficient solvers for optimization with orthogonality constraints is high.

In recent decades, lots of researchers have proposed quite a few efficient optimization approaches for solving discretized Kohn-Sham total energy minimization [36, 37, 35, 34, 31, 38, 30, 8, 19]. For the general purpose of solving optimization problems with orthogonality constraints, there are abundant algorithms. Retraction based approaches [10, 25, 1, 33, 15], splitting algorithm [17], multipliers correction framework [11], just to mention a few. Interested readers are referred to the references in [11]. There are a few successful solvers. The most famous one is the toolbox for optimization on manifolds, which is called Manopt¹, in which lots of retraction based algorithms for Problem (1.1), such as MOptQR, a QR projection algorithm, are included. Another quasi-geodesic based approach called OptM² is widely used in the area of discretized Kohn-Sham energy minimization.

However, the lack of concurrency becomes a major bottleneck of solving optimization problems with orthogonality constraints, particularly, when the number of columns of the variable matrix is large. However, parallel computation is not popular in optimization area until very recently. Refer to [29, 6, 28, 20, 27], there is urgent demand of parallelization in optimization area. Although high scalability algorithms are desired by KSDFT area for decades, there is no successful attempt in this regard so far [8].

We find that parallelization is particular difficult for optimization problems with orthogonality constraints. The main reason is that the scalability of orthonormalization is low no matter which particular way how you do it.

1.2. Contribution. In this paper, we propose an infeasible algorithm for optimization problems with orthogonality constraints. It is based on augmented Lagrangian method but employs totally different updating scheme for both prime and dual variables. The main motivation of the so-called proximal linearized augmented Lagrangian method (PLAM) is an observation that the dual variables enjoy a closed-form formula at each first-order stationary point. Therefore, we consider to use the symmetrization of this formula as the updating rule for the dual variables, to take the place of dual ascent step in the classical augmented Lagrangian method. For the prime variables, instead of solving the augmented Lagrangian subproblem to some preset precision we minimize a proximal linearized approximation of the augmented Lagrangian function, which is equivalent to take one step gradient descent.

The orthonormalization procedures are waived in all iterations except the last one to guarantee high-precision feasibility. The cost of waiving orthonormalization is to do more BLAS3 calculations (matrix-matrix multiplication) which are known to have high scalability.

We show the global convergence, worst-case complexity and local Q-linear convergence rate for PLAM under some mild assumptions. The global convergence of PLAM requires sufficiently large penalty parameter and correspondingly small stepsize. Numerical tests also verify the sensitivity of the penalty parameter. Consequently, we put forward a novel modification strategy that is to add redundant unit norm constraints to the proximal linearized augmented Lagrangian subproblem for updating the prime variables. By using this strategy, we can restrict the iterates in a compact set such that the penalty parameter is no longer required to be large. On the other hand, such modification does not destroy the structure that the subproblem has a closed-form solution which can be calculated in parallel. We call the consequent algorithm PCAL, namely, parallelizable column-wise block minimization for PLAM. The boundedness of PCAL iterates can be guaranteed automatically, and hence the penalty parameter is no longer required to be sufficiently large.

¹Available from <http://www.manopt.org>

²Available from <http://optman.blogs.rice.edu>

The numerical experiments under serial computing demonstrate the way how to choose default settings for our algorithms, and show that the infeasible algorithms are at least as efficient as the existent feasible algorithms in solving a bunch of test problems. The numerical experiments under parallel computing illustrate the computational complexity of PCAL and expose its high scalability.

1.3. Organization and Notations. The motivation of new approaches will be introduced in the next section. In Section 3, we will present the algorithm frameworks. We investigated the theoretical behaviors of the new proposed algorithms in Section 4. Numerical experiments will be demonstrated in Section 5. In the last section, we will draw a brief conclusion and discuss possible future works.

Notations. $\mathbb{S}^p := \{X \in \mathbb{R}^{p \times p} \mid X^\top = X\}$ refers to the p -by- p real symmetric matrices set. $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ stand for the largest and smallest eigenvalues of given symmetric real matrix A , respectively. $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ denote the largest and smallest singular values of given real matrix A , respectively. $X^\dagger := (X^\top X)^{-1} X^\top$ refers to the pseudo inverse of X . $\text{Diag}(v) \in \mathbb{S}^n$ denotes a diagonal matrix with all entries of $v \in \mathbb{R}^n$ in its diagonal, and $\text{diag}(A) \in \mathbb{R}^n$ extracts the diagonal entries of matrix $A \in \mathbb{R}^{n \times n}$. For convenience, $\Phi(M) := \text{Diag}(\text{diag}(M))$ represents the diagonal matrix with the diagonal entries of square matrix M in its diagonal. $\Psi(A) := \frac{1}{2}(A + A^\top)$ stands for the average of a square matrix and its transpose.

2. Motivation. As mentioned in the previous section, almost all the existent practically useful methods require feasible iterates all the time. To realize feasibility, either explicit or implicit orthonormalization requires to be invoked. Such kind of calculation lacks of scalability and hence becomes the bottleneck computation in the corresponding algorithms. For example, we consider the discretized Kohn-Sham total energy minimization (1.2). In each iteration, the function value and first-order derivative evaluation cost $O(n \log n + np)$ or $O(np)$ flops per iteration, depending on whether plane wave or finite difference, respectively, is used in the discretization scheme. For the main iteration of any algorithm for solving (1.2) developed in recent decade, the computational cost per iteration is $O(np^2)$ for BLAS3 calculation, plus $O(p^3)$ for orthonormalization which can hardly be parallelized.

To break through this bottleneck, we suggest to use infeasible methods to take the place of feasible methods.

There is no existent infeasible approach for general purpose reported to be efficient for optimization problems with orthogonality constraints. Existent infeasible approaches designed for (1.1) either work specially for Rayleigh-Ritz trace minimization [23, 32], or adopt ADMM framework after introducing auxiliary variables to split the objective and orthogonality constraints [17]. The previous ones can hardly be extended to general objective, while the latter ones does not have good performance in general.

In the following subsections, we introduce how we come up with a new idea on constructing an efficient infeasible algorithm for problem (1.1).

2.1. The Optimality Condition. We start from the optimality condition of the optimization problem with orthogonality constraints (1.1). The first-order optimality condition of problem (1.1) can be written as

$$(2.1) \quad \begin{cases} \nabla f(X) = X\Lambda; \\ X^\top X = I_p, \end{cases}$$

where $\Lambda \in \mathbb{S}^p$ consists of the Lagrangian multipliers of the orthogonality constraints. Condition (2.1) has the following equivalent form where Λ is eliminated.

$$(2.2) \quad \begin{cases} \nabla f(X) - X\nabla f(X)^\top X = 0; \\ X^\top X = I_p. \end{cases}$$

DEFINITION 2.1. We call X a first-order stationary point, if condition (2.2) holds. We call X a second-order stationary point, if it is a first-order stationary point and satisfies

$$(2.3) \quad \text{tr}(Y^\top \nabla^2 f(X)[Y] - \Lambda Y^\top Y) \geq 0, \quad \forall Y \in \mathcal{T}(X),$$

where $\mathcal{T}(X) := \{Y \mid Y^\top X + X^\top Y = 0\}$ is the tangent space of the orthogonality constraints at X .

The following proposition can be easily verified and hence its proof is omitted here.

PROPOSITION 2.2. If X is a local minimizer of (1.1), it has to be a second-order stationary point. If X is a strict local minimizer³, if and only if X is a first-order stationary point and satisfies

$$(2.4) \quad \text{tr}(Y^\top \nabla^2 f(X)[Y] - \Lambda Y^\top Y) > 0, \quad \forall 0 \neq Y \in \mathcal{T}(X).$$

2.2. Augmented Lagrangian Method. A straightforward idea to solve (1.1) without requiring feasibility in each iteration is to employ the Augmented Lagrangian Method (ALM) [26, 3], which is described in Algorithm 1.

Algorithm 1: Augmented Lagrangian Method (ALM)

- 1 **Input:** choose initial guess Λ^0 for the dual variables, and set $k := 0$;
 - 2 **while** certain stopping criterion is not reached **do**
 - 3 Minimize the augmented Lagrangian function with respect to the prime variables X :

$$X^{k+1} := \min_{X \in \mathbb{R}^{n \times p}} \mathcal{L}_\beta(X, \Lambda^k),$$

where the augmented Lagrangian function for problem (1.1) is defined as

$$(2.5) \quad \begin{aligned} \mathcal{L}_\beta(X, \Lambda) &= f(X) - \frac{1}{2} \langle \Lambda, X^\top X - I_p \rangle + \frac{\beta}{4} \|X^\top X - I_p\|_F^2 \\ &= f(X) + \frac{\beta}{4} \left\| X^\top X - \left(I_p + \frac{1}{\beta} \Lambda \right) \right\|_F^2 - \frac{1}{4\beta} \|\Lambda\|_F^2. \end{aligned}$$
 - 4 Update the Lagrangian multipliers

$$(2.6) \quad \Lambda^{k+1} := \Lambda^k - \beta(X^{k+1\top} X^{k+1} - I_p).$$
 - 5 Update the penalty parameter β if necessary. Set $k := k + 1$.
 - 6 **Output:** X^k .
-

It is well-known that the augmented Lagrangian function (2.5) is an exact penalty if the parameter β is sufficiently large. Algorithm 1 works very well for problem with linear constraints. For optimization problems with nonlinear constraints, it is not clear how to choose the parameter β in practice, which is very sensitive to the numerical performance.

The purpose of this work is to find an infeasible algorithm for solving (1.1) at similar cost of the existent feasible methods. Otherwise, we can hardly gain much from the parallelization. To this end, we carefully test Algorithm 1 and try our best to tune the parameter β . Unfortunately, for solving optimization problems with orthogonality constraints (1.1), the efficiency of classical ALM is far from being satisfactory.

³ X is called a strict local minimizer, if there exists $\delta > 0$ such that $f(X) < f(Y)$ holds for any $Y \in U_\delta(X) := \{X \mid \|X - Y\| \in (0, \delta)\}$.

Therefore, we need to employ a new idea to remould the classical ALM. According to the conditions (2.1) and (2.2), it is not difficult to verify that the Lagrangian multipliers Λ have the following closed-form expression at any first-order stationary point,

$$(2.7) \quad \Lambda = \nabla f(X)^\top X.$$

A straightforward idea is to use the following symmetrized form of (2.7)

$$(2.8) \quad \Lambda = \Psi(\nabla f(X)^\top X)$$

as a new multipliers updating rule. The symmetrization is necessary because the symmetry of the expression $\nabla f(X)^\top X$ can not be guaranteed in each iteration.

As we will demonstrate in the following lemma and the theoretical analyses in Section 4, an explicit lower bound of the penalty parameter β can be estimated if updating rule (2.8) is applied. Hence, the update of the penalty parameter β can be waived. Moreover, the numerical experiments verify the validation of this new updating rule.

LEMMA 2.3. *Let X^* be a second-order stationary point of*

$$(2.9) \quad \min_{X \in \mathbb{R}^{n \times p}} \mathcal{L}_\beta(X, \Lambda^*)$$

with $\Lambda^ = \Psi(\nabla f(X^*)^\top X^*)$. Suppose $\beta > \lambda_{\max}(\nabla^2 f(X^*))$. Then X^* is a second-order stationary point of problem (1.1). Namely, optimality conditions (2.1) and (2.2) hold at X^* .*

Proof. For convenience, we abuse the notation slightly by deleting the superscript $*$ from X^* . First, we have

$$(2.10) \quad \nabla_X \mathcal{L}_\beta(X, \Lambda) = \nabla f(X) + \beta X \left(X^\top X - \left(I_p + \frac{1}{\beta} \Lambda \right) \right);$$

$$(2.11) \quad \nabla_{XX}^2 \mathcal{L}_\beta(X, \Lambda)[S] = \nabla^2 f(X)[S] + \beta S \left(X^\top X - \left(I_p + \frac{1}{\beta} \Lambda \right) \right) + \beta X(S^\top X + X^\top S).$$

Since X is the second-order stationary point of (2.9) with $\Lambda = \nabla f(X)^\top X$, we have

$$(2.12) \quad \nabla \mathcal{L}_\beta(X, \Lambda) = 0;$$

$$(2.13) \quad \langle S, \nabla_{XX}^2 \mathcal{L}_\beta(X, \Lambda)[S] \rangle \geq 0, \quad \forall S \neq 0.$$

Substituting (2.10) into (2.12), we obtain

$$(2.14) \quad \nabla f(X) - X\Lambda - \beta X(I_p - X^\top X) = 0.$$

Left multiplying X^\top into both sides of (2.14), we have

$$(2.15) \quad X^\top \nabla f(X) = X^\top X\Lambda + \beta X^\top X(I_p - X^\top X).$$

Suppose $X = U\Sigma V^\top$ is the singular value decomposition of X in economy-size, which implies $X^\top X = V\Sigma^2 V^\top$. Then, we further have

$$X^\top \nabla f(X) - \beta V\Sigma^2 V^\top = V\Sigma^2 V^\top \Lambda - \beta V\Sigma^4 V^\top.$$

Left multiplying V^\top and right multiplying V to both sides of the above equality, we arrive at

$$V^\top X^\top \nabla f(X) V - \beta \Sigma^2 = \Sigma^2 (V^\top \Lambda V - \beta \Sigma^2).$$

Taking the Φ operator and using the fact that

$$(2.16) \quad \text{diag}(V^\top X^\top \nabla f(X) V) = \text{diag}(V^\top \nabla f(X)^\top X V) = \text{diag}(V^\top \Lambda V),$$

we have

$$(2.17) \quad (I_p - \Sigma^2)(\Phi(V^\top \Lambda V) - \beta \Sigma^2) = 0,$$

which implies that

$$(2.18) \quad D(\Phi(V^\top \Lambda V) - \beta \Sigma^2) = 0,$$

where p -by- p diagonal matrix D satisfies

$$D_{ii} = \begin{cases} 0, & \text{if } (I_p - \Sigma^2)_{ii} = 0; \\ 1, & \text{otherwise,} \end{cases} \quad \forall i = 1, \dots, p.$$

On the other hand, since $n \geq 2p$, there exists $\tilde{U} \in \mathcal{S}_{n,p}$ satisfying $\tilde{U}^\top U = 0$. let $S = \tilde{U} D V^\top$. If $S \neq 0$, we substitute it into (2.11) and obtain

$$\begin{aligned} \langle S, \nabla_{XX}^2 \mathcal{L}_\beta(X, \Lambda)[S] \rangle &= \text{tr}(S^\top \nabla^2 f(X)[S]) - \beta \text{tr}(S^\top S) - \text{tr}(S^\top S(\Lambda - \beta X^\top X)) \\ &= \text{tr}(S^\top (\nabla^2 f(X) - \beta I)[S]) - \text{tr}(V^\top S^\top S V V^\top (\Lambda - \beta V \Sigma^2 V^\top) V) \\ &= \text{tr}(S^\top (\nabla^2 f(X) - \beta I)[S]) - \text{tr}(D^2 (V^\top \Lambda V - \beta \Sigma^2)) \\ &= \text{tr}(S^\top (\nabla^2 f(X) - \beta I)[S]) - \text{tr}(D^2 (\Phi(V^\top \Lambda V) - \beta \Sigma^2)). \end{aligned}$$

Here I stands for the identity mapping from $\mathbb{R}^{n \times p}$ to $\mathbb{R}^{n \times p}$. Combining with the second-order optimality condition (2.13), relationship (2.18) and the assumption on β , we have

$$(2.19) \quad 0 \leq \langle S, \nabla_{XX}^2 \mathcal{L}_\beta(X, \Lambda)[S] \rangle = \text{tr}(S^\top (\nabla^2 f(X) - \beta I)[S]) < 0,$$

which leads to contradiction. Hence, $S = 0$, which immediately implies that $\Sigma = I_p$. Therefore, we have $X \in \mathcal{S}_{n,p}$. Together with (2.12) and (2.13), we can easily show that the optimality condition (2.1) and (2.2) hold. This completes the proof. \square

The Lemma 2.3 guarantees that augmented Lagrangian function is still an exact penalty function with the Lagrangian multipliers updated by explicit formula (2.8). However, to achieve the convergence results for first-order methods, we need a first-order version of Lemma 2.3. Moreover, to obtain the global convergence rate, the feasibility should be controlled by the first-order optimality violation.

LEMMA 2.4. *For any X^* satisfying $\sigma_{\min}(X^*) > 0$ and suppose $\beta > (\|\nabla f(X^*)\|_2 \cdot \|X^*\|_2 + \delta) / \sigma_{\min}^2(X^*)$ with $\delta > 0$. Then it holds*

$$\|X^{*\top} X^* - I_p\|_F \leq \frac{\|X^*\|_2}{\delta} \cdot \|\nabla_X \mathcal{L}_\beta(X^*, \Lambda^*)\|_F,$$

with $\Lambda^* = \Psi(\nabla f(X^*)^\top X^*)$. In particular, if it happens that X^* is a first-order stationary point of

$$\min_{X \in \mathbb{R}^{n \times p}} \mathcal{L}_\beta(X, \Lambda^*)$$

with $\Lambda^* = \Psi(\nabla f(X^*)^\top X^*)$, then X^* is a first-order stationary point of problem (1.1).

Proof. For brevity, we denote $G = \nabla_X \mathcal{L}_\beta(X, \Lambda)$. Left multiplying X^\top into both sides of (2.10) and using the singular value decomposition $X = U\Sigma V^\top$, we have

$$X^\top G = X^\top \nabla f(X) - \beta V \Sigma^2 V^\top - V \Sigma^2 V^\top \Lambda + \beta V \Sigma^4 V^\top.$$

Left multiplying V^\top and right multiplying V to both sides of the above equality, we obtain

$$V^\top X^\top G V = V^\top X^\top \nabla f(X) V - \beta \Sigma^2 - \Sigma^2 (V^\top \Lambda - \beta \Sigma^2).$$

Taking the Φ operator and using the fact (2.16), we arrive at

$$(2.20) \quad \Phi(V^\top X^\top G V) = (I_p - \Sigma^2)(\Phi(V^\top \Lambda V) - \beta \Sigma^2).$$

Suppose $\beta > (\|\nabla f(X^*)\|_F \cdot \|X^*\|_2 + \delta) / \sigma_{\min}^2(X^*)$ holds, we have

$$\beta \sigma_{\min}^2(X^*) \geq \|\nabla f(X^*)\|_2 \cdot \|X^*\|_2 + \delta,$$

which implies

$$\sigma_{\min}(\beta \Sigma^2) \geq \|V^\top \Lambda V\|_2 + \delta \geq \|\Phi(V^\top \Lambda V)\|_2 + \delta.$$

Hence, it holds that

$$(2.21) \quad \sigma_{\min}(\beta \Sigma^2 - \Phi(V^\top \Lambda V)) \geq \delta.$$

Submitting (2.21) into (2.20), we arrive at

$$\begin{aligned} \|X\|_2 \|G\|_F &\geq \|\Phi(V^\top X^\top G V)\|_F = \|(I_p - \Sigma^2)(\Phi(V^\top \Lambda V) - \beta \Sigma^2)\|_F \\ &\geq \|I_p - \Sigma^2\|_F \cdot \sigma_{\min}(\beta \Sigma^2 - \Phi(V^\top \Lambda V)) \geq \|I_p - X^\top X\|_F \cdot \delta \end{aligned}$$

and complete the proof. \square

3. Parallelizable Algorithms. In this section, we introduce a parallelizable approach and one of its variant for optimization problem with orthogonality constraints (1.1). Both of these two approaches are based on the augmented Lagrangian function (2.5) and employ the new idea of updating the multipliers by explicit expression instead of dual ascent step in Algorithm 1.

Another distinction between our algorithms and the classical ALM is that the minimization subproblem for the prime variables is replaced by minimizing a proximal linearized approximation.

3.1. The Proximal Linearized Augmented Lagrangian Algorithm. We describe our main algorithm framework in Algorithm 2.

The main calculation costs of Algorithm 2 concentrate at Step 3 and 4. Step 3 only involves BLAS3 calculation.

Algorithm 2: Proximal Linearized Augmented Lagrangian Algorithm (PLAM)

```

1 Input: choose initial guess  $X^0$ , and set  $k := 0$ ;
2 while certain stopping criterion is not reached do
3   Compute the Lagrangian multipliers
      (3.1)  $\Lambda^k := \Psi(\nabla f(X^k)^\top X^k)$ .
4   Minimize the following proximal linearized Lagrangian function
      (3.2)  $X^{k+1} := \arg \min_{X \in \mathbb{R}^{n \times p}} \tilde{\mathcal{L}}_\beta(X) = \text{tr}(\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)^\top (X - X^k)) + \frac{\eta^k}{2} \|X - X^k\|_F^2$ .
5   Set  $k := k + 1$ .
6 Output:  $X^k$ .

```

The minimization subproblem (3.2) in Step 4 is nothing but a gradient step

$$\begin{aligned}
X^{k+1} &= X^k - \frac{1}{\eta^k} \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) \\
&= X^k - \frac{1}{\eta^k} \left(\nabla f(X^k) + \beta X^k \left(X^{k\top} X^k - I_p - \frac{1}{\beta} \Lambda^k \right) \right) \\
(3.3) \quad &= X^k - \frac{1}{\eta^k} \left(\nabla f(X^k) - X^k \Psi(\nabla f(X^k)^\top X^k) + \beta X^k (X^{k\top} X^k - I_p) \right),
\end{aligned}$$

where the last step is due to the updating formula (3.1). Apparently, the arithmetic operations involved in (3.3) belong to BLAS3 as well.

We notice that $1/\eta^k$ is nothing but the stepsize of gradient step. Hence, the proximal parameter η^k can be chosen in the same manner as how we choose stepsize for gradient methods. This issue will be described in details in Section 5.

3.2. Parallelizable Column-wise Block Minimization. An obvious demerit of PLAM is the boundedness of the iterate sequence can hardly be expected without any restriction on the penalty parameter β and the proximal parameter η^k . Theoretically, to guarantee the global convergence, β should be sufficiently large. Accordingly, η^k should be large as well which means sufficiently small stepsize is required and slow convergence can be expected. In fact, according to the empirical observations, the performance of PLAM is very sensitive to parameters β and η^k . In other word, it is not easy to tune these two parameters to guarantee good performance of Algorithm 2 in general.

Therefore, we put forward an upgraded version of PLAM. It is based on PLAM, but redundant column-wise unit sphere constraints are imposed to Step 4. Therefore, the proximal gradient takes the place of the gradient step in the Step 4 of Algorithm 2. With redundant constraints, the resulting iterate sequence will then be restricted to a compact set and hence bounded. We describe the framework of this upgraded PLAM in Algorithm 3.

Subproblem (3.5) in Algorithm 3 can be solved in a column-wisely parallel fashion. In fact, it is of closed-form solution

$$X_i^{k+1} = \frac{X_i^k - \frac{1}{\eta^k} \nabla_{X_i} \mathcal{L}_\beta(X^k, \Lambda^k)}{\left\| X_i^k - \frac{1}{\eta^k} \nabla_{X_i} \mathcal{L}_\beta(X^k, \Lambda^k) \right\|_2}.$$

For PCAL, we can update the Lagrangian multipliers in the same manner as PLAM, i.e. by formula (3.1). To

Algorithm 3: Parallelizable Column-wise Block Minimization for PLAM (PCAL)

```

1 Input: choose initial guess  $X^0$ , and set  $k := 0$ ;
2 while certain stopping criterion is not reached do
3   Compute the Lagrangian multipliers by (3.1) or
      (3.4)  $\Lambda^k := \Psi(\nabla f(X^k)^\top X^k) + \Phi\left(X^k{}^\top \nabla_X \mathcal{L}_\beta(X^k, \Psi(\nabla f(X^k)^\top X^k))\right)$ .
4   for  $i = 1, \dots, p$  do
5     Minimize the following proximal linearized Lagrangian function
      (3.5)  $X_i^{k+1} := \arg \min_{x \in \mathbb{R}^n} \tilde{\mathcal{L}}_\beta^{(i)}(x) = \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)_i^\top (x - X_i^k) + \frac{\eta^k}{2} \|x - X_i^k\|_2^2,$ 
      s. t.  $\|x\|_2 = 1$ .
6   Update  $X^{k+1} = [X_1^{k+1}, \dots, X_p^{k+1}]$ , and set  $k := k + 1$ .
7 Output:  $X^k$ .

```

obtain a better performance, we can also use the heuristic formula (3.4). The motivation of updating formula (3.4) comes from the following observation. In the KKT condition (2.1), we impose an additional term for the redundant sphere constraints. Namely,

$$(3.6) \quad \begin{cases} \nabla f(X) = X\Lambda + XD; \\ X^\top X = I_p, \end{cases}$$

where D is a diagonal matrix. Furthermore, D is determined by the Lagrangian multiplier of X_i in the subproblem (3.5).

3.3. Computational Cost. In this subsection, we compare the computational cost per iteration among MOp-tQR, PLAM and PCAL. The computational cost of the basic linear algebra operations and the overall costs of the aforementioned algorithms are listed in Table 3.1.

Here, those terms in red represent the corresponding operations that cannot be parallelized.

In practice, we calculate $X\Psi(\nabla f(X)^\top X)$ instead of $X(\nabla f(X)^\top X)$ for KKT evaluation since they are very close to each other around any first-order stationary point. Consequently, it saves $2np^2$ flops computational cost.

4. Convergence of PLAM. In this section, we focus on the theoretical analyses of our proposed PLAM. The global convergence, worst-case complexity and Q-linear local convergence rate will be established under different mild assumptions.

4.1. Global Convergence of PLAM. Besides blanket Assumption 1.1, to prove the convergence of Algorithm 2, we need to impose a mild condition on the initial guess, and restrictive conditions on β and η^k . To facilitate the narrative, we first state all these conditions here.

ASSUMPTION 4.1. *For a given X^0 , we say it is a qualified initial guess, if there exists $\underline{\sigma} \in (0, 1)$ so that*

$$\sigma_{\min}(X^0) \geq \underline{\sigma}, \quad 0 < \|X^0{}^\top X^0 - I_p\|_F \leq 1 - \underline{\sigma}^2.$$

Assumption 4.1 is not restrictive at all. Therefore two types of points satisfying this assumption and can be

evaluate function				
$f(X) := \frac{1}{2}\text{tr}(X^\top AX) + \text{tr}(G^\top X)$	AX	A: dense $2n^2p$	A: sparse $O(np)$	A: sparse $O(np)$
	$\nabla f(X) = AX + G$	np		
	$\frac{1}{2}\text{tr}(X^\top AX) + \text{tr}(G^\top X)$	$4np$		
KKT: $\nabla f(X) - X\nabla f(X)^\top X$				
$\nabla f(X)^\top X$	$2np^2$			$4np^2 + np$
$X(\nabla f(X)^\top X)$	$2np^2$			
feasibility: $X^\top X - I_p$				
$X^\top X$	np^2			$np^2 + np$
solvers				
PLAM	$X(X^\top X - I_p)$	$2np^2$		$4np^2 + O(np)$
	$X\Psi(\nabla f(X)^\top X)$	$2np^2$		
PCAL	$X(X^\top X - I_p)$	$2np^2$		$4np^2 + O(np)$
	$X\Psi(\nabla f(X)^\top X)$	$2np^2$		
	$\Phi\left(X^{k^\top}\nabla_X L_\beta(X^k, \Psi(\nabla f(X^k)^\top X^k))\right)$	$O(np)$		
	$X\Lambda = X\Psi(\cdot) + X\Phi(\cdot)$	$O(np)$		
MOptQR (cholesky LL^\top)	$V := X - \tau(\nabla f(X) - X\nabla f(X)^\top X)$	$2np$		$3np^2 + O(p^3) + O(np)$
	$V^\top V$	np^2		
	$\text{chol}(V^\top V) = LL^\top$	$p^3/3$		
	VL^{-1}	$2np^2 + O(p^3)$		
MOptQR (Gram-Schmidt)	$2np^2$			$2np^2 + O(np)$
in total				
PLAM	$7np^2 + O(np)$			
PCAL	$7np^2 + O(np)$			
MOptQR	$7np^2 + O(p^3) + O(np)$ for cholesky, $4np^2 + 2np^2 + O(np)$ for Gram-Schmidt			

TABLE 3.1

The comparison of computational cost

obtained easily:

- (i) $X^0 = Q\Sigma$, where $Q \in \mathcal{S}_{n,p}$, $\Sigma = \text{Diag}(\underbrace{1, \dots, 1}_{p-1}, \sqrt{1 - \underline{\sigma}^2})$;
- (ii) $X^0 \notin \mathcal{S}_{n,p}$ satisfying $\sigma_{\min}^2(X^0) > 1 - \frac{1}{\sqrt{p}}$ and $\sigma_{\max}^2(X^0) < 1 + \frac{1}{\sqrt{p}}$.

Now, we list all the special notations to be used in this section.

$$(4.1) \quad \begin{aligned} R &= \|X^{0^\top} X^0 - I_p\|_F; \quad \mathcal{C} = \{X \mid \|X^\top X - I_p\|_F \leq R\}; \quad \underline{f} = \min_{X \in \mathcal{C}} f(X); \\ M &= \max_{X \in \mathcal{C}} \|X\|_2; \quad N = \max_{X \in \mathcal{C}} \|\nabla f(X)\|_F; \quad L = \max_{X \in \mathcal{C}} \|\nabla^2 f(X)\|_2. \end{aligned}$$

We introduce the following merit function

$$(4.2) \quad h(X) = f(X) - \frac{1}{2} \langle \Psi(\nabla f(X)^\top X), X^\top X - I_p \rangle + \frac{\beta}{4} \|X^\top X - I_p\|_F^2.$$

According to the twice continuous differentiability of $f(X)$, $\nabla f(X)$ is Lipschitz continuous on the compact set \mathcal{C} . Namely, there exists constant $L_h > 0$, related to β , so that

$$(4.3) \quad \|\nabla h(X) - \nabla h(Y)\|_F \leq L_h \|X - Y\|_F, \quad \forall X, Y \in \mathcal{C}.$$

The algorithm parameters β and η^k , and the constants used in the proof can be selected by the following rules.

ASSUMPTION 4.2.

$$(4.4) \quad c_1 \in \left(0, \frac{1}{2}\right); \quad \beta > \max \left\{ \frac{MN}{\underline{\sigma}^2} + \sqrt{\frac{M^2 N^2}{\underline{\sigma}^4} + \frac{(N + LM)^2}{4\underline{\sigma}^2(1 - 2c_1)}}, \frac{MN}{\underline{\sigma}}, \frac{4MN}{\underline{\sigma}^2} \right\};$$

$$(4.5) \quad c_2 \in \left(0, \frac{R^2(\beta\underline{\sigma}^2 - 4MN)}{2N_L^2}\right]; \quad \eta^k \in [\underline{\eta}, \bar{\eta}],$$

$$\text{where } \underline{\eta} = \max \left\{ \frac{L_h}{2c_1}, \frac{2N_L M + N_L \sqrt{4M^2 + 2R}}{R}, \frac{R + 2M^2}{c_2} \right\},$$

$$N_L = (1 + M^2)N + \beta RM, \quad \bar{\eta} \geq \underline{\eta}.$$

- N_L :

$$\begin{aligned} \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F &= \left\| \nabla f(X^k) - X^k \Psi(\nabla f(X^k)^\top X^k) + \beta X^k (X^{k\top} X^k - I_p) \right\|_F \\ &\leq (1 + M^2)N + \beta RM = N_L \end{aligned}$$

Now we give the sketch of our proof. Suppose $\{X^k\}$ is the iterate sequence generated by Algorithm 2. The main steps of the proof include:

- (1) Any iterate X^k is in \mathcal{C} , and $\underline{\sigma}$ is a unified lower bound of the smallest singular values of the iterates X^k ;
- (2) The merit function $h(X)$ is bounded below;
- (3) $\{h(X^k)\}$ monotonically decreases, and hence $\{X^k\}$ has at least one convergent subsequence;
- (4) Any cluster point of $\{X^k\}$, say X^* , is a first-order stationary point of minimizing the augmented Lagrangian function (2.9) with $\Lambda^* = \Psi(\nabla f(X^*)^\top X^*)$;
- (5) Any cluster point of $\{X^k\}$, say X^* , is a first-order stationary point of the original optimization problem with orthogonality constraints (1.1).

Next we provide five concrete lemmas or corollaries following the above-mentioned sketch.

LEMMA 4.3. *Suppose $\{X^k\}$ is the iterate sequence generated by Algorithm 2 initiated from X^0 satisfying Assumption 4.1, and the problem parameters satisfy Assumption 4.2. Then it holds that*

$$(4.6) \quad \sigma_{\min}(X^k) \geq \underline{\sigma}, \quad X^k \in \mathcal{C}.$$

Proof. We use mathematical induction. The argument (4.6) directly holds for X^0 resulting from Assumption 4.1. Next we investigate whether (4.6) holds at X^{k+1} provided that it holds for X^k .

Case I, $\|X^{k\top} X^k - I_p\|_F \leq \frac{R}{2}$. We have

$$\begin{aligned} &\|X^{k+1\top} X^{k+1} - I_p\|_F \\ &= \left\| \left(X^k - \frac{1}{\eta^k} \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) \right)^\top \left(X^k - \frac{1}{\eta^k} \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) \right) - I_p \right\|_F \\ &\leq \|X^{k\top} X^k - I_p\|_F + \frac{2}{\eta^k} \|X^k\|_2 \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F + \frac{1}{(\eta^k)^2} \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2. \end{aligned}$$

It is not difficult to verify that $\|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F \leq N_L$ holds for any $X^k \in \mathcal{C}$. By using the facts $X^k \in \mathcal{C}$, (4.1) and (4.5), we have

$$\frac{2}{\eta^k} \|X^k\|_2 \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F + \frac{1}{(\eta^k)^2} \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2 \leq \frac{R}{2},$$

which implies $\|X^{k+1\top} X^{k+1} - I_p\|_F \leq R$.

Case II, $\|X^{k\top} X^k - I_p\|_F > \frac{R}{2}$. For convenience, we denote $c(X) = \frac{1}{2} \|X^\top X - I_p\|_F^2$,

$$(4.7) \quad d = \nabla f(X^k) - X^k \Lambda^k, \quad C = X^{k\top} X^k - I_p, \quad \delta = X^k C.$$

According to the facts $\sigma_{\min}(X^k) \geq \underline{\sigma}$ and $X^k \in \mathcal{C}$, we have

$$(4.8) \quad \|\delta\|_{\text{F}} > \frac{R\underline{\sigma}}{2}.$$

By using the fact that $\text{tr}(AB) = \text{tr}(AB^{\top})$ if A is symmetric, we have

$$\text{tr}(CX^{k\top} \nabla f(X^k)) = \text{tr}(C \nabla f(X^k)^{\top} X^k) = \text{tr}(C\Lambda^k).$$

Hence, we have

$$(4.9) \quad \begin{aligned} \langle d, \delta \rangle &= \text{tr}(CX^{k\top} \nabla f(X^k) - CX^{k\top} X^k \Lambda^k) \\ &= \text{tr}(CX^{k\top} \nabla f(X^k) - C(C + I_p)\Lambda^k) = -\text{tr}(C^2 \Lambda^k). \end{aligned}$$

Notice that $L_c = 2R + 4M^2$ is the Lipschitz constant of $\nabla c(X)$ over \mathcal{C} . Due to the fact (4.5), (4.8) and (4.9), we have

$$\begin{aligned} &\text{tr}(\nabla_X \mathcal{L}_{\beta}(X^k, \Lambda^k)^{\top} \nabla c(X^k)) - c_2 \|\nabla_X \mathcal{L}_{\beta}(X^k, \Lambda^k)\|_{\text{F}}^2 \\ &\geq 2\langle d + \beta\delta, \delta \rangle - c_2 N_L^2 = 2\beta \|\delta\|_{\text{F}}^2 + 2\langle d, \delta \rangle - c_2 N_L^2 \\ &> \frac{\beta R^2 \underline{\sigma}^2}{2} - 2\|C\|_{\text{F}}^2 \cdot \text{tr}(\Lambda^k) - c_2 N_L^2 \\ &\geq \frac{\beta R^2 \underline{\sigma}^2}{2} - 2R^2 MN - c_2 N_L^2 \geq 0. \end{aligned}$$

According to the Taylor expansion, we have

$$\begin{aligned} c(X^{k+1}) &= c\left(X^k - \frac{1}{\eta^k} \nabla_X \mathcal{L}_{\beta}(X^k, \Lambda^k)\right) \\ &\leq c(X^k) - \frac{1}{\eta^k} \langle \nabla_X \mathcal{L}_{\beta}(X^k, \Lambda^k), \nabla c(X^k) \rangle + \frac{L_c}{2(\eta^k)^2} \|\nabla_X \mathcal{L}_{\beta}(X^k, \Lambda^k)\|_{\text{F}}^2 \\ &< c(X^k) - \left(\frac{c_2}{\bar{\eta}} - \frac{L_c}{2\bar{\eta}^2}\right) \cdot \|\nabla_X \mathcal{L}_{\beta}(X^k, \Lambda^k)\|_{\text{F}}^2 \leq c(X^k). \end{aligned}$$

According to assumption, $R \leq 1 - \underline{\sigma}^2$, we can easily obtain that $\sigma_{\min}(X^{k+1}) \geq \underline{\sigma}$. This completes the proof. \square

LEMMA 4.4. $h(X)$ defined by (4.2) is bounded below at \mathcal{C} .

This lemma immediately holds by the continuous differentiability of $h(X)$ and the compactness of \mathcal{C} , and hence, the proof is omitted.

LEMMA 4.5. Suppose $\{X^k\}$ is the iterate sequence generated by Algorithm 2 initiated from X^0 satisfying Assumption 4.1, the problem parameters satisfy Assumption 4.2, and $h(X)$ is defined by (4.2). Then it holds that

$$(4.10) \quad h(X^k) - h(X^{k+1}) \geq c_3 \|\nabla_X \mathcal{L}_{\beta}(X^k, \Lambda^k)\|_{\text{F}}^2,$$

where $c_3 = \frac{c_1}{\bar{\eta}} - \frac{L_h}{2\bar{\eta}^2} > 0$.

Proof. Firstly, we notice that

$$\nabla h(X) = \nabla_X \mathcal{L}_{\beta}(X, \Psi(\nabla f(X)^{\top} X)) - \frac{1}{2}(\nabla^2 f(X)[X] + \nabla f(X))(X^{\top} X - I_p).$$

We keep using the notations (4.7) and investigate

$$\begin{aligned}
& \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2 - \frac{1}{1-2c_1} \|\nabla h(X^k) - \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2 \\
& \geq \|d + \beta\delta\|_F^2 - \frac{(N+LM)^2}{4(1-2c_1)} \|C\|_F^2 \geq 2\beta\langle d, \delta \rangle + \beta^2 \|\delta\|_F^2 - \frac{(N+LM)^2}{4(1-2c_1)} \|C\|_F^2 \\
& \geq -\beta \|C\|_F^2 \cdot \text{tr}(\Lambda^k) + \left(\beta^2 \underline{\sigma}^2 - \frac{(N+LM)^2}{4(1-2c_1)} \right) \cdot \|C\|_F^2 \\
& \geq -2\beta MN \|C\|_F^2 + \left(\beta^2 \underline{\sigma}^2 - \frac{(N+LM)^2}{4(1-2c_1)} \right) \cdot \|C\|_F^2 \geq 0,
\end{aligned}$$

where the second last inequality is implied by the relationship (4.9). Hence, we arrive at

$$(4.11) \quad \langle \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k), \nabla h(X^k) \rangle \geq c_1 \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2.$$

Substituting (4.5) and (4.11) into the Taylor expansion, we have

$$\begin{aligned}
h(X^{k+1}) &= h\left(X^k - \frac{1}{\eta^k} \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\right) \\
&\leq h(X^k) - \frac{1}{\eta^k} \langle \nabla_X \nabla h(X^k), \mathcal{L}_\beta(X^k, \Lambda^k) \rangle + \frac{L_h}{2(\eta^k)^2} \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2 \\
&\leq h(X^k) - \left(\frac{c_1}{\eta} - \frac{L_h}{2\eta^2} \right) \cdot \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2.
\end{aligned}$$

We complete the proof. \square

With the boundedness of $h(X)$ at \mathcal{C} , Lemma 4.5 immediately implies the convergent of $\{h(X^k)\}$. More precisely, we have the following corollary.

COROLLARY 4.6. *Suppose $\{X^k\}$ is the iterate sequence generated by Algorithm 2 initiated from X^0 satisfying Assumption 4.1, and the problem parameters satisfy Assumption 4.2. Then the algorithm is finitely terminated at k -th iteration with $\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) = 0$, or*

$$\lim_{k \rightarrow +\infty} \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) = 0.$$

Moreover, $\{X^k\}$ has at least one convergent subsequence. Any cluster point of $\{X^k\}$, X^* , is a first-order stationary point of minimizing the augmented Lagrangian function (2.9) with $\Lambda^* = \Psi(\nabla f(X^*)^\top X^*)$.

Proof. This is a direct corollary of Lemmas 4.3 and 4.5. \square

Finally, we give the global convergence rate of PLAM, namely, the worst case complexity.

THEOREM 4.7. *Suppose $\{X^k\}$ is the iterate sequence generated by Algorithm 2 initiated from X^0 satisfying Assumption 4.1, and the problem parameters satisfy Assumption 4.2. Then the sequence $\{X^k\}$ has at least one cluster point, and any which is a first-order stationary point of problem (1.1). More precisely, for any $K > 1$, it holds*

$$(4.12) \quad \min_{k=0, \dots, K-1} \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F < \sqrt{\frac{f(X^0) - \underline{f} + MNR + \beta R^2/4}{c_3 K}}$$

Proof. The first part directly holds from Corollaries 4.6 and Lemma 2.4. Recalling Lemma 4.5, we have

$$(4.13) \quad h(X^0) - \min_{X \in \mathcal{C}} h(X) \geq h(X^0) - h(X^K) \geq \sum_{k=0}^{K-1} c_3 \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2$$

$$(4.14) \quad \geq c_3 K \cdot \min_{k=0, \dots, K-1} \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F^2.$$

Moreover, we have

$$(4.15) \quad h(X^0) \leq f(X^0) + \frac{1}{2}MNR + \frac{\beta}{4}R^2, \quad \min_{X \in \mathcal{C}} h(X) \geq \underline{f} - \frac{1}{2}MNR.$$

Combining (4.13)-(4.15), we arrive at the argument (4.12). \square

COROLLARY 4.8. *Suppose all the assumptions of Theorem 4.7 hold. Besides, for a given positive parameter δ , it holds that $\beta > (MN + \delta)/\underline{\sigma}$, then it holds that*

$$\min_{k=0, \dots, K-1} \max \left\{ \|I_p - X^k{}^\top X^k\|_F, \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F \right\} < \max \left\{ \frac{M}{\delta}, 1 \right\} \sqrt{\frac{f(X^0) - \underline{f} + MNR + \beta R^2/4}{c_3 K}}.$$

Proof. This is a direct corollary of Lemma 2.4 and Theorem 4.7. \square

REMARK 4.9. *The sublinear convergence rate of Corollary 4.8 actually tells us that Algorithm 2 terminates after $O(1/\epsilon^2)$ iterations, if the stopping criterion is set as $\max \left\{ \|I_p - X^k{}^\top X^k\|_F, \|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)\|_F \right\} < \epsilon$.*

4.2. Local Convergence Rate of PLAM and PCAL. In this subsection, we consider the local convergence of PLAM once the optimization problem with orthogonality constraints (1.1) has an isolated local minimizer.

THEOREM 4.10. *Suppose X^* is an isolated minimizer of (1.1), and we denote*

$$\tau := \inf_{0 \neq Y \in \mathcal{T}(X)} \frac{\text{tr}(Y^\top \nabla^2 f(X)[Y] - \Lambda Y^\top Y)}{\|Y\|_F^2}.$$

The algorithm parameters satisfy $\beta \geq \frac{L+MN+\tau}{2}$ and $\eta^k \in [\underline{\eta}, \bar{\eta}]$, where $\bar{\eta} \geq \underline{\eta} \geq L + MN + 2\beta$. Then, there exists $\varepsilon > 0$ such that starting from any X^0 satisfying $\|X^0 - X^\|_F < \varepsilon$, and the iterate sequence $\{X^k\}$ generated by Algorithm 2 converges to X^* Q -linearly.*

Proof. We study the iterate formula (3.2).

$$\begin{aligned} X^{k+1} &= X^k - \frac{1}{\eta^k} \nabla_X \mathcal{L}_\beta(X^k, \Psi(\nabla f(X^k)^\top X^k)); \\ X^* &= X^* - \frac{1}{\eta^k} \nabla_X \mathcal{L}_\beta(X^*, \Psi(\nabla f(X^*)^\top X^*)). \end{aligned}$$

Subtracting the second one from the first one and using the Taylor expansion, we have

$$(4.16) \quad \delta^{k+1} = \delta^k - \frac{1}{\eta^k} \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \Psi(\nabla f(X^*)^\top X^*))[\delta^k] + o(\|\delta^k\|),$$

where $\delta^k = X^k - X^*$. Recall the expression of Hessian (2.13), the fact that $\nabla f(X^*)^\top X^* = \Psi(\nabla f(X^*)^\top X^*)$ and the assumption on η , we have

$$(4.17) \quad \left\| \frac{1}{\eta^k} \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*)[\delta^k] \right\|_F \leq \|\delta^k\|_F.$$

On the other hand, δ^k can be decomposed as the summation of three parts:

$$(4.18) \quad \delta^k = X^*S + X^*W + K,$$

where $S \in \mathbb{R}^{p \times p}$ is symmetric, $W \in \mathbb{R}^{p \times p}$ is skew-symmetric, $K \in \mathbb{R}^{n \times p}$ is perpendicular to X^* . Since X^* is a strict local minimizer, and $\mathcal{T}(X)$ is closed, we have $\tau > 0$. Hence, it holds that

$$(4.19) \quad \text{tr} \left((X^*W + K)^\top \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*) [X^*W + K] \right) \geq \tau \|X^*W + K\|_F^2,$$

as $X^*W + K \in \mathcal{T}(X)$. Moreover, it follows from the assumption on β that

$$(4.20) \quad \begin{aligned} & \text{tr} \left((X^*S)^\top \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*) [X^*S] \right) \\ &= \text{tr}(SX^*\nabla^2 f(X^*)X^*S - S^\top \nabla f(X^*)^\top X^* + 2\beta S^2) \geq \tau \|X^*S\|_F^2. \end{aligned}$$

Combining (4.19), (4.20), the symmetry of S , the skew symmetry of W , $K^\top X^* = 0$ together with the assumption on η , we arrive at

$$(4.21) \quad \begin{aligned} & \text{tr} \left(\delta^k{}^\top \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*) [\delta^k] \right) \\ &= \text{tr} \left((X^*W + K)^\top \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*) [X^*W + K] \right) \\ & \quad + \text{tr} \left((X^*W + K)^\top \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*) [X^*S] \right) \\ & \quad + \text{tr} \left((X^*S)^\top \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*) [X^*W + K] \right) \\ & \quad + \text{tr} \left((X^*S)^\top \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*) [X^*S] \right) \\ & \geq \tau \|X^*W + K\|_F^2 + \tau \|X^*S\|_F^2 = \tau \|\delta^k\|_F^2. \end{aligned}$$

Notice that (4.17) implies the positive semi-definiteness of the linear operator

$$I - \frac{1}{\eta^k} \nabla_{XX}^2 \mathcal{L}_\beta(X^*, \nabla f(X^*)^\top X^*).$$

Together with (4.21), we can conclude that

$$\|\delta^{k+1}\|_F \leq (1 - \tau) \|\delta^k\|_F + o(\|\delta^k\|),$$

which completes the proof. \square

REMARK 4.11. *The global and local convergence of PCAL can be established in the same manner as PLAM, if the multipliers are updated by the same formula, (3.1), as PLAM.*

5. Numerical Experiments. In this section, we evaluate the numerical performance of our proposed algorithms PLAM and PCAL. We first introduce the implementation details and the testing problems in Subsection 5.1 and 5.2, respectively. Then, we report the numerical experiments which are mainly of three folds.

In the first part, we mainly determine the default settings of our proposed algorithms, which will be discussed in Subsection 5.3. Then, in Subsection 5.4, we compare our PLAM and PCAL with a few existing solvers by testing a bunch of instances, which are chosen from a MATLAB toolbox KSSOLV [35]. All the algorithms tested in the first two parts are run in serial. The corresponding experiments are performed on a workstation with one Intel(R) Xeon(R) Processor E5-2697 v2 (at 2.70GHz \times 12, 30M Cache) and 128GB of RAM running in MATLAB R2016b under Ubuntu

12.04.

Finally, we investigate the parallel efficiency of PCAL by comparing with a parallelized version of MOptQR in Subsection 5.5. All the experiments in this subsection are performed on a single node of LSSC-IV⁴, which is a high-performance computing cluster (HPCC) maintained at the State Key Laboratory of Scientific and Engineering Computing (LSEC), Chinese Academy of Sciences. The operating system of LSSC-IV is Red Hat Enterprise Linux Server 7.3. This node called “b01” consists of two Intel(R) Xeon(R) Processor E7-8890 v4 (at 2.20GHz×24, 60M Cache) with 4TB shared memory. The total number of processor cores in this node is 96.

5.1. Implementation Details. There are two parameters in our algorithms PLAM and PCAL. According to Theorem 4.7, the penalty parameter β for PLAM should be sufficiently large. Although we can estimate a suitable β to satisfy the assumption of the theorem, it would be too large in practice. In the numerical experiments, we set β as an upper bound of $s := \|\nabla^2 f(0)\|_2$ for PLAM, and 1 for PCAL.

Another one is the proximal parameter η , whose reciprocal is the step size of the gradient step in Algorithm 2 and 3. Similar to β , we can not use the rigorous restriction in the theoretical analysis. In practice, we have the following strategies to choose this parameter:

- (i) $\eta_C^k := \gamma$, where $\gamma > 0$ is a sufficiently large constant.
- (ii) Differential approximation:

$$\eta_D^k := \frac{\|\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) - \nabla_X \mathcal{L}_\beta(X^{k-1}, \Lambda^{k-1})\|_F}{\|X^k - X^{k-1}\|_F}.$$

- (iii) Barzilai-Borwein(BB) strategy [2]:

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

where

$$S^k = X^k - X^{k-1}, \quad Y^k = \nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) - \nabla_X \mathcal{L}_\beta(X^{k-1}, \Lambda^{k-1}).$$

- (iv) Alternating BB strategy [9]:

$$\eta_{ABB}^k := \begin{cases} \eta_{BB1}^k, & \text{for odd } k, \\ \eta_{BB2}^k, & \text{for even } k. \end{cases}$$

If there is no special mention, the stopping criterion used for both serial and parallel experiments can be described as follows,

$$\frac{\|\nabla f(X) - X \nabla f(X)^\top X\|_F}{\|\nabla f(X^0) - X^0 \nabla f(X^0)^\top X^0\|_F} < 10^{-8}.$$

The maximum number of iteration for all those solvers is set to 3000 without special mention.

5.2. Testing Problems. In this subsection, we introduce six types of problems which will be used in the numerical experiments.

⁴More information at <http://lsec.cc.ac.cn/chinese/lsec/LSSC-IVintroduction.pdf>

Problem 1: A simplification of discretized Kohn-Sham total energy minimization.

$$(5.1) \quad \begin{aligned} \min_{X \in \mathbb{R}^{n \times p}} \quad & \frac{1}{2} \text{tr}(X^\top L X) + \frac{\alpha}{4} \rho(X)^\top L^\dagger \rho(X) \\ \text{s.t.} \quad & X^\top X = I_p, \end{aligned}$$

where the matrix $L \in \mathbb{S}^n$ and $\rho(X) := \text{diag}(X X^\top)$. In the numerical experiments, we set $\alpha = 1$, and L is randomly generated by Gauss distribution, i.e., $L = \text{randn}(n)$ in MATLAB language, and set $L := \frac{1}{2}(L + L^\top)$. In this instance, $s = \|L\|_2$.

Problem 2: A class of quadratic minimization with orthogonality constraints.

$$(5.2) \quad \begin{aligned} \min_{X \in \mathbb{R}^{n \times p}} \quad & \frac{1}{2} \text{tr}(X^\top A X) + \text{tr}(G^\top X) \\ \text{s.t.} \quad & X^\top X = I_p, \end{aligned}$$

where the matrices $A \in \mathbb{S}^n$ and $G \in \mathbb{R}^{n \times p}$. This problem is adequately discussed in [11]. In the numerical experiments, the matrices A and G are randomly generated in the same manner as in [11]. Namely,

$$(5.3) \quad A := P \Lambda P^\top,$$

$$(5.4) \quad G := \kappa \cdot Q D,$$

where the matrices $P = \text{qr}(\text{rand}(n, n)) \in \mathbb{R}^{n \times n}$, $\tilde{Q} = \text{rand}(n, p) \in \mathbb{R}^{n \times p}$, $Q \in \mathbb{R}^{n \times p}$ and $Q_i = \tilde{Q}_i / \|\tilde{Q}_i\|_2$ ($i = 1, 2, \dots, p$), and matrices $\Lambda \in \mathbb{R}^{n \times n}$ and $D \in \mathbb{R}^{p \times p}$ are diagonal matrices with

$$(5.5) \quad \Lambda_{ii} := \begin{cases} \theta^{1-i}, & \text{rand}(1, 1) < \xi \\ -\theta^{1-i}, & \text{rand}(1, 1) \geq \xi \end{cases} \quad \forall i = 1, 2, \dots, n,$$

$$(5.6) \quad D_{jj} := \zeta^{j-1}, \quad \forall j = 1, 2, \dots, p.$$

Here, parameter $\theta \geq 1$ determines the decay of eigenvalues of A ; parameter $\zeta \geq 1$ refers to the growth rate of column's norm of G . The parameter $\kappa > 0$ represents the scale difference between the quadratic term and the linear term. The default setting of these parameters are $\kappa = 1, \theta = 1.01, \zeta = 1.01, \xi = 1$. In this instance, $s = \|A\|_2$.

Problem 3: Rayleigh-Ritz trace minimization, which is a special case of Problem 2.

$$(5.7) \quad \begin{aligned} \min_{X \in \mathbb{R}^{n \times p}} \quad & \frac{1}{2} \text{tr}(X^\top A X) \\ \text{s.t.} \quad & X^\top X = I_p, \end{aligned}$$

where the matrices $A \in \mathbb{S}^n$. In our experiments, the matrix A is generated in the same manner as in Problem 2. In this instance, $s = \|A\|_2$.

Problem 4: Another class of quadratic minimization with orthogonality constraints.

$$(5.8) \quad \begin{aligned} \min_{X \in \mathbb{R}^{n \times p}} \quad & \frac{1}{2} \text{tr}(A^\top X B X^\top) \\ \text{s.t.} \quad & X^\top X = I_p. \end{aligned}$$

where the matrices $A \in \mathbb{S}^n$ and $B \in \mathbb{S}^p$. This problem is out of the scope of problems discussed in [11], but can be solved by PLAM or PCAL. The matrices A and B are randomly generated by $A = \text{randn}(n)$, $A := \frac{1}{2}(A + A^\top)$ and $B = \text{randn}(p)$, $B := \frac{1}{2}(B + B^\top)$, respectively. In this instance, $s = \|A\|_2 \cdot \|B\|_2$

Problem 5: Discretized Kohn-Sham total energy minimization instances from KSSOLV [35].

$$(5.9) \quad \min_{X \in \mathbb{R}^{n \times p}} E(X), \quad \text{s. t. } X^\top X = I_p,$$

where the discretized Kohn-Sham total energy function $E(X)$ is defined by (1.2). All the data comes from MATLAB toolbox KSSLOV.

Problem 6: A synthetic instance of discretized Kohn-Sham total energy minimization.

$$(5.10) \quad \min_{X \in \mathbb{R}^{n \times p}} \quad \frac{1}{2} \text{tr}(X^\top LX) + \frac{1}{2} \rho(X)^\top L^\dagger \rho(X) - \frac{3}{4} \gamma \rho(X)^\top \rho(X)^{\frac{1}{3}} \\ \text{s. t.} \quad X^\top X = I_p,$$

where the matrix $L \in \mathbb{R}^{n \times n}$ and $\rho(X) := \text{diag}(XX^\top)$. The parameter $\gamma = 2(\frac{3}{\pi})^{1/3}$ and $\rho(X)^{\frac{1}{3}}$ denotes the component-wise cubic root of the vector $\rho(X)$. This problem adopts a special exchange functional $-\frac{3}{4} \gamma \rho(X)^\top \rho(X)^{\frac{1}{3}}$ (the correlation term is ignored), which is introduced in [22]. The generation of L is in the same manner as in Problem 1.

5.3. Default Settings. In this subsection, we determine the default settings for the proposed algorithms PLAM and PCAL.

In the first experiment, we test PLAM and PCAL with these four different choices of η^k on Problem 1-4. Here, we only illustrate the results of η_{BB1} for strategy (iii), since its performances overwhelms those with η_{BB2} . The penalty parameter is fixed as $\beta = s + 0.1$. FIG. 5.1 shows the results of PLAM and PCAL with different η^k . From subfigures (a)-(d), we observe that PLAM with η_{ABB} outperforms others. Under the same setting, a comparison among PCAL with different η^k is reported in subfigures (e)-(h). We notice that PCAL with η_{ABB} is superior to the other η^k choices. Then we set η_{ABB} as the default setting for PLAM and PCAL.

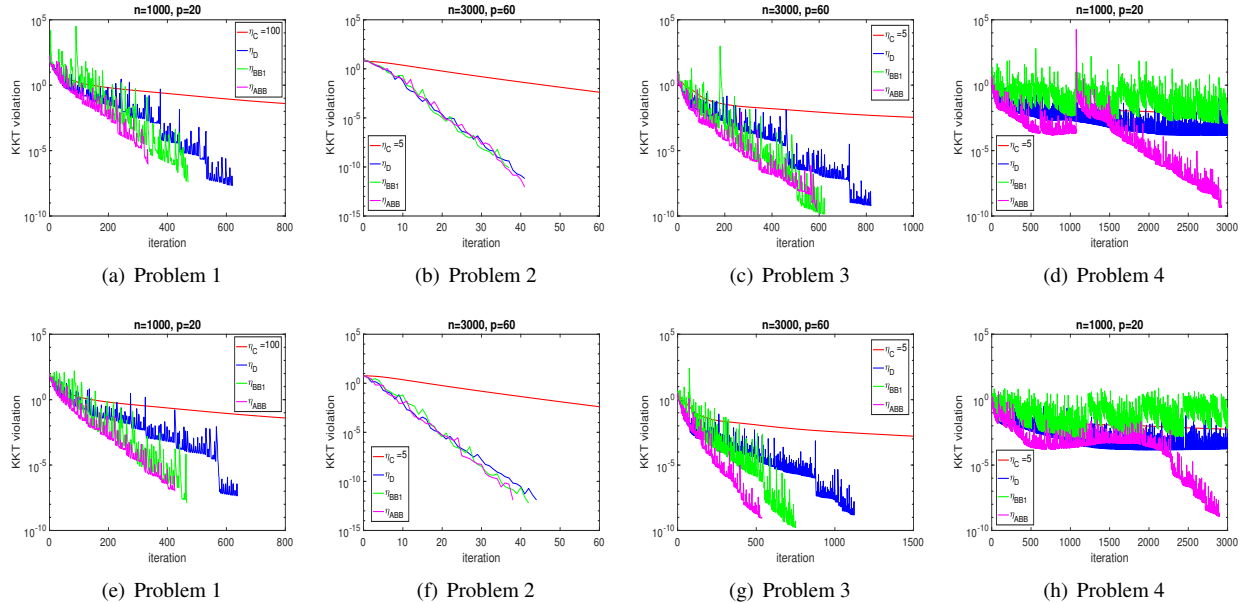


FIG. 5.1. A comparison of KKT violation for PLAM (a)-(d) and PCAL (e)-(h) with different η ($\beta = s + 0.1$)

We next compare the performance among PLAM and PCAL variations corresponding to different β . In the comparison, we set β varying among $0, 0.01s, 0.1s, s + 0.1, 10s + 1$. The proximal parameter is fixed as its default

$\eta = \eta_{\text{ABB}}$. We present all the numerical results in Figure 5.2. We notice from subfigures (a)-(d) that PLAM with small β might be divergent in some cases, while large β causes slow convergence. Therefore, a suitable chosen β , often unreachable in practice, is necessary for good performance of PLAM. On the other hand, the dependence on β of PCAL can be learnt from subfigures (e)-(h). The smaller β for PCAL has the better performance in some instances, and the behavior of PCAL is completely not sensitive to β in other instances. To take more distinctive look at the difference between PLAM and PCAL, we present a comparison in Figure 5.3. Therefore, in practice, we suggest an approximation of s to be the default β of PLAM and 1 for PCAL. Since it is easier to tune β for PCAL than PLAM, we choose PCAL to be the default algorithm of ours in Subsection 5.5.

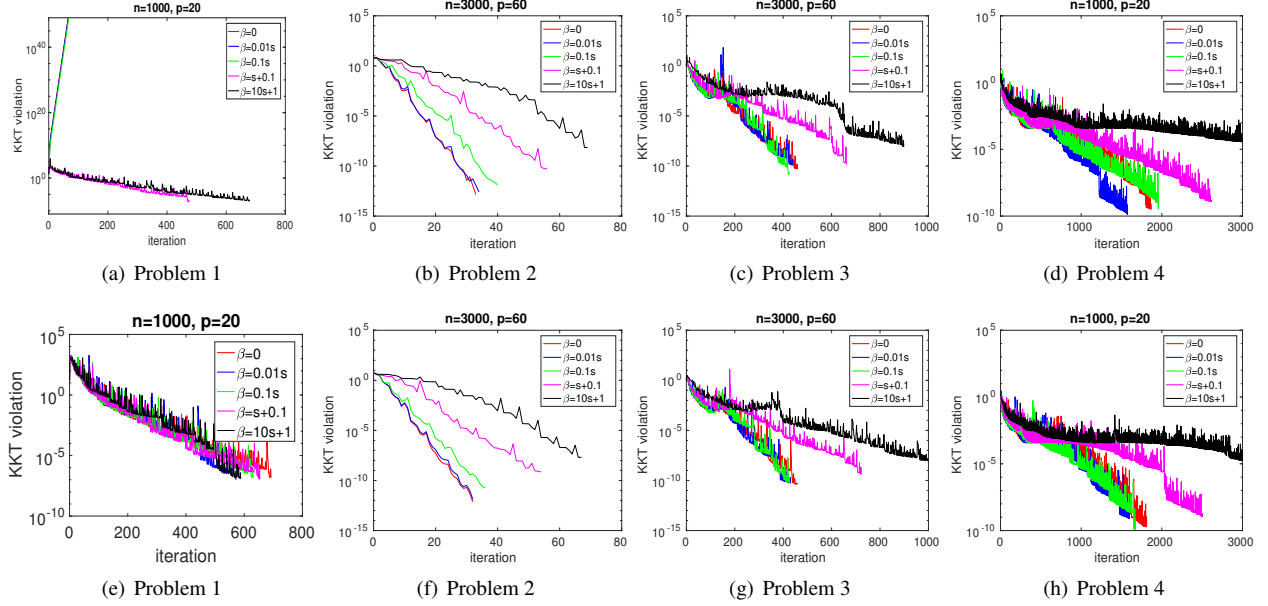


FIG. 5.2. A comparison of KKT violation for PLAM (a)-(d) and PCAL (e)-(h) with different β ($\eta = \eta_{\text{ABB}}$)

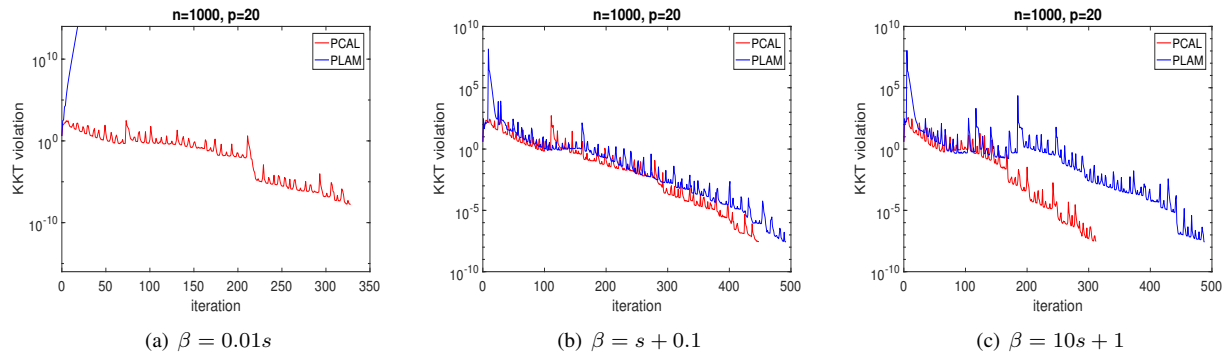


FIG. 5.3. A comparison between PLAM and PCAL with different β on Problem 1

There are two distinctions between PLAM and ALM. Firstly, a gradient step takes the place of solving the subproblem to some given precision in the update of the prime variables. Secondly, a closed-form expression is used to update the Lagrangian multipliers in stead of dual ascend. In order to show that the new update formula for multipliers is a crucial fact of the efficiency of PLAM and PCAL, we compare PLAM and PCAL with PLAM-DA and PCAL-DA, respectively. Here PLAM-DA and PCAL-DA stand for Algorithm 2 and 3 with Step 3 using dual ascend to

update the multipliers, respectively. We report the numerical results in Figure 5.4. It can be observed that the closed-form expression for updating Lagrangian multipliers is superior to dual ascend in solving optimization problems with orthogonality constraints.

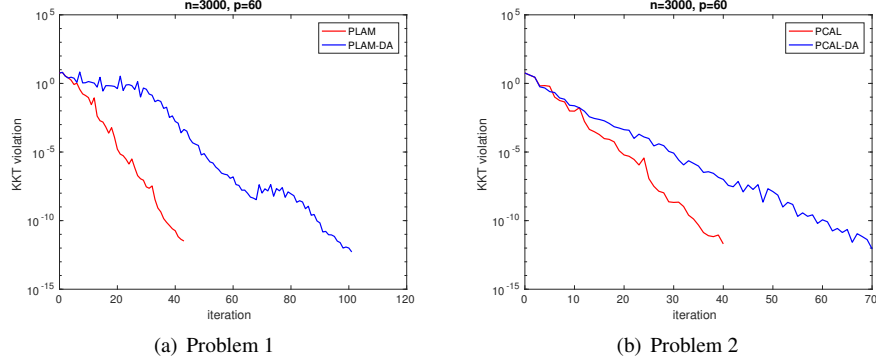


FIG. 5.4. A comparison between PLAM and PCAL on multiplier

In the end of this subsection, we show how KKT and feasibility violations decay in the iterations, when PLAM and PCAL are used to solve Problem 1. The numerical results are presented in Figure 5.5. We notice that the decay of feasibility violations is nonmonotone and has a similar variation tendency as KKT violations, which coincides our theoretical analysis Lemma 2.4. If we want a high accuracy for the feasibility but a mild one for KKT conditions, we can set a mild tolerance for KKT violation and impose an orthonormalization step as a post process when we obtain the last iterate by PLAM or PCAL. Table 5.1 illustrates that such post process does not affect the KKT violation, but do improve the feasibility. Here, "stop" and "orth" represent the relative values at the last iterate and the one after post process, respectively. Hereinafter, the orthonormalization post process, achieved by an internal function $\text{qr}(\cdot)$ assembled in MATLAB, is the default last step of PLAM and PCAL.

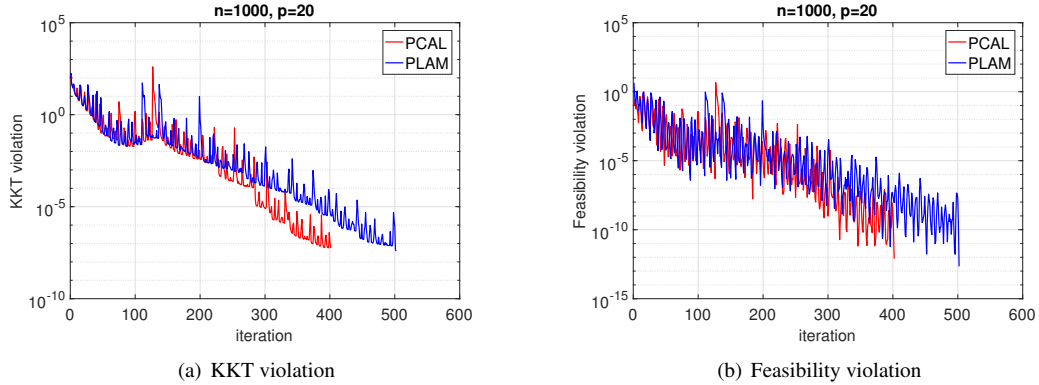


FIG. 5.5. The results of KKT and Feasibility violation for PLAM and PCAL on Problem 1

5.4. Kohn-Sham Total Energy Minimization. In this subsection, we compare PLAM and PCAL with the state-of-the-art solvers in solving Kohn-Sham total energy minimization (5.9) in serial. In other word, we aim to investigate the numerical performance of two proposed infeasible algorithms as general solvers for optimization problems with orthogonality constraints without consideration of parallelization.

Solver		Function value	KKT violation	Feasibility violation
$n = 1000, p = 20, \alpha = 1$				
PLAM	stop	-4.205530767124e+02	8.74e-06	2.56e-09
	orth	-4.205530767662e+02	8.74e-06	5.61e-15
PCAL	stop	-4.205530767773e+02	6.01e-06	1.13e-08
	orth	-4.205530767665e+02	6.00e-06	2.00e-14

TABLE 5.1
The results of orthogonal step for PLAM and PCAL on Problem 1

Our test is based on KSSOLV⁵ [35], which is a MATLAB toolbox for electronic structure calculation. It allows researchers to investigate their own algorithms easily and friendly for different steps in electronic structure calculation. We choose two integrated solvers in KSSOLV. One is the self-consistent field (SCF) iteration, which minimize a quadratic surrogate of the objective of (5.9) with orthogonality constraints in each iteration [21]. SCF and its variations are the most widely used in real KSDFT calculation. The other one is called trust-region direct constrained minimization (TRDCM) [37], which combines the trust-region framework and SCF to solve the subproblem. Besides SCF and TRDCM, which are particularly for KSDFT, we also pick up two state-of-the-art solvers in solving general optimization problems with orthogonality constraints. One is OptM⁶, which is based on the algorithm proposed in [33]. OptM adopts Cayley transform to preserve the feasibility on the Stiefel manifold in each iteration. Nonmonotone line search with BB stepsize is the default setting in OptM. Another existing solver for comparison, we intend to choose MOptQR, which is based on a projection-like retraction method introduced in [1]. Its original version is MOptQR-LS (manifold QR method with line search⁷). For fair comparison, we implement the same alternating BB stepsize strategy as PLAM and PCAL to MOptQR-LS, and form the MOptQR used in this section.

We select 18 testing problems with respect to different molecules, which are assembled in KSSOLV. For all the methods, the stopping criterion is set as $\|(I_n - XX^\top)H(X)X\|_F < 10^{-5}$. And we set the max iteration number $\text{MaxIter} = 200$ for methods SCF and TRDCM, while MOptQR, OptM, PLAM and PCAL set their max iteration number with $\text{MaxIter} = 1000$ to get a comparable solution with other methods. The penalty parameter β_{PLAM} for PLAM is tuned case by case to achieve a good performance. Meanwhile, β_{PCAL} for PCAL is always set as 1. Other parameters for all these methods take their default values. For all of the testing algorithms, we set the same initial guess X^0 by using the function “getX0”, which is provided by KSSOLV. The numerical results are illustrated in Tables 5.2, 5.3 and 5.4.

Here, “ E_{tot} ” represents the total energy function value, and “KKT violation”, “Iteration”, “Feasibility violation” and “Time(s)” stand for $\|(I_n - XX^\top)H(X)X\|_F$, the number of iteration, $\|X^\top X - I_p\|_F$ and the total running wall-clock time in second, respectively. From the tables, we observe that PCAL has a better performance than other algorithms, and in most cases, it obtains a comparable total energy function value and a lower KKT violation. In particular, in the large size problem “graphene30”, PCAL achieves the same total energy function value and same magnitude KKT violation in much less CPU time than others. In the problem “qdot”, we observe that only PLAM and PCAL can output a point satisfying the KKT violation tolerance, while all the other algorithms terminate abnormally. Therefore, we can conclude that PCAL and PLAM perform comparable with the existent feasible algorithms in solving discretized Kohn-Sham total energy minimization.

⁵Available from <http://crd-legacy.lbl.gov/~chao/KSSOLV/>

⁶Available from <http://optman.blogs.rice.edu>

⁷Available from <http://www.manopt.org>

Solver	E_{tot}	KKT violation	Iteration	Feasibility violation	Time(s)
al, $n = 16879, p = 12$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	-1.5789379003e+01	4.88e-03	200	6.53e-15	539.51
TRDCM	-1.5803791151e+01	6.36e-06	154	4.94e-15	336.79
MOptQR	-1.5803814080e+01	1.88e-04	1000	1.33e-14	393.54
OptM	-1.5803791098e+01	2.38e-05	1000	3.19e-14	378.80
PLAM	-1.5803790675e+01	1.29e-05	1000	3.34e-07	399.80
PCAL	-1.5803791055e+01	8.96e-06	596	5.95e-15	228.06
alanine, $n = 12671, p = 18$			$(\beta_{PLAM} = 13, \beta_{PCAL} = 1)$		
SCF	-6.1161921212e+01	3.80e-07	13	7.20e-15	21.46
TRDCM	-6.1161921213e+01	6.02e-06	15	5.20e-15	16.97
MOptQR	-6.1161921213e+01	7.52e-06	64	6.77e-15	14.89
OptM	-6.1161921213e+01	2.27e-06	69	4.03e-14	16.44
PLAM	-6.1161921212e+01	9.50e-06	76	7.90e-15	17.14
PCAL	-6.1161921213e+01	4.14e-06	61	7.19e-15	15.89
benzene, $n = 8407, p = 15$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	-3.7225751349e+01	2.10e-07	10	7.82e-15	10.07
TRDCM	-3.7225751363e+01	9.23e-06	15	7.12e-15	9.83
MOptQR	-3.7225751362e+01	8.12e-06	146	7.24e-15	19.91
OptM	-3.7225751363e+01	2.50e-06	70	1.54e-14	9.61
PLAM	-3.7225751362e+01	9.37e-06	71	4.62e-15	9.55
PCAL	-3.7225751362e+01	9.22e-06	50	5.15e-15	7.74
c2h6, $n = 2103, p = 7$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	-1.4420491315e+01	3.70e-09	10	3.66e-15	3.40
TRDCM	-1.4420491322e+01	8.75e-06	13	2.76e-15	4.01
MOptQR	-1.4420491321e+01	8.59e-06	47	2.58e-15	2.57
OptM	-1.4420491322e+01	2.62e-06	55	1.18e-14	2.87
PLAM	-1.4420491322e+01	7.91e-06	69	2.92e-15	3.41
PCAL	-1.4420491322e+01	4.91e-06	45	2.33e-15	2.58
c12h26, $n = 5709, p = 37$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	-8.1536091894e+01	4.95e-08	14	1.40e-14	30.08
TRDCM	-8.1536091937e+01	4.84e-06	16	1.17e-14	21.77
MOptQR	-8.1536091936e+01	6.68e-06	147	1.43e-14	39.57
OptM	-8.1536091937e+01	1.07e-06	83	7.10e-14	22.65
PLAM	-8.1536091936e+01	5.88e-06	96	1.55e-14	25.11
PCAL	-8.1536091936e+01	8.75e-06	70	1.45e-14	22.88
co2, $n = 2103, p = 8$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	-3.5124395789e+01	6.17e-08	10	2.53e-15	2.61
TRDCM	-3.5124395801e+01	4.14e-06	14	4.11e-15	2.09
MOptQR	-3.5124395800e+01	9.30e-06	88	2.35e-15	2.90
OptM	-3.5124395801e+01	1.70e-06	48	3.55e-14	1.68
PLAM	-3.5124395801e+01	7.92e-06	57	2.30e-15	1.84
PCAL	-3.5124395801e+01	9.15e-06	43	2.11e-15	1.74

TABLE 5.2
The results in Kohn-Sham total energy minimization

Solver	E_{tot}	KKT violation	Iteration	Feasibility violation	Time(s)
ctube661, $n = 12599, p = 48$ ($\beta_{PLAM} = 13, \beta_{PCAL} = 1$)					
SCF	-1.3463843175e+02	3.88e-07	11	1.43e-14	56.43
TRDCM	-1.3463843176e+02	6.85e-06	23	1.09e-14	87.41
MOptQR	-1.3463843176e+02	7.21e-06	152	1.78e-14	107.62
OptM	-1.3463843176e+02	2.35e-06	82	2.15e-14	59.23
PLAM	-1.3463843176e+02	4.34e-06	107	2.37e-14	72.18
PCAL	-1.3463843176e+02	9.68e-06	65	1.95e-14	54.07
glutamine, $n = 16517, p = 29$ ($\beta_{PLAM} = 13, \beta_{PCAL} = 1$)					
SCF	-9.1839425202e+01	1.12e-07	15	1.07e-14	67.40
TRDCM	-9.1839425244e+01	3.23e-06	16	7.00e-15	54.65
MOptQR	-9.1839425243e+01	9.83e-06	78	9.07e-15	51.46
OptM	-9.1839425244e+01	2.47e-06	87	9.73e-15	57.65
PLAM	-9.1839425243e+01	8.72e-06	104	9.26e-15	66.31
PCAL	-9.1839425243e+01	6.28e-06	74	9.33e-15	53.53
graphene16, $n = 3071, p = 37$ ($\beta_{PLAM} = 10, \beta_{PCAL} = 1$)					
SCF	-9.4023322108e+01	2.07e-03	200	1.32e-14	309.33
TRDCM	-9.4046217545e+01	8.85e-06	45	1.08e-14	47.87
MOptQR	-9.4046217225e+01	9.90e-06	422	1.15e-14	80.67
OptM	-9.4046217545e+01	2.27e-06	245	1.03e-14	48.66
PLAM	-9.4046217854e+01	9.52e-06	278	1.34e-14	51.57
PCAL	-9.4046217542e+01	8.68e-06	176	1.17e-14	41.11
graphene30, $n = 12279, p = 67$ ($\beta_{PLAM} = 13, \beta_{PCAL} = 1$)					
SCF	-1.7358453985e+02	5.19e-03	200	1.93e-14	2815.79
TRDCM	-1.7359510506e+02	4.80e-06	71	1.42e-14	765.92
MOptQR	-1.7359510505e+02	9.92e-06	456	2.59e-14	800.08
OptM	-1.7359510506e+02	2.47e-06	472	2.49e-14	904.44
PLAM	-1.7359510505e+02	8.88e-06	330	2.75e-14	601.41
PCAL	-1.7359510505e+02	8.52e-06	253	2.62e-14	548.70
h2o, $n = 2103, p = 4$ ($\beta_{PLAM} = 10, \beta_{PCAL} = 1$)					
SCF	-1.6440507245e+01	1.16e-08	8	1.15e-15	1.29
TRDCM	-1.6440507246e+01	6.48e-06	11	1.11e-15	1.02
MOptQR	-1.6440507246e+01	3.84e-06	49	9.30e-16	1.14
OptM	-1.6440507246e+01	2.01e-06	61	6.40e-15	1.50
PLAM	-1.6440507245e+01	6.43e-06	56	2.37e-15	1.29
PCAL	-1.6440507246e+01	7.42e-06	42	1.86e-15	1.06
hnco, $n = 2103, p = 8$ ($\beta_{PLAM} = 10, \beta_{PCAL} = 1$)					
SCF	-2.8634664360e+01	9.44e-08	12	3.82e-15	4.32
TRDCM	-2.8634664365e+01	9.54e-06	13	3.47e-15	4.47
MOptQR	-2.8634664363e+01	9.74e-06	163	3.17e-15	12.26
OptM	-2.8634664365e+01	5.30e-06	117	2.26e-15	8.30
PLAM	-2.8634664364e+01	9.95e-06	105	3.18e-15	7.39
PCAL	-2.8634664364e+01	9.03e-06	70	2.60e-15	5.36

TABLE 5.3
The results in Kohn-Sham total energy minimization

Solver	E_{tot}	KKT violation	Iteration	Feasibility violation	Time(s)
nic, $n = 251, p = 7$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	-2.3543529950e+01	2.13e-10	11	2.99e-15	1.47
TRDCM	-2.3543529955e+01	7.94e-06	15	4.49e-15	0.99
MOptQR	-2.3543529955e+01	3.04e-06	111	2.73e-15	1.53
OptM	-2.3543529955e+01	3.86e-07	63	8.80e-15	0.90
PLAM	-2.3543529955e+01	4.02e-06	67	1.39e-15	0.89
PCAL	-2.3543529955e+01	8.42e-06	52	1.88e-15	0.99
pentacene, $n = 44791, p = 51$			$(\beta_{PLAM} = 13, \beta_{PCAL} = 1)$		
SCF	-1.3189029494e+02	5.76e-07	13	1.58e-14	293.68
TRDCM	-1.3189029495e+02	7.60e-06	22	1.08e-14	276.25
MOptQR	-1.3189029495e+02	7.78e-06	112	3.21e-14	306.97
OptM	-1.3189029495e+02	1.39e-06	97	3.39e-14	283.02
PLAM	-1.3189029495e+02	8.66e-06	123	3.52e-14	321.04
PCAL	-1.3189029495e+02	7.67e-06	89	3.08e-14	271.32
ptnio, $n = 4069, p = 43$			$(\beta_{PLAM} = 13, \beta_{PCAL} = 1)$		
SCF	-2.2678884268e+02	1.09e-05	53	1.46e-14	168.25
TRDCM	-2.2678882693e+02	2.81e-04	200	1.07e-14	471.34
MOptQR	-2.2678884271e+02	9.57e-06	786	1.06e-14	347.38
OptM	-2.2678884273e+02	9.52e-06	508	1.14e-14	203.63
PLAM	-2.2678884271e+02	9.00e-06	579	1.01e-14	213.60
PCAL	-2.2678884271e+02	8.55e-06	386	1.19e-14	189.70
qdot, $n = 2103, p = 8$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	2.7700280133e+01	6.70e-03	5	2.92e-15	1.09
TRDCM	2.7699537080e+01	1.43e-02	200	2.73e-15	27.01
MOptQR	1.0483319768e+02	3.45e+01	1000	1.77e-15	28.72
OptM	2.7699807230e+01	1.45e-04	1000	2.39e-15	29.89
PLAM	2.7699800860e+01	9.68e-06	678	1.98e-15	19.30
PCAL	2.7699800851e+01	5.41e-06	962	2.88e-15	35.01
si2h4, $n = 2103, p = 6$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	-6.3009750375e+00	5.25e-07	11	3.62e-15	2.97
TRDCM	-6.3009750459e+00	8.24e-06	16	3.12e-15	4.30
MOptQR	-6.3009750460e+00	3.70e-06	116	2.00e-15	5.96
OptM	-6.3009750459e+00	9.60e-06	68	1.41e-14	4.15
PLAM	-6.3009750455e+00	7.27e-06	89	1.58e-15	5.33
PCAL	-6.3009750459e+00	4.33e-06	62	2.42e-15	3.90
sih4, $n = 2103, p = 4$			$(\beta_{PLAM} = 10, \beta_{PCAL} = 1)$		
SCF	-6.1769279820e+00	2.07e-08	8	1.75e-15	1.91
TRDCM	-6.1769279850e+00	9.53e-06	10	1.14e-15	1.60
MOptQR	-6.1769279851e+00	4.32e-06	34	1.58e-15	1.07
OptM	-6.1769279851e+00	8.18e-06	46	8.52e-16	1.62
PLAM	-6.1769279849e+00	7.37e-06	56	1.99e-15	1.79
PCAL	-6.1769279847e+00	9.16e-06	47	1.55e-15	1.69

TABLE 5.4
The results in Kohn-Sham total energy minimization

5.5. Parallel Efficiency. In this subsection, we examine the parallel efficiency of our algorithms PCAL and PCAL. To investigate the parallel scalability, we need to test large scale problems in a single core, which consumes lots of CPU time. To avoid meaningless tests, we only compare the parallel performances of PCAL with MOptQR in this subsection.

Both algorithms are implemented in the C++ language and parallelized by OpenMP. The linear algebra library we used in comparison is Eigen⁸ (version 3.3.4), which is an open and popular C++ template library for matrix computation. We define the speedup factor for running a code on m cores as

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

BLAS3 type arithmetic operations contribute a high proportion in computational cost in both PCAL and MOptQR. Therefore, a good parallel strategy for BLAS3 calculation is unnegligible in saving CPU time. Given this, we first determine the parallel strategy for matrix-matrix multiplication by a set of tests. We have two choices. The library Eigen provides its own multi-threading computing⁹ that is the default parallel strategy for dense matrix-matrix products and row-major-sparse*dense vector/matrix products in OpenMP. Another strategy is to parallelize BLAS3 computation in the manner of column-wise product. Namely, when we calculate AB , we multiply matrix A by each column of B in parallel. To figure out which strategy is better, we test the parallel scalability of BLAS3 computation under these two schemes. We generate $A = \text{Random}(1000, 10000)$ and $B = \text{Random}(10000, 1000)$, where “Random(\cdot, \cdot)” is an internal generation function provided by Eigen. We run the code in parallel with 1, 2, 4, 8, 16, 32, 64 and 96 cores, respectively. The result of matrix-matrix multiplication AB is illustrated in Figure 5.6. “Eigen” and “Column-wise” represent the default parallel strategy and column-wise product strategy, respectively. We can observe that column-wise parallelization obviously outperforms the default setting of Eigen in multi-threading computing. Hence, in the following implementation, we choose column-wise parallelization strategy for BLAS3 in our experiments.

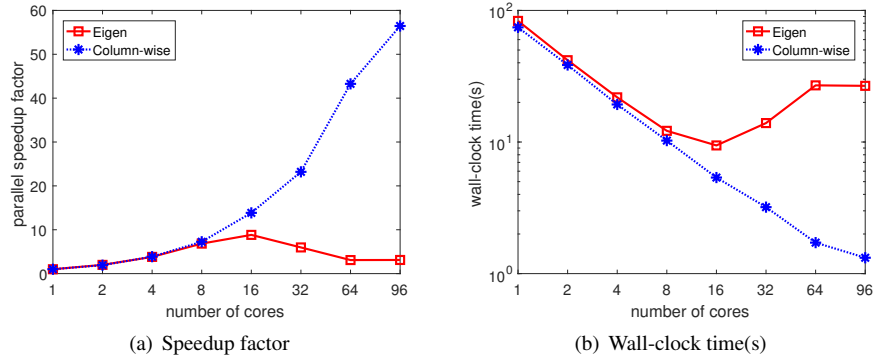


FIG. 5.6. The results of dense-dense BLAS3: $A^{1000 \times 10000} B^{10000 \times 1000}$

Next, we investigate the parallel scalability of the new proposed PCAL and MOptQR. According to the existent numerical report of Eigen¹⁰, we select the class “LLT” in Eigen to compute QR factorization. The calculation of orthonormalization consists of a small size (p -by- p) Cholesky decomposition and solving a p -by- p linear system. The maximum number of iterations for MOptQR and PCAL is set to 1000. All the parameters for MOptQR and PCAL take their default values. The initial guess X^0 is generated by $X^0 = \text{random}(n, p)$ and $X^0 = \text{qr}(X^0)$.

⁸Available from http://eigen.tuxfamily.org/index.php?title=Main_Page

⁹More information at <http://eigen.tuxfamily.org/dox/TopicMultiThreading.html>

¹⁰More information at http://eigen.tuxfamily.org/dox/group_TutorialLinearAlgebra.html

We first focus on the test Problems 1 and 2. For Problem 1, we set L as a block diagonal matrix, i.e., $L = \text{Diag}(L_1, \dots, L_s)$, where $L_i \in \mathbb{R}^{5 \times 5}$ is a tridiagonal matrix with 2 on its main diagonal and -1 on subdiagonal, for $i = 1, \dots, s$. The coefficient α is set to 1. For the generation of Problem 2, we set A as a tridiagonal matrix with 2 on its main diagonal and -1 on subdiagonal, and $G = \text{Random}(n, p)$. The advantage of such generation is to make function value and gradient calculations parallelizable. In the first group of tests, we aim to figure out how MOptQR and PCAL perform with the increasing width of variables. We set $n = 10000$ and p varying from a set of increasing values $\{500, 1000, 1500, 2000, 2500\}$. Both algorithms are run in parallel with 96 cores. The wall-clock time results are shown in Figure 5.7. Here, “#cores” stands for the number of cores. From Figure 5.7, we notice that PCAL always takes less amount of wall-clock time than MOptQR. As the width of the matrix variable increases, the running time of MOptQR increases much more rapidly than PCAL.

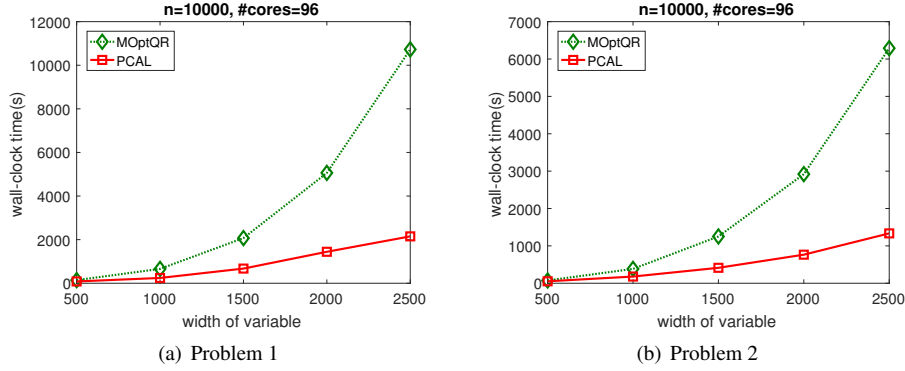


FIG. 5.7. The wall-clock time results on varying width of the matrix variable

In Figure 5.8, we show wall-clock time of four categories: three categories: “BLAS3” (dense-dense matrix multiplication), “Func” (function value and gradient evaluation) and “Orth” (orthonormalization including QR factorization for MOptQR and the final correction step in PCAL). These are the major computational components of both PCAL and MOptQR, albeit in different proportions. We have to clarify two issues: firstly, we categorize these categories of calculation only at the highest solver level. As such, any matrix-matrix multiplication involved in function value and gradient evaluation is not counted as in the “BLAS3” category. Secondly, although the correctness of such a classification scheme may be debatable, it does not alter the overall fact, as is clearly shown by our computational results, that the category “BLAS3” is much more scalable than the category “Orth” on our test platform. The running time of each category is measured in terms of the percentage of wall-clock time spent in that category over the total wall-clock time. We can clearly see that for PCAL the run time of “BLAS3” dominates the entire computation in almost all cases. The “BLAS3” time increases steadily as p increases from 500 to 2500, while the “Func” time decreases steadily. The run time of “Orth” is negligible. However, for MOptQR, the “BLAS3” time takes around 60% of total run time and decreases steadily with the increasing of p . Meanwhile, the “Orth” time takes around “40%” of total run time and increasing steadily.

Now, we set $n = 10000$ and $p = 1000, 2000$, and run PCAL and MOptQR in parallel with 1, 2, 4, 8, 16, 32, 64 and 96 cores, respectively. Figure 5.9 and 5.10 illustrate the speedup factors associated with total running wall-clock time, “BLAS3”, “Func” and “Orth”, respectively. From these two figures, we can observe that BLAS3 operation has high parallel scalability, while the speedup factor of “Orth” increases slowly as the number of cores increases, which directly leads to the higher overall scalability of PCAL than MOptQR. Moreover, as the width of the matrix variable increasing, the advantage of PCAL in parallel scalability becomes more obvious.

In the end, we test Problem 6 under $n = 10000$, $p = 1000$. Figure 5.11 illustrate the results of speedup factors

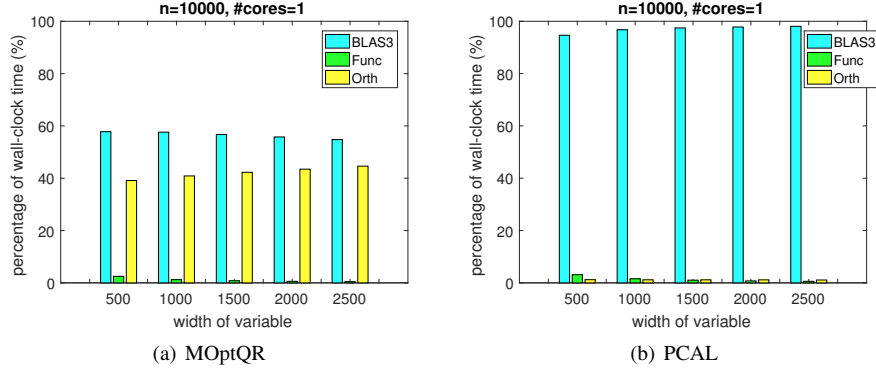


FIG. 5.8. A comparison of timing profile on a single core for Problem 2

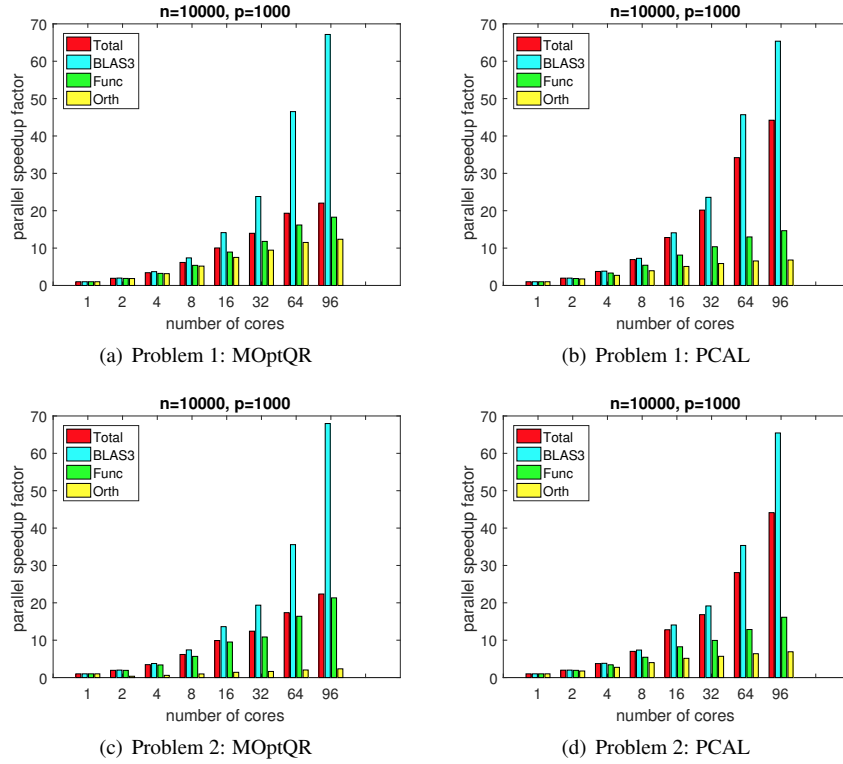


FIG. 5.9. A comparison of speedup factor among MOptQR and PCAL ($p = 1000$)

associated with total running wall-clock time, “BLAS3”, “Func” and “Orth” of PCAL and MOptQR, respectively. We can learn from this figure that the overall scalability of PCAL is again superior to that of MOptQR.

6. Conclusion. Optimization problems with orthogonality constraints have wide application in materials science, machine learning, image processing and so on. Particularly, when we apply Kohn-Sham density functional theory (KSDFT) to electronic structure calculation, the last step is to solve a Kohn-Sham total energy minimization with orthogonality constraints. There are plenty of existent algorithms based on manifold optimization, which work quite well when the number of columns of the matrix variable p is relatively few. With the increasing of p , a bottleneck of existent algorithms emerges, that is, lack of concurrency. The main reason leads to this bottleneck is that the orthonormalization process has low parallel scalability.

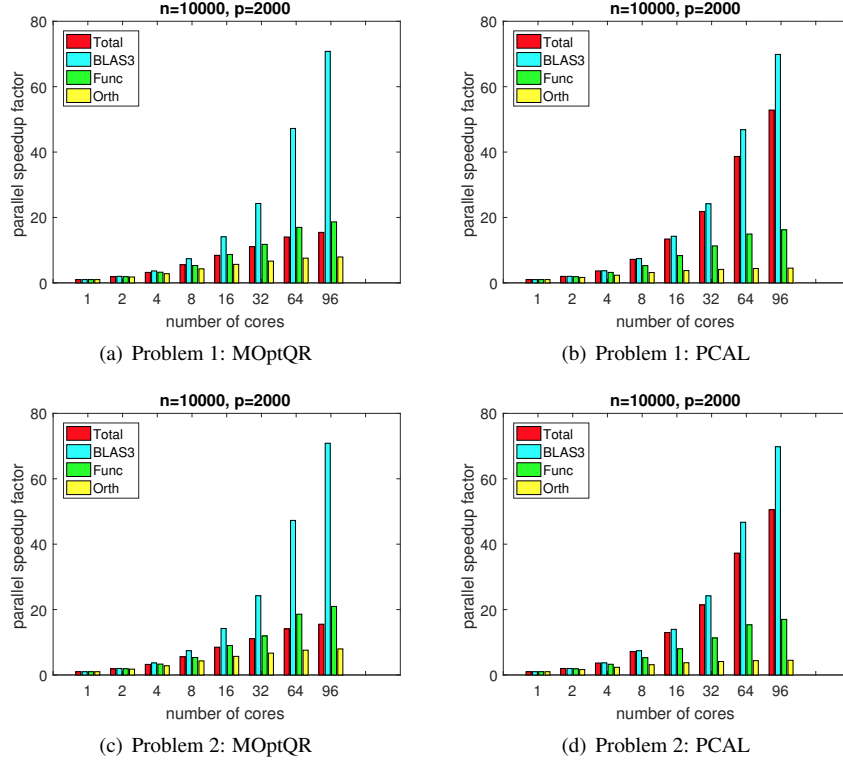


FIG. 5.10. A comparison of speedup factor among *MOptQR* and *PCAL* ($p = 2000$)

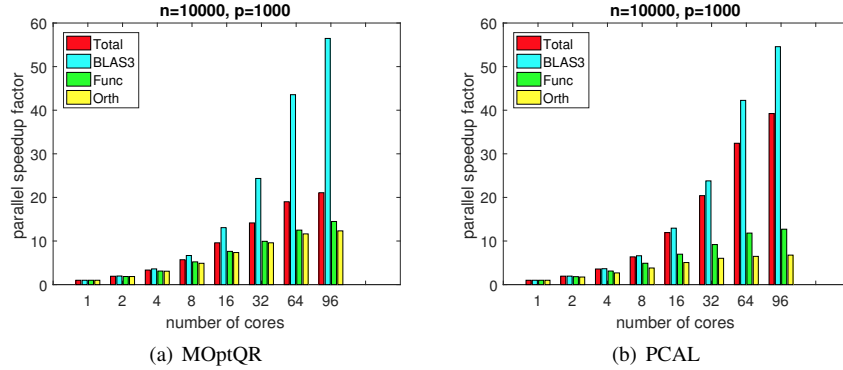


FIG. 5.11. A comparison of speedup factor among *MOptQR* and *PCAL* on the simplified Kohn-Sham total energy minimization

To solve this issue, we need to employ infeasible approaches. However, the existent infeasible approaches including augmented Lagrangian method (ALM) is far less efficient than the existent retraction based feasible methods. Even though the parallelization reduces the running time of ALM more significantly than that of manifold methods, ALM is still less efficient than manifold methods in parallel computing. The main purpose of this paper is to provide practical efficient infeasible algorithms for optimization problems with orthogonality constraints. Our main motivation is that the Lagrangian multipliers have closed-form expression at any stationary points. Hence, we use such expression to update multipliers instead to dual ascent step, at the same time, the subproblem for the prime variables only takes one gradient step instead of being solved to a given tolerance. The resultant algorithm, called PLAM, does not involve any orthonormalization. PLAM is comparable with the existent feasible algorithms under well chosen penalty parameter

β . To avoid such restriction, we propose a modified version, PCAL, of PLAM. The motivation of PCAL is to use normalized gradient step instead of gradient step in updating prime variables. The numerical experiments show that PCAL works efficient, robust and insensitive with penalty parameter β . Remarkably, it outperforms the existent feasible algorithms in solving the KSDFT problems in MATLAB platform KSSOLV. We also run PCAL and MOptQR, an excellent representative of retraction based optimization approach, in parallel with up to 96 cores. Numerical experiments illustrate PCAL has higher scalability than MOptQR, and its superiority becomes more and more noticeable with the increasing of p .

The potential of PCAL has already emerged. In the future work, we will apply our PCAL to real KSDFT calculation.

Acknowledgements. Xin Liu would like to thank Michael Overton for hosting his visit in Courant Institute of Mathematical Sciences, New York University. The authors would like to thank Tao Cui and Xingyu Gao for the insightful suggestions on designing the numerical experiments.

REFERENCES

- [1] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009.
- [2] J. BARZILAI AND J. M. BORWEIN, *Two-point step size gradient methods*, IMA journal of numerical analysis, 8 (1988), pp. 141–148.
- [3] D. P. BERTSEKAS, *Constrained optimization and Lagrange multiplier methods*, Academic press, 2014.
- [4] J. BOLTE, S. SABACH, AND M. TEBoulLE, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Mathematical Programming, 146 (2014), pp. 459–494.
- [5] N. BOUMAL, P.-A. ABSIL, AND C. CARTIS, *Global rates of convergence for nonconvex optimization on manifolds*, IMA Journal of Numerical Analysis, (2016).
- [6] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, J. ECKSTEIN, ET AL., *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends® in Machine learning, 3 (2011), pp. 1–122.
- [7] A. CASTRO, H. APPEL, M. OLIVEIRA, C. A. ROZZI, X. ANDRADE, F. LORENZEN, M. A. MARQUES, E. GROSS, AND A. RUBIO, *octopus: a tool for the application of time-dependent density functional theory*, physica status solidi (b), 243 (2006), pp. 2465–2488.
- [8] X. DAI, Z. LIU, L. ZHANG, AND A. ZHOU, *A conjugate gradient method for electronic structure calculations*, SIAM Journal on Scientific Computing, 39 (2017), pp. A2702–A2740.
- [9] Y.-H. DAI AND R. FLETCHER, *Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming*, Numerische Mathematik, 100 (2005), pp. 21–47.
- [10] A. EDELMAN, T. A. ARIAS, AND S. T. SMITH, *The geometry of algorithms with orthogonality constraints*, SIAM journal on Matrix Analysis and Applications, 20 (1998), pp. 303–353.
- [11] B. GAO, X. LIU, X. CHEN, AND Y.-X. YUAN, *A new first-order algorithmic framework for optimization problems with orthogonality constraints*, SIAM Journal on Optimization, 28 (2018), pp. 302–332.
- [12] W. GAO, C. YANG, AND J. C. MEZA, *Solving a class of nonlinear eigenvalue problems by Newton’s method*, Technical Reports, (2009).
- [13] P. GARCÍA-RISUEÑO, J. ALBERDI-RODRIGUEZ, M. J. OLIVEIRA, X. ANDRADE, M. PIPPIG, J. MUGUERZA, A. ARRUBARRENA, AND A. RUBIO, *A survey of the parallel performance and accuracy of Poisson solvers for electronic structure calculations*, Journal of computational chemistry, 35 (2014), pp. 427–444.
- [14] J. HU, A. MILZAREK, Z. WEN, AND Y. YUAN, *Adaptive quadratically regularized Newton method for Riemannian optimization*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), pp. 1181–1207.
- [15] B. JIANG AND Y.-H. DAI, *A framework of constraint preserving update schemes for optimization on Stiefel manifold*, Mathematical Programming, 153 (2015), pp. 535–575.
- [16] W. KOHN AND L. J. SHAM, *Self-consistent equations including exchange and correlation effects*, Physical review, 140 (1965), p. A1133.
- [17] R. LAI AND S. OSHER, *A splitting method for orthogonality constrained problems*, Journal of Scientific Computing, 58 (2014), pp. 431–449.
- [18] J. D. LEE, M. SIMCHOWITZ, M. I. JORDAN, AND B. RECHT, *Gradient descent only converges to minimizers*, in Conference on Learning Theory, 2016, pp. 1246–1257.
- [19] Y. LI, Z. WEN, C. YANG, AND Y. YUAN, *A semi-smooth Newton method for solving semidefinite programs in electronic structure calculations*, arXiv:1708.08048, (2017).
- [20] J. LIU, S. J. WRIGHT, C. RÉ, V. BITTORF, AND S. SRIDHAR, *An asynchronous parallel stochastic coordinate descent algorithm*, The Journal of Machine Learning Research, 16 (2015), pp. 285–322.

- [21] X. LIU, X. WANG, Z. WEN, AND Y. YUAN, *On the convergence of the self-consistent field iteration in Kohn–Sham density functional theory*, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 546–558.
- [22] X. LIU, Z. WEN, X. WANG, M. ULBRICH, AND Y. YUAN, *On the analysis of the discretized Kohn–Sham density functional theory*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 1758–1785.
- [23] X. LIU, Z. WEN, AND Y. ZHANG, *An efficient Gauss–Newton algorithm for symmetric low-rank product matrix approximations*, SIAM Journal on Optimization, 25 (2015), pp. 1571–1608.
- [24] M. A. MARQUES, A. CASTRO, G. F. BERTSCH, AND A. RUBIO, *octopus: a first-principles tool for excited electron–ion dynamics*, Computer Physics Communications, 151 (2003), pp. 60–78.
- [25] Y. NISHIMORI AND S. AKAHO, *Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold*, Neurocomputing, 67 (2005), pp. 106–135.
- [26] J. NOCEDAL AND S. J. WRIGHT, *Numerical optimization*, 2nd, Springer, 2006.
- [27] Z. PENG, Y. XU, M. YAN, AND W. YIN, *Arock: an algorithmic framework for asynchronous parallel coordinate updates*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2851–A2879.
- [28] Z. PENG, M. YAN, AND W. YIN, *Parallel and distributed sparse optimization*, in Signals, Systems and Computers, 2013 Asilomar Conference on, IEEE, 2013, pp. 659–646.
- [29] B. RECHT, C. RE, S. WRIGHT, AND F. NIU, *Hogwild: A lock-free approach to parallelizing stochastic gradient descent*, in Advances in neural information processing systems, 2011, pp. 693–701.
- [30] M. ULBRICH, Z. WEN, C. YANG, D. KLOCKNER, AND Z. LU, *A proximal gradient method for ensemble density functional theory*, SIAM Journal on Scientific Computing, 37 (2015), pp. A1975–A2002.
- [31] Z. WEN, A. MILZAREK, M. ULBRICH, AND H. ZHANG, *Adaptive regularized self-consistent field iteration with exact Hessian for electronic structure calculation*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1299–A1324.
- [32] Z. WEN, C. YANG, X. LIU, AND Y. ZHANG, *Trace-penalty minimization for large-scale eigenspace computation*, Journal of Scientific Computing, 66 (2016), pp. 1175–1203.
- [33] Z. WEN AND W. YIN, *A feasible method for optimization with orthogonality constraints*, Mathematical Programming, 142 (2013), pp. 397–434.
- [34] C. YANG, W. GAO, AND J. C. MEZA, *On the convergence of the self-consistent field iteration for a class of nonlinear eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, 30 (2009), pp. 1773–1788.
- [35] C. YANG, J. C. MEZA, B. LEE, AND L.-W. WANG, *Kssolv—a MATLAB toolbox for solving the Kohn–Sham equations*, ACM Transactions on Mathematical Software (TOMS), 36 (2009), p. 10.
- [36] C. YANG, J. C. MEZA, AND L.-W. WANG, *A constrained optimization algorithm for total energy minimization in electronic structure calculations*, Journal of Computational Physics, 217 (2006), pp. 709–721.
- [37] C. YANG, J. C. MEZA, AND L.-W. WANG, *A trust region direct constrained minimization algorithm for the Kohn–Sham equation*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1854–1875.
- [38] X. ZHANG, J. ZHU, Z. WEN, AND A. ZHOU, *Gradient type optimization methods for electronic structure calculations*, SIAM Journal on Scientific Computing, 36 (2014), pp. C265–C289.