SIAM J. Sci. Comput.
Vol. 41, No. 3, pp. A1949–A1983

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 a 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 columnwise 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

DOI. 10.1137/18M1221679

1. Introduction.

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

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

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

Throughout this paper, we assume the following.

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.

[*]Submitted to the journal's Methods and Algorithms for Scientific Computing section October 19, 2018; accepted for publication (in revised form) March 25, 2019; published electronically June 20, 2019.

http://www.siam.org/journals/sisc/41-3/M122167.html

Funding: The work of the second author was supported by the NSFC through grants 11622112, 11471325, 91530204, and 11688101 and by the National Center for Mathematics and Interdisciplinary Sciences, CAS, and Key Research Program of Frontier Sciences QYZDJ-SSW-SYS010, CAS. The work of the third author was supported by NSFC grants 11331012 and 11461161005.

[†]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, liuxin@lsec.cc.ac.cn, yyx@lsec.cc.ac.cn).

1.1. Literature survey. Kohn–Sham density functional theory (KSDFT) is known to be an important topic in materials science [12]. The last step of KSDFT is to minimize a discretized Kohn–Sham total energy function,

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

subject to orthogonality constraints. Here $\rho(X) := \mathrm{diag}(XX^\top)$ 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 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 pseudoinverse of L. The exchange correlation function ϵ_{xc} is used to model the nonclassical 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 for efficient solvers for optimization with orthogonality constraints is high.

In recent decades, researchers have proposed quite a few efficient optimization approaches for discretized Kohn–Sham total energy minimization [33, 34, 32, 31, 28, 35, 27, 6, 14]. For the general purpose of solving optimization problems with orthogonality constrains, there are abundant algorithms: retraction based approaches [9, 20, 1, 30, 11], splitting algorithm [13], multipliers correction framework [10], just to mention a few. Interested readers are referred to the references in [10]. There are a few successful solvers. The most famous one is the toolbox for optimization on manifolds, which is called Manopt,[1] 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 OptM[2] 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. Unfortunately, parallel computation has not attracted much attention from the optimization area until very recently. Refer to [26, 5, 23, 15, 22]; there is an urgent demand for parallelization in the optimization area. Although high scalability algorithms have been desired in the KSDFT area for decades, there has been no successful attempt in this regard so far [6].

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

1.2. Contribution. In this paper, we propose an infeasible algorithm for optimization problems with orthogonality constraints. It is based on the augmented Lagrangian method (ALM) but employs a totally different updating scheme for both prime and dual variables. The main motivation of the so-called proximal linearized ALM (PLAM) is an observation that the dual variables enjoy a closed-form formula at each first-order stationary point. Therefore, we intend to use the symmetrization of this formula as the updating rule for the dual variables, to replace the dual ascent (DA) step in the classical ALM. 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 taking one gradient descent step.

[1] Available from http://www.manopt.org.

[2] Available from http://optman.blogs.rice.edu.

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 a 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 columnwise 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.

The numerical experiments under serial computing demonstrate 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 will investigate 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 pseudoinverse of X. $\mathrm{Diag}(v) \in \mathbb{S}^n$ denotes a diagonal matrix with all entries of $v \in \mathbb{R}^n$ in its diagonal, and $\mathrm{diag}(A) \in \mathbb{R}^n$ extracts the diagonal entries of matrix $A \in \mathbb{R}^{n\times n}$. For convenience, $\Phi(M) := \mathrm{Diag}(\mathrm{diag}(M))$ represents the diagonal matrix with the diagonal entries of square matrix M in its diagonal. $\mathbb{D}^p := \{X \in \mathbb{R}^{p\times p} \mid \Phi(X) = X\}$ refers to the p-by-p real symmetric matrices set. $\Psi(A) := \frac{1}{2}(A + A^\top)$ stands for the average of a square matrix and its transpose.

2. Motivation.

As mentioned in the previous section, almost all the existing practically useful methods require feasible iterates all the time. To realize feasibility, either explicit or implicit orthonormalization must be invoked. This kind of calculation lacks 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 the 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 using 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. Previous infeasible approaches designed for (1.1) either work specially for Rayleigh–Ritz trace minimization [18, 29] or adopt an alternating direction method of multipliers (ADMM) framework after introducing auxiliary variables to split the objective and orthogonality constraints [13]. The previous ones can hardly be extended to general objective, while the latter one does not always perform well for general purpose,[3] although it is practically useful in many applications.

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 the following.

Definition 2.1. *Given a point $X \in \mathbb{R}^{n\times p}$, if the relationship*

$$\begin{cases} \mathrm{tr}(Y^\top \nabla f(X)) & \geq & 0; \\ X^\top X & = & I_p \end{cases}$$

holds for any $Y \in \mathcal{T}(X)$, we call X a first-order stationary point of (1.1). *Here, $\mathcal{T}(X) := \{Y \mid Y^\top X + X^\top Y = 0\}$ is the tangent space of the orthogonality constraints at X.*

According to Lemma 2.2 in [10], a point X is a first-order stationary point if and only if

$$\begin{cases} (I_n - XX^\top)\nabla f(X) & = & 0; \\ X^\top \nabla f(X) & = & \nabla f(X)^\top X; \\ X^\top X & = & I_p. \end{cases} \tag{2.1}$$

Due to the fact that

$$||\nabla f(X) - X\nabla f(X)^\top X||_{\mathrm{F}}^2 = ||\nabla f(X) - XX^\top \nabla f(X)||_{\mathrm{F}}^2 + ||X^\top \nabla f(X) - \nabla f(X)^\top X||_{\mathrm{F}}^2,$$

we can obtain that condition (2.1) is equivalent to

$$\begin{cases} \nabla f(X) = X\Lambda, \ \text{with}\, \Lambda = \nabla f(X)^\top X; \\ X^\top X = I_p. \end{cases} \tag{2.2}$$

Here, $\Lambda \in \mathbb{S}^p$ can be viewed as the Lagrangian multipliers of the orthogonality constraints.

Definition 2.2. *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*

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

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

[3] Numerical evidence can be found in Appendix A.

PROPOSITION 2.3. *If X is a local minimizer of* (1.1)*, it has to be a second-order stationary point. X is a strict local minimizer*[4] *if and only if X is a first-order stationary point and satisfies*

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

2.2. Augmented Lagrangian method. A straightforward idea to solve (1.1) without requiring feasibility in each iteration is to employ the ALM [25, 21, 3], which is described in Algorithm 1.

Algorithm 1. Augmented Lagrangian Method (ALM).

```
Input: choose initial guess $\Lambda^0$ for the dual variables, and set $k := 0$;
while certain stopping criterion is not reached do
    Minimize the augmented Lagrangian function with respect to the prime variables $X$:
        $X^{k+1} := \min_{X \in \mathbb{R}^{n\times p}} \quad \mathcal{L}_\beta(X, \Lambda^k)$,
    where the augmented Lagrangian function of problem (1.1) is defined as
        $\mathcal{L}_\beta(X,\Lambda) = f(X) - \frac{1}{2}\langle \Lambda, X^\top X - I_p\rangle + \frac{\beta}{4}||X^\top X - I_p||_{\mathrm{F}}^2$
  (2.5)     $= f(X) + \frac{\beta}{4}\left\| X^\top X - \left(I_p + \frac{1}{\beta}\Lambda\right)\right\|_{\mathrm{F}}^2 - \frac{1}{4\beta}||\Lambda||_{\mathrm{F}}^2$.
    Update the Lagrangian multipliers
  (2.6)     $\Lambda^{k+1} := \Lambda^k - \beta({X^{k+1}}^\top X^{k+1} - I_p)$.
    Update the penalty parameter $\beta$ if necessary. Set $k := k + 1$.
Output: $X^k$.
```

It is well-known that the augmented Lagrangian function is an exact penalty if the Lagrangian multipliers are correct and the penalty parameter β is sufficiently large. Algorithm 1 works very well for a 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.

Therefore, we need to employ a new idea to remold the classical ALM. According to the conditions (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:

$$\Lambda = \nabla f(X)^\top X. \tag{2.7}$$

[4] X is called a strict local minimizer if $X \in \mathcal{S}_{n,p}$ and there exists $\delta > 0$ such that $f(X) < f(Y)$ holds for any $Y \in U_\delta(X) := \{Y \in \mathcal{S}_{n,p} \mid ||X - Y|| \in (0, \delta)\}$.

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

$$\Lambda = \Psi(\nabla f(X)^\top X), \tag{2.8}$$

as a new multipliers updating rule. The symmetrization is necessary because the symmetry of the expression $\nabla f(X)^\top X$ cannot 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 2.4. *Let X^* be a second-order stationary point of*

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

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* (2.2) *hold at X^*.*

Proof. First, we have

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

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

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

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

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

Substituting (2.10) into (2.12), we obtain

$$\nabla f(X^*) - X^*\Lambda^* - \beta X^*(I_p - {X^*}^\top X^*) = 0. \tag{2.14}$$

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

$${X^*}^\top \nabla f(X^*) = {X^*}^\top X^* \Lambda^* + \beta {X^*}^\top X^*(I_p - {X^*}^\top X^*). \tag{2.15}$$

Suppose $X^* = U\Sigma V^\top$ is the singular value decomposition of X^* with $U \in \mathcal{S}_{n,p}$, $\Sigma \in \mathbb{D}^p$, and $V \in \mathcal{S}_{p,p}$, 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

$$\mathrm{diag}(V^\top {X^*}^\top \nabla f(X^*) V) = \mathrm{diag}(V^\top \nabla f(X^*)^\top X^* V) = \mathrm{diag}(V^\top \Lambda^* V), \tag{2.16}$$

we have

$$(I_p - \Sigma^2)(\Phi(V^\top \Lambda^* V) - \beta\Sigma^2) = 0, \tag{2.17}$$

which implies that

$$D(\Phi(V^\top \Lambda^* V) - \beta\Sigma^2) = 0, \tag{2.18}$$

where matrix $D \in \mathbb{D}^p$ satisfies

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

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 S into (2.11) and obtain

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

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

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

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 (2.2) hold. This completes the proof. □

Lemma 2.4 guarantees that the 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 2.4. Moreover, to obtain the global convergence rate, the feasibility should be controlled by the first-order optimality violation.

Lemma 2.5. *For any X^* satisfying $\sigma_{\min}(X^*) > 0$, suppose*

$$\beta > \left(||\nabla f(X^*)||_2 \cdot ||X^*||_2 + \delta\right) / \sigma^2_{\min}(X^*)$$

with $\delta > 0$. Then it holds that

$$||{X^*}^\top X^* - I_p||_{\mathrm{F}} \leq \frac{||X^*||_2}{\delta} \cdot ||\nabla_X \mathcal{L}_\beta(X^*, \Lambda^*)||_{\mathrm{F}} \tag{2.20}$$

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 also 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

$$ V^\top {X^*}^\top G V = V^\top {X^*}^\top \nabla f(X^*) V - \beta\Sigma^2 - \Sigma^2 \left(V^\top \Lambda^* V - \beta\Sigma^2 \right). $$

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

$$ \Phi(V^\top {X^*}^\top G V) = (I_p - \Sigma^2)(\Phi(V^\top \Lambda^* V) - \beta\Sigma^2). \tag{2.21} $$

Since $\beta > \left(||\nabla f(X^*)||_\mathrm{F} \cdot ||X^*||_2 + \delta \right) / \sigma^2_{\min}(X^*)$, we have

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

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

$$ \sigma_{\min}\left(\beta\Sigma^2 - \Phi(V^\top \Lambda^* V) \right) \geq \delta. \tag{2.22} $$

Submitting (2.22) into (2.21), we arrive at

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

and complete the proof. ∎

3. Parallelizable algorithms.

In this section, we introduce a parallelizable approach and one of its variants for optimization problems 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 DA 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 a proximal linearized approximation [4].

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 steps 3 and 4. Step 3 only involves BLAS3 calculation. The minimization subproblem (3.2) in step 4 is nothing but a gradient step,

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

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 the 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 detail in section 5.

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
5
          (3.2)
          $X^{k+1} := \arg\min_{X \in \mathbb{R}^{n\times p}} \tilde{\mathcal{L}}_\beta(X) = \mathrm{tr}(\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)^\top (X - X^k)) + \frac{\eta^k}{2} ||X - X^k||_{\mathrm{F}}^2.$
6     Set $k := k + 1$.
7 Output: $X^k$.
```

3.2. Parallelizable columnwise 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 columnwise unit sphere constraints are imposed on step 4. Therefore, the proximal gradient takes the place of the gradient step in 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 columnwise 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 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 (2.2), we impose an additional term for the redundant sphere constraints. Namely,

$$\left\{ \begin{array}{c} \nabla f(X) = X\Lambda + XD; \\ X^\top X = I_p, \end{array} \right. \tag{3.6}$$

where $D \in \mathbb{D}^p$. Furthermore, D is determined by the Lagrangian multiplier of X_i in the subproblem (3.5).

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

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

3.3. Computational cost. In this subsection, we compare the computational cost per iteration among MOptQR, 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 of 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

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

Remark 4.2. Assumption 4.1 is not restrictive at all. There are two types of points, which can be generated easily, satisfying this assumption:

Case 1: $X^0 = Q\Sigma$, where

$$Q \in \mathcal{S}_{n,p}, \Sigma = \mathrm{Diag}(\underbrace{1, ..., 1}_{p-1}, \underline{\sigma}),$$

for any given $\underline{\sigma} \in (0,1)$.

TABLE 3.1
The comparison of computational cost.

Evaluate function				
$f(X) := \frac{1}{2}\mathrm{tr}(X^\top AX) + \mathrm{tr}(G^\top X)$	AX	A: dense	A: sparse	A: sparse
		$2n^2p$	$O(np)$	$O(np)$
	$\nabla f(X) = AX + G$	np		
	$\frac{1}{2}\mathrm{tr}(X^\top AX) + \mathrm{tr}(G^\top X)$	$4np$		
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: $X^\top X - I$				
$X^\top X$	np^2			$np^2 + np$
Solvers				
PLAM	$X(X^\top X - I)$	$2np^2$		$4np^2 + O(np)$
	$X\Psi(\nabla f(X)^\top X)$	$2np^2$		
PCAL	$X(X^\top X - I)$	$2np^2$		$4np^2 + O(np)$
	$X\Psi(\nabla f(X)^\top X)$	$2np^2$		
	$\Phi\left(X^{k\top}\nabla_X L_\beta(X^k, \Psi(\nabla f(X^k)^\top X^k))\right)$	$O(np)$		
	$X\Lambda = X\Psi(\cdot) + X\Phi(\cdot)$	$O(np)$		
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		
	$\mathrm{chol}(V^\top V) = LL^\top$	$p^3/3$		
	$VL^{-\top}$	$2np^2 + O(p^3)$		
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			

It can be verified that $\sigma_{\min}(X^0) = \underline{\sigma}$ and $||X^{0\top}X^0 - I_p||_{\mathrm{F}} = 1 - \underline{\sigma}^2 > 0$ in this case.

Case 2: $X^0 \notin \mathcal{S}_{n,p}$ satisfying $\sigma^2_{\min}(X^0) > 1 - \frac{1}{\sqrt{p}}$ and $\sigma^2_{\max}(X^0) < 1 + \frac{1}{\sqrt{p}}$. In this case, $0 < ||X^{0\top}X^0 - I_p||_{\mathrm{F}}$ holds immediately. Let

$$\underline{\sigma} = \sqrt{\min\left\{1 - \frac{1}{\sqrt{p}},\ \sqrt{p}\left(1 + \frac{1}{\sqrt{p}} - \sigma^2_{\max}(X^0)\right),\ \sqrt{p}\left(\sigma^2_{\min}(X^0) - 1 + \frac{1}{\sqrt{p}}\right)\right\}};$$

then it is not difficult to deduce that $\sigma_{\min}(X^0) > \underline{\sigma} > 0$. Moreover, we have

$$\begin{aligned}&||X^{0\top}X^0 - I_p||_{\mathrm{F}}\\ &\leq \sqrt{p}\cdot||X^{0\top}X^0 - I_p||_2 \leq \sqrt{p}\cdot\sqrt{\max\{\lambda^2_{\max}(X^{0\top}X^0 - I_p),\ \lambda^2_{\min}(X^{0\top}X^0 - I_p)\}}.\end{aligned}$$

Since

$$\begin{aligned}\lambda_{\max}(X^{0\top}X^0 - I) = \lambda_{\max}(X^{0\top}X^0) - 1 = \sigma^2_{\max}(X^0) - 1 \leq \frac{1}{\sqrt{p}} - \frac{\underline{\sigma}^2}{\sqrt{p}},\\ \lambda_{\min}(X^{0\top}X^0 - I) = \lambda_{\min}(X^{0\top}X^0) - 1 = \sigma^2_{\min}(X^0) - 1 \geq \frac{\underline{\sigma}^2}{\sqrt{p}} - \frac{1}{\sqrt{p}},\end{aligned}$$

we obtain $||X^{0\top}X^0 - I_p||_{\mathrm{F}} \leq \sqrt{p}\cdot\left(\frac{1}{\sqrt{p}} - \frac{\underline{\sigma}^2}{\sqrt{p}}\right) = 1 - \underline{\sigma}^2$.

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

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

We introduce the following merit function:

$$h(X) = f(X) - \frac{1}{2}\left\langle \Psi(\nabla f(X)^\top X), X^\top X - I_p \right\rangle + \frac{\beta}{4}||X^\top X - I_p||_{\mathrm{F}}^2. \tag{4.2}$$

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

$$||\nabla h(X) - \nabla h(y)||_{\mathrm{F}} \le L_h ||X - Y||_{\mathrm{F}} \quad \forall\, X, Y \in \mathcal{C}. \tag{4.3}$$

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

Assumption 4.3.

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

$$\begin{aligned} &c_2 \in \left(0, \frac{R^2(\beta\underline{\sigma}^2 - 4MN)}{2N_L^2}\right]; \quad \eta^k \in \left[\underline{\eta}, \bar{\eta}\right], \\ &\text{where } \underline{\eta} = \max\left\{ \frac{L_h}{2c_1}, \frac{2N_L M + N_L\sqrt{4M^2 + 2R}}{R}, \frac{R + 2M^2}{c_2} \right\}, \\ &N_L = (1 + M^2)N + \beta R M, \quad \bar{\eta} \ge \underline{\eta}. \end{aligned} \tag{4.5}$$

Remark 4.4. The conditions in Assumption 4.3 are introduced for theoretical analysis. Parameters β and η^k satisfying these conditions are usually restrictive in practical use.

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

(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 is convergent;
(4) Any cluster point of $\{X^k\}$, say, X^*, is a first-order stationary point of 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 4.5. *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* 4.3*. Then it holds that*

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

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, $\boldsymbol{||X^{k^\top}X^k - I_p||_{\mathrm{F}} \leq \frac{R}{2}}$. We have

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

It is not difficult to verify that

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

holds for any $X^k \in \mathcal{C}$. By using the facts $X^k \in \mathcal{C}$, (4.1), and (4.5), we have

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

which implies $||X^{k+1^\top}X^{k+1} - I_p||_{\mathrm{F}} \leq R$. This shows (4.6) is true for $k+1$.

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

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

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

$$\|\delta\|_{\mathrm{F}} > \frac{R\underline{\sigma}}{2}. \tag{4.8}$$

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

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

Hence, we have

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

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

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

$$\geq \frac{\beta R^2 \underline{\sigma}^2}{2} - 2R^2 MN - c_2 N_L^2 \geq 0.$$

According to the Taylor expansion, we have

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

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. □

LEMMA 4.6. *$h(X)$ defined by (4.2) is bounded below at $\mathcal{C}$.*

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

LEMMA 4.7. *Suppose $\{X^k\}$ is the iterate sequence generated by Algorithm 2 initiated from X^0 satisfying Assumption 4.1, the problem parameters satisfy Assumption 4.3, and $h(X)$ is defined by (4.2). Then it holds that*

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

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

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

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

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

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

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

$$\leq h(X^k) - \left(\frac{c_1}{\bar{\eta}} - \frac{L_h}{2\bar{\eta}^2}\right) \cdot ||\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k)||_{\mathrm{F}}^2.$$

We complete the proof. □

With the boundedness of $h(X)$ at $\mathcal{C}$, Lemma 4.7 immediately implies the convergence of $\{h(X^k)\}$. More precisely, we have the following corollary.

COROLLARY 4.8. *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* 4.3*. Then the algorithm is finitely terminated at kth iteration with $\nabla_X \mathcal{L}_\beta(X^k, \Lambda^k) = 0$, or*

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

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

Proof. This is a direct corollary of Lemmas 4.5 and 4.7. □

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

THEOREM 4.9. *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* 4.3*. Then the sequence $\{X^k\}$ has at least one cluster point, and any cluster point is a first-order stationary point of problem* (1.1)*. More precisely, for any $K > 1$, it holds that*

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

Proof. The first part directly holds from Corollary 4.8 and Lemma 2.5. Recalling Lemma 4.7, we have

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

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

Moreover, we have

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

Combining (4.13)–(4.15), we arrive at the argument (4.12). □

COROLLARY 4.10. *Suppose all the assumptions of Theorem* 4.9 *hold. Besides, for a given positive parameter δ, it holds that $\beta > (MN + \delta)/\underline{\sigma}$; then it holds that*

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

Proof. This is a direct corollary of Lemma 2.5 and Theorem 4.9. □

Remark 4.11. The sublinear convergence rate of Corollary 4.10 actually tells us that Algorithm 2 terminates after $O(1/\epsilon^2)$ iterations if the stopping criterion is set as $\max\{||I_p - {X^k}^\top X^k||_{\mathrm{F}}, ||\nabla_X \mathcal{L}_\beta(X^k,\Lambda^k)||_{\mathrm{F}}\} < \epsilon$.

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

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

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 $||X^0 - X^*||_{\mathrm{F}} < \varepsilon$, 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

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

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

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

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

$$\delta^k = X^*S + X^*W + K, \tag{4.18}$$

where $S \in \mathbb{R}^{p\times p}$ is symmetric, $W \in \mathbb{R}^{p\times p}$ is skew-symmetric, and $K \in \mathbb{R}^{n\times p}$ 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

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

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

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

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

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

Notice that (4.17) implies the positive semidefiniteness of the linear operator

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

Together with (4.21), we can conclude that

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

which completes the proof. □

Remark 4.13. 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 the MATLAB toolbox KSSOLV [32]. All the algorithms tested in the first two parts are run in serial. The corresponding experiments are performed on a workstation with one Intel Xeon Processor E5-2697 v2 (at 2.70GHz×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-IV,[5] which is a high-performance computing cluster maintained at the State Key Laboratory of Scientific and Engineering Computing, 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 Xeon 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 4.9, the penalty parameter β for PLAM should be sufficiently large. Although we can estimate a suitable β to satisfy the

[5] More information at http://lsec.cc.ac.cn/chinese/lsec/LSSC-IVintroduction.pdf.

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 stepsize of the gradient step in Algorithms 2 and 3. Similar to β, we cannot use the rigorous restriction in the theoretical analysis. In practice, we have the following strategies to choose this parameter:

(i) $\eta_{\mathrm{C}}^k := \gamma$, where $\gamma > 0$ is a sufficiently large constant.

(ii) Differential approximation:

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

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

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

where

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

(iv) Alternating BB strategy [7]:

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

Unless specifically mentioned, the stopping criterion used for both serial and parallel experiments can be described as follows:

$$\frac{\left\|\nabla f(X) - X\nabla f(X)^\top X\right\|_{\mathrm{F}}}{\left\|\nabla f(X^0) - X^0\nabla f(X^0)^\top X^0\right\|_{\mathrm{F}}} < 10^{-8}.$$

The maximum number of iterations for all those solvers is set to 3000.

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

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

$$\begin{array}{cc} \min\limits_{X\in\mathbb{R}^{n\times p}} & \frac{1}{2}\mathrm{tr}(X^\top LX) + \frac{\alpha}{4}\rho(X)^\top L^\dagger \rho(X) \\ \text{s. t.} & X^\top X = I_p, \end{array} \tag{5.1}$$

where the matrix $L \in \mathbb{S}^n$ and $\rho(X) := \mathrm{diag}(XX^\top)$. In the numerical experiments, we set $\alpha = 1$, and L is randomly generated by Gauss distribution, i.e., `L=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.

$$\begin{array}{cc} \min\limits_{X\in\mathbb{R}^{n\times p}} & \frac{1}{2}\mathrm{tr}(X^\top AX) + \mathrm{tr}(G^\top X) \\ \text{s.t.} & X^\top X = I_p, \end{array} \tag{5.2}$$

where the matrices $A \in \mathbb{S}^n$ and $G \in \mathbb{R}^{n\times p}$. This problem is adequately discussed in [10]. In the numerical experiments, the matrices A and G are randomly generated in the same manner as in [10]. Namely,

$$A := P\Lambda P^\top, \tag{5.3}$$

$$G := \kappa \cdot QD, \tag{5.4}$$

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

$$\Lambda_{ii} := \left\{ \begin{array}{rl} \theta^{1-i}, & \mathbf{rand}(1,1) < \xi \\ -\theta^{1-i}, & \mathbf{rand}(1,1) \geq \xi \end{array} \right. \quad \forall i = 1, 2, \ldots, n, \tag{5.5}$$

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

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 settings 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.

$$\begin{array}{cc} \min\limits_{X\in\mathbb{R}^{n\times p}} & \frac{1}{2}\mathrm{tr}(X^\top AX) \\ \text{s.t.} & X^\top X = I_p, \end{array} \tag{5.7}$$

where the matrix $A \in \mathbb{S}^n$. 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.

$$\begin{array}{cc} \min\limits_{X\in\mathbb{R}^{n\times p}} & \frac{1}{2}\mathrm{tr}(A^\top XBX^\top) \\ \text{s.t.} & X^\top X = I_p, \end{array} \tag{5.8}$$

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 [10] but can be solved by PLAM or PCAL. The matrices A and B are randomly generated by A =`randn(n)`, $A := \frac{1}{2}(A + A^\top)$ and B =`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 [32].

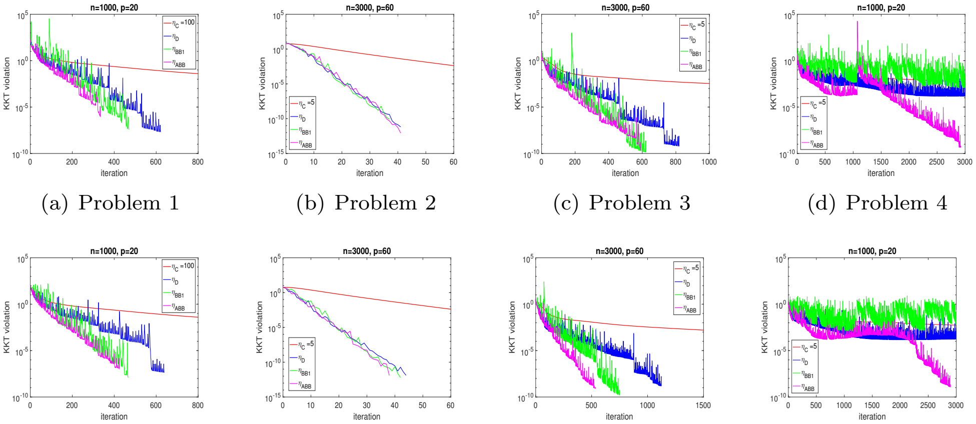
$$\min_{X\in\mathbb{R}^{n\times p}} E(X) \qquad \text{s. t. } X^\top X = I_p, \tag{5.9}$$

where the discretized Kohn–Sham total energy function $E(X)$ is defined by (1.2) with the exchange correlation function ϵ_{xc} taking the widely accepted formula developed in [24]. All the data comes from MATLAB toolbox KSSLOV.

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

$$\begin{array}{cc} \min\limits_{X\in\mathbb{R}^{n\times p}} & \frac{1}{2}\mathrm{tr}(X^\top LX) + \frac{1}{2}\rho(X)^\top L^\dagger \rho(X) - \frac{3}{4}\gamma\rho(X)^\top \rho(X)^{\frac{1}{3}} \\ \text{s. t.} & X^\top X = I_p, \end{array} \tag{5.10}$$

where the matrix $L \in \mathbb{R}^{n\times n}$ and $\rho(X) := \mathrm{diag}(XX^\top)$. The parameter $\gamma = 2(\frac{3}{\pi})^{1/3}$ and $\rho(X)^{\frac{1}{3}}$ denotes the componentwise 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 [17]. The generation of L is in the same manner as in Problem 1.



(a) Problem 1 (b) Problem 2 (c) Problem 3 (d) Problem 4

(e) Problem 1 (f) Problem 2 (g) Problem 3 (h) Problem 4

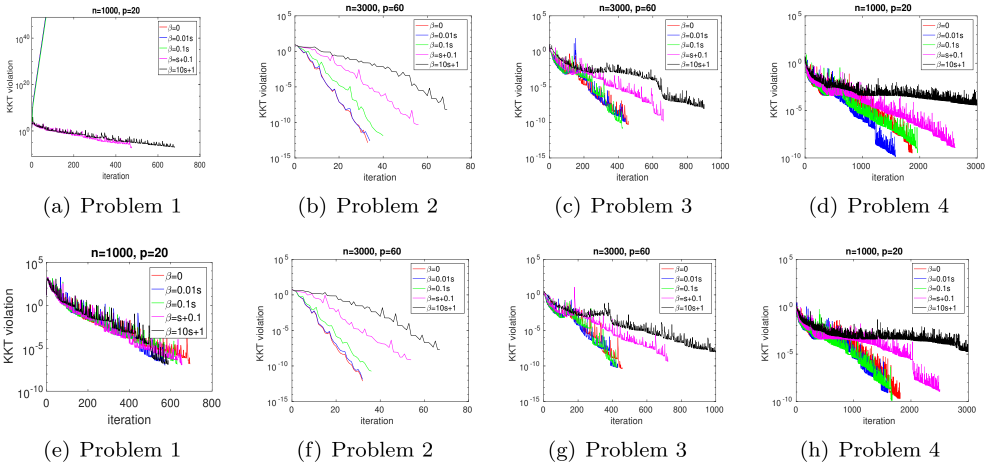
FIG. 5.1. *A comparison of KKT violation for PLAM* (a)–(d) *and PCAL* (e)–(h) *with different* η *(*$\beta = s + 0.1$*).*

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 $\beta = s + 0.1$. Figure 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.

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 $\eta = \eta_{\mathrm{ABB}}$. 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 learned 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 a 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.

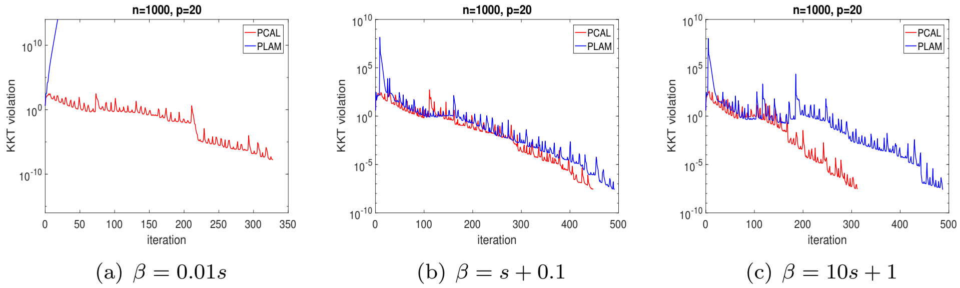
There are two distinctions between PLAM and ALM. Firstly, a gradient step takes the place of solving the subproblem to some given precision in the update of the prime variables. Secondly, a closed-form expression is used to update the Lagrangian multipliers instead of dual ascend. In order to show that 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



(a) Problem 1 (b) Problem 2 (c) Problem 3 (d) Problem 4

(e) Problem 1 (f) Problem 2 (g) Problem 3 (h) Problem 4

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



(a) $\beta = 0.01s$ (b) $\beta = s + 0.1$ (c) $\beta = 10s + 1$

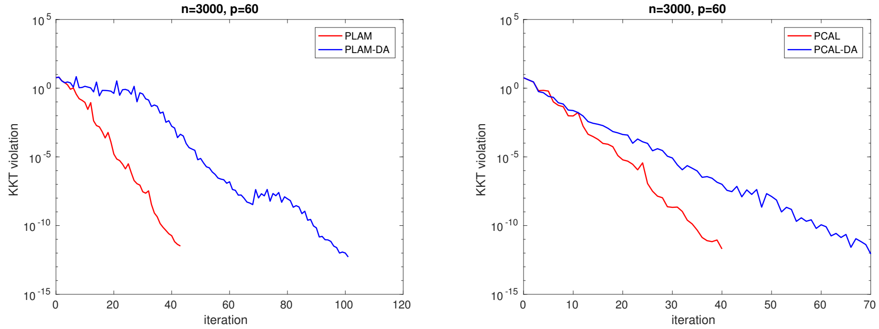
FIG. 5.3. *A comparison between PLAM and PCAL with different* β *on Problem* 1.

and PCAL-DA stand for Algorithms 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.

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 with our theoretical analysis of Lemma 2.5. 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 the orthonormalization step

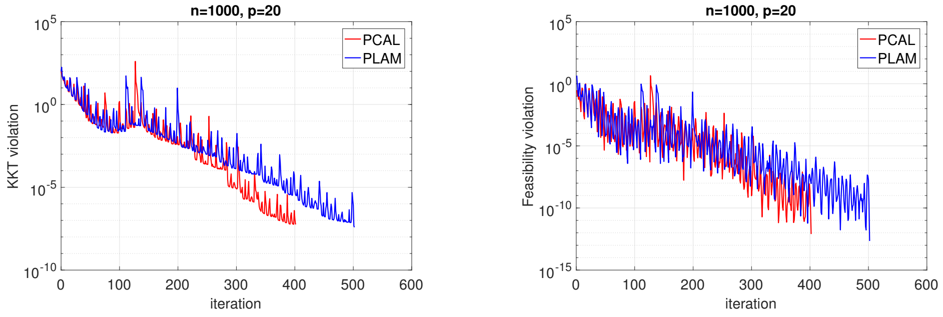
$$\mathrm{orth}(X^*) := UV^\top, \tag{5.11}$$

where $X^* = U\Sigma V^\top$ is the SVD of X^* with $U \in \mathcal{S}_{n,p}$, $\Sigma \in \mathbb{D}^p$, and $V \in \mathcal{S}_{p,p}$, as a postprocess when we obtain the last iterate X^* by PLAM or PCAL. Proposition 6.1 in Appendix B guarantees that postprocess (5.11) does not affect the KKT violation too much in theory, particularly when δ is sufficiently large, which implies a large β.



(a) Problem 1 (b) Problem 2

FIG. 5.4. *A comparion bewteen PLAM and PCAL on multilplier.*



(a) KKT violation (b) Feasibility violation

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

Numerically, Table 5.1 illustrates that such a postprocess does not affect the KKT violation but does improve the feasibility. Here, "X^*" and "orth(X^*)" represent the relative values at the last iterate and the one after postprocess, respectively. Hereinafter, the orthonormalization postprocess, achieved by an internal function $\texttt{qr}(\cdot)$ assembled in MATLAB, is the default last step of PLAM and PCAL.

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 words, 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.

Our test is based on KSSOLV[6] [32], which is a MATLAB toolbox for electronic structure calculation. It allows researchers to investigate their own algorithms in an easy and friendly manner for different steps in electronic structure calculation. We choose two integrated solvers in KSSOLV. One is the self-consistent field (SCF) iteration, which minimizes a quadratic surrogate of the objective of (5.9) with orthogonality constraints in each iteration [16]. SCF and its variations are the most widely used in real KSDFT calculation. The other one is called trust-region direct constrained minimization (TRDCM) [34], which combines the trust-region framework

[6] Available from http://crd-legacy.lbl.gov/∼chao/KSSOLV/.

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

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

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 OptM,[7] which is based on the algorithm proposed in [30]. 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. As 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 search[8]). 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

$$\left\|(I_n - XX^\top)\nabla f(X)\right\|_{\mathrm{F}} < 10^{-5}.$$

And we set the max iteration number MaxIter = 200 for methods SCF and TRDCM, while MOptQR, OptM, PLAM, and PCAL set their max iteration number with 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 detailed numerical results are illustrated in Appendix C. Since we have different problems and different solvers, to make a more straightforward comparison, we use performance profiles [8] to visualize the expected performance difference among those solvers. We describe such a test in the following. For problem m and solver s, we denote $t_{m,s}$ to represent the CPU time. Performance ratio is defined as $r_{m,s} := t_{m,s}/\min_s\{t_{m,s}\}$. If solver s fails to solve problem m, the ratio $r_{m,s}$ will be set to infinity or some sufficiently large number. Finally, the overall performance of solver s is defined by

$$\pi_s(\omega) := \frac{\text{number of problems where } r_{m,s} \le \omega}{\text{total number of problems}}.$$

It means the percentage of testing problems that can be solved in $\omega \min_s t_{m,s}$ seconds. Of course, the closer π_s is to 1, the better performance solver s has. The performance profile results with respect to CPU time are given in Figure 5.6. We observe that PCAL performs best among all six algorithms in solving Kohn–Sham total energy minimization problems in CPU time.

[7] Available from http://optman.blogs.rice.edu.

[8] Available from http://www.manopt.org.

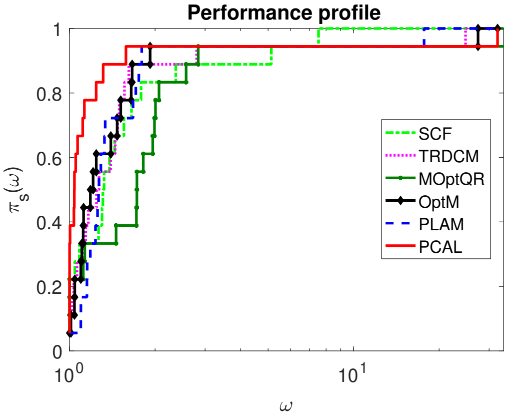


Fig. 5.6. *Performance profile in CPU time.*

5.5. Parallel efficiency. In this subsection, we examine the parallel efficiency of our algorithms PLAM 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.

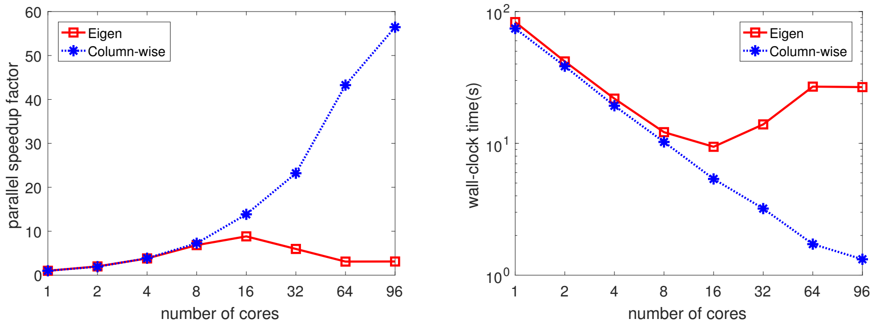
Both algorithms are implemented in the C++ language and parallelized by OpenMP. The linear algebra library we used in comparison is Eigen[9] (version 3.3.4), which is an open and popular C++ template library for matrix computation. We define the speedup factor for running a code on m cores as

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

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 nonnegligible 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 multithreading computing[10] 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 columnwise 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=Random(1000,10000)` and `B=Random(10000,1000)`, where "`Random(·,·)`" 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 5.7. "Eigen" and "Columnwise" represent the default parallel strategy and columnwise product strategy, respectively. We can observe that columnwise parallelization obviously outperforms the default setting of Eigen in multithreading computing. Hence, in the following implementation, we choose a columnwise parallelization strategy for BLAS3 in our experiments.

[9] Available from http://eigen.tuxfamily.org/index.php?title=Main_Page.

[10] More information at http://eigen.tuxfamily.org/dox/TopicMultiThreading.html.



(a) Speedup factor (b) Wall-clock time(s)

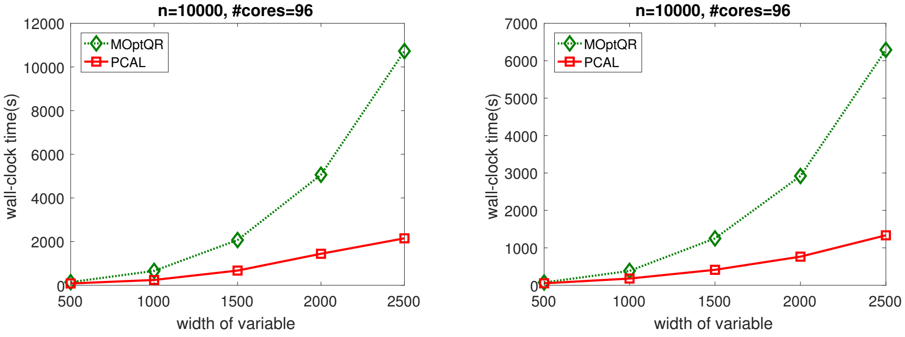
FIG. 5.7. *The results of dense-dense BLAS3:* $A^{1000\times 10000}B^{10000\times 1000}$.

Next, we investigate the parallel scalability of the new proposed PCAL and MOptQR. According to the existent numerical report of Eigen,[11] 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 $X^0 =$`random(n,p)`" and $X^0 =$`qr`(X^0).

We first focus on the test Problems 1 and 2. For Problem 1, we set L as a block diagonal matrix, i.e., $L = \mathrm{Diag}(L_1, \ldots, L_s)$, where $L_i \in \mathbb{R}^{5\times 5}$ is a tridiagonal matrix with 2 on its main diagonal and -1 on subdigonal, for $i = 1, \ldots, s$. 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 subdigonal and G=`Random(n,p)`. The advantage of such a 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 5.8. Here, "#cores" stands for the number of cores. From Figure 5.8, we notice that PCAL always takes less wall-clock time than MOptQR. As the width of the matrix variable increases, the running time of MOptQR increases much more rapidly than that of PCAL.

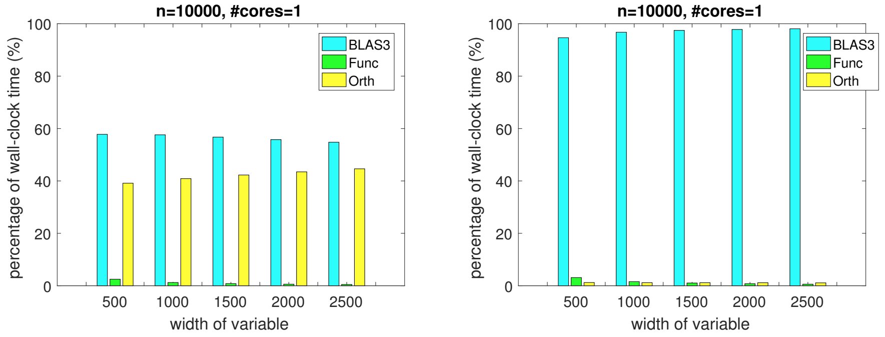
In Figure 5.9, we show wall-clock time of three categories: "BLAS3" (dense-dense matrix multiplication), "Func" (function value and gradient evaluation), and "Orth" (orthonormalization including QR factorization for MOptQR and the final correction step in PCAL). These are the major computational components 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

[11] More information at http://eigen.tuxfamily.org/dox/group__TutorialLinearAlgebra.html.



(a) Problem 1 (b) Problem 2

FIG. 5.8. *The wall-clock time results on varying width of the matrix variable.*



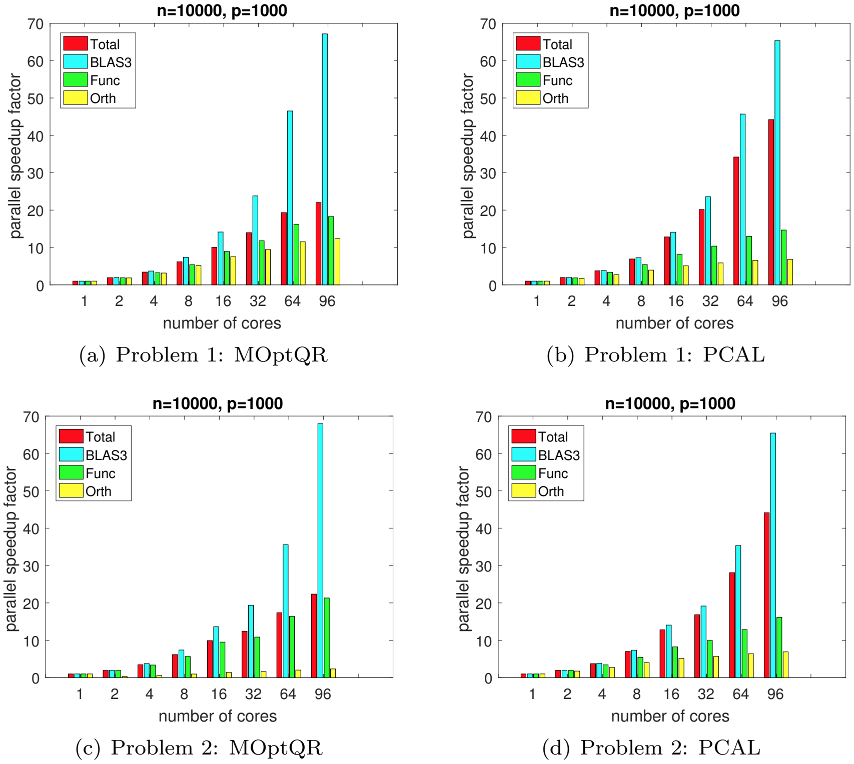
(a) MOptQR (b) PCAL

FIG. 5.9. *A comparison of timing profile on a single core for Problem* 2.

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 increases 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. Figures 5.10 and 5.11 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 increases, the advantage of PCAL in parallel scalability becomes more obvious.

In the end, we test Problem 6 under $n = 10000$, $p = 1000$. Figure 5.12 illustrates the results of speedup factors 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.



(a) Problem 1: MOptQR

(b) Problem 1: PCAL

(c) Problem 2: MOptQR

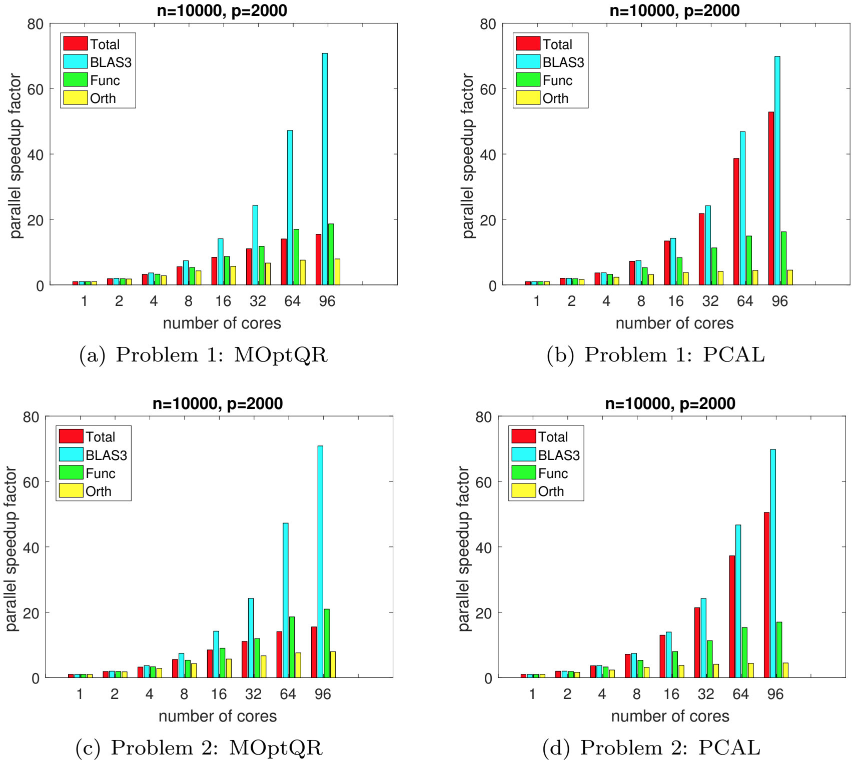
(d) Problem 2: PCAL

FIG. 5.10. *A comparison of speedup factor among MOptQR and PCAL ($p = 1000$).*

6. Conclusion.

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

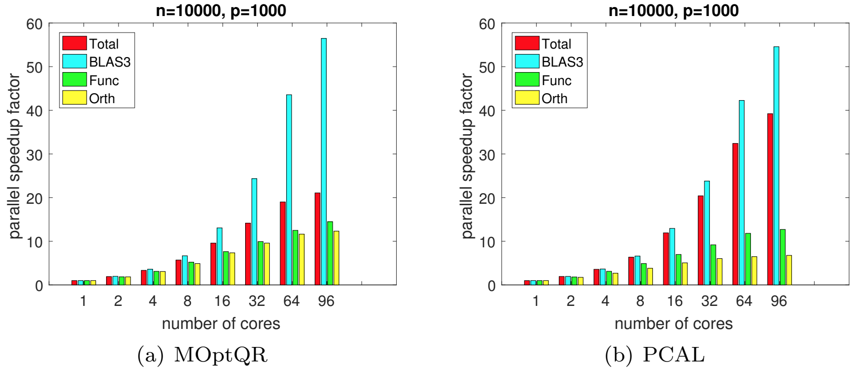
To solve this issue, we need to employ infeasible approaches. However, previous infeasible approaches, including, the ALM, far less efficient than retraction based feasible methods. Even though the parallelization reduces the running time of ALM more significantly than that of manifold methods, ALM is 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 of DA 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.



(a) Problem 1: MOptQR (b) Problem 1: PCAL

(c) Problem 2: MOptQR (d) Problem 2: PCAL

FIG. 5.11. *A comparison of speedup factor among MOptQR and PCAL* ($p = 2000$).



(a) MOptQR (b) PCAL

FIG. 5.12. *A comparison of speedup factor among MOptQR and PCAL on the simplified Kohn-Sham total energy minimization.*

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 in an efficient, robust, and insensitive manner 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 future work, we will apply our PCAL to real KSDFT calculation. Moreover, we have not mentioned the performances of our algorithms in solving linear eigenvalue problems, compared with other solvers. This is because some linear algebraic issues should be taken into account if we want to tune our algorithms as linear eigenvalue solvers, which are beyond the scope of this paper. But we are interested in investigating it in the future.

Appendix A. We implement the ADMM-based algorithm introduced in [13], which is called splitting orthogonality constraints (SOC). The testing problems are Problems 1–4 introduced in subsection 5.2 with the same settings as those in subsection 5.3. We compare SOC with our PCAL. The PCAL takes its default setting. The penalty parameter r of SOC is sensitive to the performance; we tune their algorithm many times and choose the parameter to be the one with the best performance, namely, $r = 90, 1, 5, 5$ in Problems 1–4, respectively. The tolerance of the inner iteration of SOC is set to 10^{-8}.

Figures 6.1 and 6.2 illustrate the change of KKT and feasibility violation of PCAL's and SOC's iterates as the iterations progress. Figure 6.3 mainly shows the number of inner iterations of SOC. We can learn from these figures that

- PCAL converges faster than SOC in these testing problems;
- PCAL requires much less computation cost than SOC in each iteration, given that the computational cost of one iteration in PCAL is in the same order of its in an inner iteration of SOC without orthonormalization.

We have as a byproduct of this experiment that the decreases of the KKT violation and feasibility violation of PCAL have similar trends.

Appendix B.

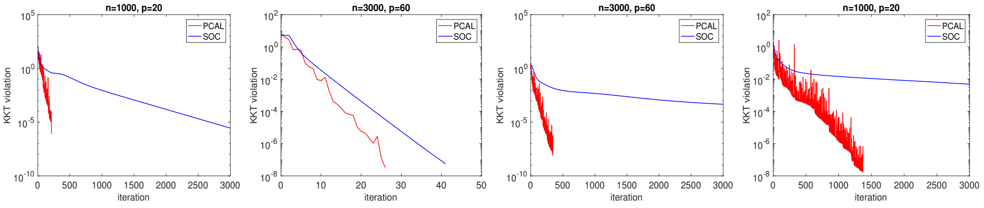
Proposition 6.1. *Suppose the objective function f satisfies* (4.1) *and the assumption in Lemma* 2.5 *holds. Let $\tilde{X} = \mathrm{orth}(X^*)$, where* orth *is defined by* (5.11)*. Then it holds that*

(6.1)
$$||\nabla_X \mathcal{L}_\beta(\tilde{X}, \tilde{\Lambda})||_{\mathrm{F}} \leq \left(1 + \frac{(2L + (N+\beta)||X^*||_2 + N)\,||X^*||_2}{\delta}\right) \cdot ||\nabla_X \mathcal{L}_\beta(X^*, \Lambda^*)||_{\mathrm{F}},$$

where $\Lambda^* = \Psi(\nabla f(X^*)^\top X^*)$ and $\tilde{\Lambda} = \Psi(\nabla f(\tilde{X})^\top \tilde{X})$.

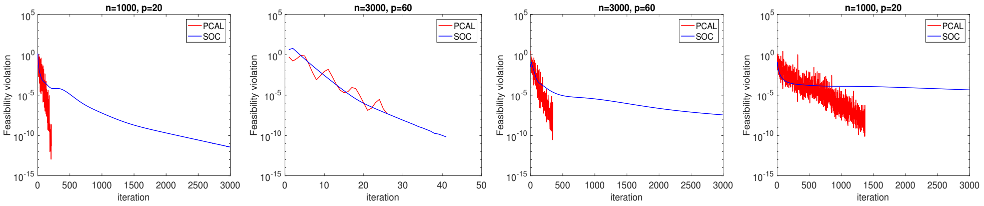
Proof. Due to the fact that $||\Sigma - I||_{\mathrm{F}} \leq ||(\Sigma - I)(\Sigma + I)||_{\mathrm{F}} = ||\Sigma^2 - I||_{\mathrm{F}}$, it holds that

$$\begin{aligned}
&||\nabla_X \mathcal{L}_\beta(X^*, \Lambda^*) - \nabla_X \mathcal{L}_\beta(\tilde{X}, \tilde{\Lambda})||_{\mathrm{F}} \\
&\quad \leq ||\nabla f(X) - \nabla f(\tilde{X})||_{\mathrm{F}} + ||X^* \nabla f(X^*)^\top X^* - X^* \nabla f(X^*)^\top \tilde{X}||_{\mathrm{F}} \\
&||X^* \nabla f(X^*)^\top \tilde{X} - \tilde{X} \nabla f(X^*)^\top \tilde{X}||_{\mathrm{F}} + ||\tilde{X} \nabla f(X^*)^\top \tilde{X} - \tilde{X} \nabla f(\tilde{X})^\top \tilde{X}||_{\mathrm{F}} \\
&\qquad\qquad + \beta||X^*({X^*}^\top X^* - I_p)||_{\mathrm{F}} \\
&\leq L \cdot ||X^* - \tilde{X}||_{\mathrm{F}} + N \cdot ||X^*||_2 \cdot ||X^* - \tilde{X}||_{\mathrm{F}} + N \cdot ||X^* - \tilde{X}||_{\mathrm{F}} + L \cdot ||X^* - \tilde{X}||_{\mathrm{F}} \\
&\qquad\qquad + \beta||X^*||_2 \cdot ||{X^*}^\top X^* - I_p||_{\mathrm{F}} \\
&= (2L + N||X^*||_2 + N) \cdot ||U\Sigma V^\top - UV^\top||_{\mathrm{F}} + \beta||X^*||_2 \cdot ||{X^*}^\top X^* - I_p||_{\mathrm{F}} \\
&\leq (2L + (N+\beta)||X^*||_2 + N) \cdot ||{X^*}^\top X^* - I_p||_{\mathrm{F}}.
\end{aligned}$$



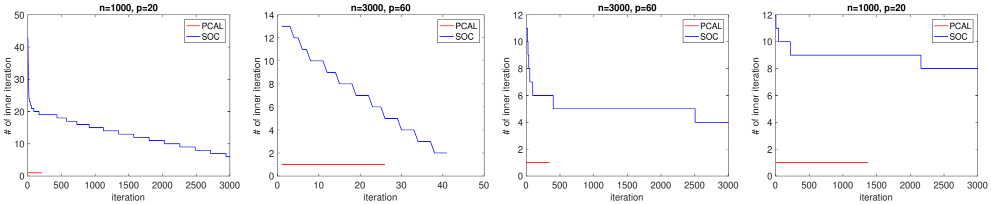
(a) Problem 1 (b) Problem 2 (c) Problem 3 (d) Problem 4

Fig. 6.1. *A comparison of KKT violation for PCAL and ADMM.*



(a) Problem 1 (b) Problem 2 (c) Problem 3 (d) Problem 4

Fig. 6.2. *A comparison of feasibility violation for PCAL and ADMM.*



(a) Problem 1 (b) Problem 2 (c) Problem 3 (d) Problem 4

Fig. 6.3. *A comparison of inner iteration for PCAL and ADMM.*

From the above inequality and (2.20), we obtain

$$\begin{aligned}||\nabla_X\mathcal{L}_\beta(\tilde{X},\tilde{\Lambda})||_{\mathrm{F}} - ||\nabla_X\mathcal{L}_\beta(X^*,\Lambda^*)||_{\mathrm{F}} &\leq ||\nabla_X\mathcal{L}_\beta(X^*,\Lambda^*) - \nabla_X\mathcal{L}_\beta(\tilde{X},\tilde{\Lambda})||_{\mathrm{F}}\\ &\leq (2L+(N+\beta)||X^*||_2+N)\cdot||{X^*}^\top X^* - I_p||_{\mathrm{F}}\\ &\leq \frac{(2L+(N+\beta)||X^*||_2+N)\,||X^*||_2}{\delta}\cdot||\nabla_X\mathcal{L}_\beta(X^*,\Lambda^*)||_{\mathrm{F}},\end{aligned}$$

which implies the inequality (6.1). □

Remark 6.2. Suppose β is close enough to $(||\nabla f(X^*)||_2\cdot||X^*||_2+\delta)/\sigma^2_{\min}(X^*)$ and δ is sufficiently large; then the coefficient of (6.1) is close to $(1+\frac{||X^*||_2^2}{\sigma^2_{\min}(X^*)})$, which is further close to 2 when X^* is almost orthogonal.

Appendix C. In this section, we illustrate the detailed numerical results of subsection 5.4 in Tables 6.1, 6.2, and 6.3. Here, "E_{tot}" represents the total energy function value, and "KKT violation," "Iteration," "Feasibility violation," and "Time(s)" stand for $\left\|(I_n - XX^\top)\nabla f(X)\right\|_{\mathrm{F}}$, the number of iteration, $\left\|X^\top X - I_p\right\|_{\mathrm{F}}$, 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

Table 6.1
The results in Kohn–Sham total energy minimization.

Solver	E_{tot}	KKT violation	Iteration	Feasibility violation	Time(s)
	al, $n = 16879$, $p = 12$		($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)		
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, $n = 12671$, $p = 18$		($\beta_{\text{PLAM}} = 13, \beta_{\text{PCAL}} = 1$)		
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, $n = 8407$, $p = 15$		($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)		
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, $n = 2103$, $p = 7$		($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)		
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, $n = 5709$, $p = 37$		($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)		
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, $n = 2103$, $p = 8$		($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)		
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 6.2
The results in Kohn–Sham total energy minimization.

Solver	E_{tot}	KKT violation	Iteration	Feasibility violation	Time(s)
	ctube661, $n = 12599$, $p = 48$			$(\beta_{\text{PLAM}} = 13, \beta_{\text{PCAL}} = 1)$	
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, $n = 16517$, $p = 29$			$(\beta_{\text{PLAM}} = 13, \beta_{\text{PCAL}} = 1)$	
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, $n = 3071$, $p = 37$			$(\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1)$	
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, $n = 12279$, $p = 67$			$(\beta_{\text{PLAM}} = 13, \beta_{\text{PCAL}} = 1)$	
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, $n = 2103$, $p = 4$			$(\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1)$	
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, $n = 2103$, $p = 8$			$(\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1)$	
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 6.3
The results in Kohn–Sham total energy minimization.

Solver	E_{tot}	KKT violation	Iteration	Feasibility violation	Time(s)
nic, $n = 251$, $p = 7$ ($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)					
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, $n = 44791$, $p = 51$ ($\beta_{\text{PLAM}} = 13, \beta_{\text{PCAL}} = 1$)					
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, $n = 4069$, $p = 43$ ($\beta_{\text{PLAM}} = 13, \beta_{\text{PCAL}} = 1$)					
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, $n = 2103$, $p = 8$ ($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)					
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, $n = 2103$, $p = 6$ ($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)					
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, $n = 2103$, $p = 4$ ($\beta_{\text{PLAM}} = 10, \beta_{\text{PCAL}} = 1$)					
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

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.

Acknowledgments. The authors would like to thank Michael Overton, Tao Cui, and Xingyu Gao for the insightful discussions.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2009.

[2] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[3] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 2014.

[4] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization or nonconvex and nonsmooth problems*, Math. Program., 146 (2014), pp. 459–494.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends. Mach. Learn., 3 (2011), pp. 1–122.

[6] X. Dai, Z. Liu, L. Zhang, and A. Zhou, *A conjugate gradient method for electronic structure calculations*, SIAM J. Sci. Comput., 39 (2017), pp. A2702–A2740.

[7] Y.-H. Dai and R. Fletcher, *Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming*, Numer. Math., 100 (2005), pp. 21–47.

[8] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.

[9] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[10] B. Gao, X. Liu, X. Chen, and Y.-x. Yuan, *A new first-order algorithmic framework for optimization problems with orthogonality constraints*, SIAM J. Optim., 28 (2018), pp. 302–332.

[11] B. Jiang and Y.-H. Dai, *A framework of constraint preserving update schemes for optimization on Stiefel manifold*, Math. Program., 153 (2015), pp. 535–575.

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

[13] R. Lai and S. Osher, *A splitting method for orthogonality constrained problems*, J. Sci. Comput., 58 (2014), pp. 431–449.

[14] Y. Li, Z. Wen, C. Yang, and Y. Yuan, *A Semi-Smooth Newton Method for Solving Semidefinite Programs in Electronic Structure Calculations*, 2017, https://arxiv.org/abs/1708.08048.

[15] J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar, *An asynchronous parallel stochastic coordinate descent algorithm*, J. Mach. Learn. Res., 16 (2015), pp. 285–322.

[16] X. Liu, X. Wang, Z. Wen, and Y. Yuan, *On the convergence of the self-consistent field iteration in Kohn–Sham density functional theory*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 546–558.

[17] X. Liu, Z. Wen, X. Wang, M. Ulbrich, and Y. Yuan, *On the analysis of the discretized Kohn–Sham density functional theory*, SIAM J. Numer. Anal., 53 (2015), pp. 1758–1785.

[18] X. Liu, Z. Wen, and Y. Zhang, *An efficient Gauss–Newton algorithm for symmetric low-rank product matrix approximations*, SIAM J. Optim., 25 (2015), pp. 1571–1608.

[19] M. A. Marques, A. Castro, G. F. Bertsch, and A. Rubio, *octopus: A first-principles tool for excited electron–ion dynamics*, Comput Phys. Commun., 151 (2003), pp. 60–78.

[20] Y. Nishimori and S. Akaho, *Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold*, Neurocomputing, 67 (2005), pp. 106–135.

[21] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.

[22] Z. Peng, Y. Xu, M. Yan, and W. Yin, *ARock: An algorithmic framework for asynchronous parallel coordinate updates*, SIAM J. Sci. Comput., 38 (2016), pp. A2851–A2879.

[23] Z. Peng, M. Yan, and W. Yin, *Parallel and distributed sparse optimization*, in Proceedings of the Asilomar Conference on Signals, Systems and Computers, IEEE, 2013, pp. 659–646.

[24] J. P. Perdew and A. Zunger, *Self-interaction correction to density-functional approximations for many-electron systems*, Phys. Rev., B, 23 (1981), 5048.

[25] M. J. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, London, 1969, pp. 283–298.

[26] 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, NIPS, 2011, pp. 693–701.

[27] M. Ulbrich, Z. Wen, C. Yang, D. Klockner, and Z. Lu, *A proximal gradient method for ensemble density functional theory*, SIAM J. Sci. Comput., 37 (2015), pp. A1975–A2002.

[28] 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.

[29] Z. Wen, C. Yang, X. Liu, and Y. Zhang, *Trace-penalty minimization for large-scale eigenspace computation*, J. Sci. Comput., 66 (2016), pp. 1175–1203.

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

[31] 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 J. Matrix Anal. Appl., 30 (2009), pp. 1773–1788.

[32] C. Yang, J. C. Meza, B. Lee, and L.-W. Wang, *KSSOLV–a MATLAB toolbox for solving the Kohn-Sham equations*, ACM Trans. Math. Softw, 36 (2009), p. 10.

[33] C. Yang, J. C. Meza, and L.-W. Wang, *A constrained optimization algorithm for total energy minimization in electronic structure calculations*, J. Comput. Phys., 217 (2006), pp. 709–721.

[34] 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.

[35] X. Zhang, J. Zhu, Z. Wen, and A. Zhou, *Gradient type optimization methods for electronic structure calculations*, SIAM J. Sci. Comput., 36 (2014), pp. C265–C289.