AN ALTERNATE GRADIENT METHOD FOR OPTIMIZATION PROBLEMS WITH ORTHOGONALITY CONSTRAINTS

YANMEI SUN
School of Science, Hebei University of Technology
Tianjin, 300401, P. R. China

YAKUI HUANG∗
Institute of Mathematics, Hebei University of Technology
Tianjin, 300401, P. R. China

(Communicated by Zhong Wan)

Abstract. In this paper, we propose a new alternate gradient (AG) method to solve a class of optimization problems with orthogonal constraints. In particular, our AG method alternately takes several gradient reflection steps followed by one gradient projection step. It is proved that any accumulation point of the iterations generated by the AG method satisfies the first-order optimal condition. Numerical experiments show that our method is efficient.

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

$$\min_{X \in \mathbb{R}^{n \times p}} f(X)$$

$$\text{s.t. } X^T X = I_p$$

where $f : \mathbb{R}^{n \times p} \to \mathbb{R}$ with $p \leq n$, $X \in \mathbb{R}^{n \times p}$ is the variable and $I_p$ is the $p$-by-$p$ identity matrix. The feasible set $S_{n,p} = \{X \in \mathbb{R}^{n \times p} | X^T X = I_p\}$ is often referred to as the Stiefel manifold.

Throughout the paper, we assume that the objective function satisfies the following assumption.

Assumption 1.1. $f$ is twice differentiable. $f(X)$ can be represented as $h(X) + \text{tr}(G^T X)$, where $h(X)$ is orthogonal invariant, $h(XQ) = h(X)$ holds for any $Q \in S_{p,p}$, $\nabla h(X) = H(X)X$, $G \in \mathbb{R}^{n \times p}$, and $\text{tr}(\cdot)$ is the trace of a matrix.

The above assumption is satisfied by many practical problems, for example,

$$f(X) = \frac{1}{2} \text{tr}(X^T AX) + \text{tr}(G^T X),$$

2010 Mathematics Subject Classification. Primary: 65F15, 65K05; Secondary: 90C06.
Key words and phrases. Orthogonality constraint, Stiefel manifold, Alternate gradient method, Gradient projection, Gradient reflection.

The second author is supported by NSFC grant 11701137.
∗ Corresponding author: Yakui Huang.
where $A \in S_n$, $S_n$ refers to the set of $n \times n$ symmetric matrices. We notice that if $G = 0$, $f(X)$ in (2) reduces to the objective function of the eigenvalue problem. Another example appears in electronic structure calculations, where

$$f(X) = \frac{1}{2} \text{tr}(X^TAX) + \frac{1}{2} \sum_{i=1}^{m} q_i(z),$$

with $A \in S_n$, $z = \text{diag}(XX^T)$, $q_i : R^n \to R$ and diag$(B)$ represents the vector formed by the diagonal entries of matrix $B$.

Optimization problems with orthogonality constraints have many applications in data science and scientific engineering computing, such as eigenvalue problem [11, 14], electronic structure calculations [12, 15, 17, 21], sparse principal component analysis [4, 20] and the orthogonal Procrustes problem [6].

Problem (1) is difficult to solve due to the nonconvexity of orthogonal constraints and several of these problems in special forms are NP-hard [9]. In order to ensure feasibility, the concept retraction was introduced, which maps a tangent vector into the manifold, see [1] for example. Geodesic method is a kind of commonly used retraction method. In [7], Edelman et al. proposed a computable geodesic update scheme, which requires the calculation of the matrix exponential of order $2p \times 2p$ in each iteration. However, the above method is computationally expensive when $p > n/2$. Abrudan et al. [2] presented another geodesic method that needs to calculate the matrix exponent of order $n \times n$ in each iteration. To avoid computing exponentials of matrices, based on Cayley transform, Nishimori et al. [13] introduced a quasi-geodesic method that requires to solve a $n \times n$ linear system. Wen et al. [16] used the Sherman-Morrison-Woodbury (SMW) theorem to improve the formula in [13] and proposed the GBB method, which incorporates the Barzilai-Borwein (BB) method [3] and nonmonotone line search [19]. One great advantage of the GBB method is that it only needs to solve a $2p \times 2p$ linear system in each iteration. By decomposing each feasible point $X$ into the range space of $X$ and the null space of $X^T$, Jiang et al. [10] developed a framework of constraint preserving update schemes which unifies most existing schemes. They further suggested to combine the two BB stepsizes alternately with a nonmonotone line search and presented the AFBB method. Recently, Gao et al. [8] introduced a new first-order algorithmic framework that combines a function value reduction step with a correction step. A remarkable feature of their framework is the function value reduction step searches along the standard Euclidean descent directions while the correction step not only reduces the function value but also guarantees a symmetric dual variable. Based on the framework, two efficient methods, namely gradient projection (GP) and gradient reflection (GR), were proposed. Moreover, they suggested to combine the two methods with BB stepsizes to get good performance.

In this paper, motivated by the success of the GR and GP methods, we consider to take both GR and GP steps to solve problem (1) which results in an alternate gradient (AG) method. Particularly, our AG method takes several GR steps followed by one GP step to capture the advantages of the two methods. We prove that any accumulation point of the iterations generated by our AG method satisfies the first-order optimal condition. Numerical comparisons with GP, GR, GBB and AFBB illustrate the effectiveness of the AG method.

The rest of this paper is arranged as follows. In Section 2, we present our AG method. In Section 3, we prove the global convergence of our AG method.
2. **Alternate gradient method.** Let $\nabla f(X)$ be the Euclidean gradient. The first-order optimality condition of problem (1) can be expressed as follows.

**Definition 2.1.** For a given point $X \in R_{n \times p}$, if the following relationship

$$
\text{tr}(Y^T \nabla f(X)) \geq 0,
$$

(4)

$$
X^T X = I_p,
$$

(5)

holds for all $Y \in T_X S_{n,p}$, then $X$ is called a first-order stationary point of (1), where $T_X S_{n,p} = \{ Y \in R_{n \times p} \mid Y^T X + X^T Y = 0 \}$ is the tangent space of the Stiefel manifold $S_{n,p}$ at $X$.

Since condition (4) cannot be verified numerically, Gao et al. [8] proved that $X$ is a first-order stationary point if and only if

$$
(I_n - XX^T) \nabla f(X) = 0,
$$

(6)

$$
X^T \nabla f(X) = \nabla f(X)^T X,
$$

(7)

$$
X^T X = I_p,
$$

(8)

where the three equations in (6)-(8) are called substationarity, symmetry and feasability, respectively.

Our AG method employs the GP and GR steps in [8]. Suppose we have $X^k$ and let $V = X^k - \tau \nabla f(X^k)$ for some $\tau > 0$, the GP step is defined by

$$
P_{GP}(V) = \arg \min_{X \in S_{n,p}} \|X - V\|_F^2
$$

(9)

and the GR step is given by

$$
P_{GR}(V) = (-I_n + 2VV^T V^\dagger V^T) X^k,
$$

(10)

where $B^\dagger$ means the pseudoinverse of $B$. Under the condition $\tau \in (0, \rho^{-1})$, where $\rho = \sup \|\nabla^2 f(X)\|_2 \text{ and } S = \{ Y \mid \|Y\|^2_F < p + 1 \}$, it can be shown that both the GP and GR steps provide feasible points and reduce the objective values.

**Lemma 2.2.** [8] Let $\theta = \|G\|_2$, $X^k \in S_{n,p}$, $\bar{X}^k_{GR} = P_{GR}(V)$ and $\bar{X}^k_{GP} = P_{GP}(V)$. Then it holds that $\bar{X}^k_{GR} \in S_{n,p}$, $\bar{X}^k_{GP} \in S_{n,p}$ and

$$
f(X^k) - f(\bar{X}^k_{GR}) \geq \frac{2(\tau^{-1} - \rho)}{(\tau^{-1} + \rho + \theta)^2} \cdot \|(I_n - X^k(X^k)^T) \nabla f(X^k)\|^2_F,
$$

$$
f(X^k) - f(\bar{X}^k_{GP}) \geq \frac{(\tau^{-1} - \rho)}{2(\tau^{-1} + \rho + \theta)^2} \cdot \|(I_n - X^k(X^k)^T) \nabla f(X^k)\|^2_F.
$$

Clearly, for the same $X^k$, the GR step has a larger reduction in objective value than the GP step. However, the computational cost of a GP step is $7np^2 + 3np + O(p^3)$ which is lower than a GR step’s cost $9np^2 + 4np + O(p^3)$. As $p$ increases, the cost difference between the GR and GP steps will become larger. In order to capture the advantages of the two steps, we consider to take some GR steps followed by one GP step which gives the AG method described in Algorithm 1, where $c(X) = (I_n - XX^T) \nabla f(X)$. 

Numerical experiments are reported in Section 4. Finally, conclusions are given in the last section.
Algorithm 1: AG method

Input: $X^0 \in S_{n,p}$, a positive integer $m$, $\epsilon > 0$, and set $k := 0$

1. while $\|c(X^k)\|_F > \epsilon$
   
   2. $V = X^k - \tau \nabla f(X^k)$;
   
   3. if $\text{mod}(k, m) = 0$
      
      4. $\bar{X}^k = P_{GP}(V)$;
   
   5. else
      
      6. $\bar{X}^k = P_{GR}(V)$;
   
   7. end

8. Based on $\bar{X}^k$, calculate a feasible point;

9. $X^{k+1} = \left\{ \begin{array}{ll}
               \bar{X}^k, & (\bar{X}^k)^T G = G^T \bar{X}^k, \\
               -\bar{X}^k U T^T, & (\bar{X}^k)^T G \neq G^T \bar{X}^k,
             \end{array} \right.$

10. where $U$ and $T$ come from the singular value decomposition $(\bar{X}^k)^T G = U \Lambda T^T$;

11. Set $k = k + 1$.

12. end

13. Return $X^k$.

Note that, by Lemma 2.2, $\bar{X}^k$ obtained by the AG method is always feasible. The correction step $X^{k+1}$ in line 9 is necessary to get a point satisfying the symmetry condition in (7). Hence the AG method stops with a stationary point when $\epsilon = 0$. We mention that our AG method reduces to the GP method when $m = 1$ and to the GR method when $m = +\infty$.

To illustrate the affect of our alternate scheme, we compare GP, GR, AG ($m = 2$), AG ($m = 3$) and AG ($m = 5$) on problem (2) with $n = 500$, $p = 6$. Numerical results are presented in Table 1, where “cpu” means CPU time in seconds, “iter” mean the number of iterations, “fun” means function value, “KKT violation” is the value of $\|\nabla f(X^k) - X \nabla f(X^k) T X\|_F$, and “feasibility” is the value of $\|X^T X - I_p\|_F$. From Table 1 we see that AG takes less CPU time and fewer number of iterations than GP and GR. Moreover, the function value, KKT violation and feasibility obtained by our AG method are as good as those by GP and GR.

Table 1. The results of GP, GR and AG on problem (2) with $n = 500$, $p = 6$.

|       | cpu  | iter  | fun          | KKT violation | feasibility |
|-------|------|-------|--------------|---------------|-------------|
| GP    | 3.74 | 682.0 | -91.651414   | 3.8477E-04    | 4.0577E-15  |
| GR    | 2.40 | 502.4 | -91.646891   | 3.8305E-04    | 1.9524E-13  |
| AG($m = 2$) | 2.01 | 433.4 | -91.646891   | 3.8447E-04    | 4.5582E-15  |
| AG($m = 3$) | 2.21 | 483.4 | -91.642235   | 3.8370E-04    | 6.2797E-15  |
| AG($m = 5$) | 2.34 | 503.0 | -91.646891   | 3.8142E-04    | 6.1260E-15  |

3. Convergence analysis. In this section, we establish the global convergence of our AG method.

Lemma 3.1. Let $\{\bar{X}^k\}$ be the sequence generated by Algorithm 1. Then it holds that $X^k \in S_{n,p}$ and

$$f(X^k) - f(\bar{X}^k) \geq C \cdot \|(I_n - X^k X^k^T) \nabla f(X^k)\|_F^2,$$

(11)
Theorem 3.2. Let $C = \frac{(\tau - 1 - \rho)}{2(\tau - 1 + \rho + \theta)^2}$.

Proof. By Lemma 2.2, we have

$$f(X^k) - f(\hat{X}_{GR}^k) \geq \frac{2(\tau - 1 - \rho)}{(\tau - 1 + \rho + \theta)^2} \cdot \|(I_n - X^k(X^k)^T)\nabla f(X^k)\|_F^2,$$

and

$$f(X^k) - f(\hat{X}_{GP}^k) \geq \frac{(\tau - 1 - \rho)}{2(\tau - 1 + \rho + \theta)^2} \cdot \|(I_n - X^k(X^k)^T)\nabla f(X^k)\|_F^2,$$

which implies that

$$f(X^k) - f(\bar{X}) \geq \min \left\{ \frac{2(\tau - 1 - \rho)}{(\tau - 1 + \rho + \theta)^2}, \frac{(\tau - 1 - \rho)}{2(\tau - 1 + \rho + \theta)^2} \right\} \cdot \|(I_n - X^k(X^k)^T)\nabla f(X^k)\|_F^2$$

$$= C \cdot \|(I_n - X^k(X^k)^T)\nabla f(X^k)\|_F^2,$$

where $C = \frac{(\tau - 1 - \rho)}{2(\tau - 1 + \rho + \theta)^2}$.

Now we are ready to show that our AG method is globally convergent.

Theorem 3.2. Let $\{X^k\}$ be the sequence generated by Algorithm 1. Then there exists a convergent subsequence of $\{X^k\}$. Moreover, each accumulation point of $\{X^k\}$ satisfies the first-order optimality condition of (1).

Proof. Notice that $\{X^k\}$ is bounded due to the feasibility of each iterate $X^k$. Hence it has a convergent subsequence. Without loss of generality, we assume that $X^k \to \bar{X}$ as $k \to \infty$. Due to the feasibility of $X^k$, $\bar{X}$ satisfies the feasibility condition in (8). Let $k = mh + j, h = 0, 1, 2, ..., j = 1, 2, ..., m$. From Lemmas 2.2 and 3.1, we have

$$f(X^{mh}) - f(X^{mh+1})$$

$$= \sum_{j=0}^{m-1} f(X^{mh+j}) - f(\hat{X}^{mh+j}) + f(\hat{X}^{mh+j}) - f(X^{mh+1+j})$$

$$\geq \sum_{j=0}^{m-1} C_{j+1} \|\nabla f(X^{mh+j}) - X^mh+j(X^mh+j)^T\nabla f(X^{mh+j})\|_F^2$$

$$+ \sum_{j=0}^{m-1} \frac{1}{8\theta + 1} \|(X^{mh+j})^T\nabla f(\hat{X}^{mh+j}) - \nabla f(\hat{X}^{mh+j})^T\hat{X}^{mh+j}\|_F^2$$

$$\geq C_1 \|\nabla f(X^{mh}) - X^mh(X^mh)^T\nabla f(X^{mh})\|_F^2,$$

where $C_{j+1}, j = 1, ..., m - 1$ are positive numbers, which implies

$$\lim_{h \to \infty} \|\nabla f(X^{mh}) - X^{mh}\nabla f(X^{mh})^T X^{mh}\|_F^2 = \|\nabla f(\bar{X}) - \bar{X}\nabla f(\bar{X})^T \bar{X}\|_F^2 = 0.$$
Corollary 3.3. Let \{\tilde{X}^k\} be the sequence generated by Algorithm 1. Then for any \( N \geq 1 \), we have

\[
\min_{k=1,\ldots,N} \| \tilde{X}^k - X^k \|_F^2 \leq \frac{f(X^0) - f(X^*)}{N \cdot \tilde{C}},
\]

where \( \tilde{C} = \frac{1 - \rho \tau}{2\tau} \).

Proof. From Lemma 2.4 and Lemma 3.1 of [8], we obtain

\[
f(X^k) - f(\tilde{X}^k) \geq \tilde{C} \cdot \| \tilde{X}^k - X^k \|_F^2,
\]

\[
f(X^{k+1}) \leq f(\tilde{X}^k) - \frac{1}{8\theta} \| (\tilde{X}^{k+1})^T \nabla f(\tilde{X}^k) - \nabla f(\tilde{X}^{k+1})^T \tilde{X}^k \|_F^2.
\]

where \( \tilde{C} = \frac{1 - \rho \tau}{2\tau} \), which implies

\[
f(X^k) \leq f(X^{k-1}) - \tilde{C} \cdot \| \tilde{X}^{k-1} - X^{k-1} \|_F^2
\]

\[
- \frac{1}{8\theta + 1} \| (\tilde{X}^{k-1})^T \nabla f(\tilde{X}^{k-1}) - \nabla f(\tilde{X}^{k-1})^T \tilde{X}^{k-1} \|_F^2.
\]

Let \( k = 1, \ldots, N \). We get

\[
f(X^N) \leq f(X^{N-1}) - \tilde{C} \cdot \| \tilde{X}^{N-1} - X^{N-1} \|_F^2
\]

\[
- \frac{1}{8\theta + 1} \| (\tilde{X}^{N-1})^T \nabla f(\tilde{X}^{N-1}) - \nabla f(\tilde{X}^{N-1})^T \tilde{X}^{N-1} \|_F^2,
\]

\[
\ldots
\]

\[
f(X^1) \leq f(X^0) - \tilde{C} \cdot \| \tilde{X}^0 - X^0 \|_F^2
\]

\[
- \frac{1}{8\theta + 1} \| (\tilde{X}^0)^T \nabla f(\tilde{X}^0) - \nabla f(\tilde{X}^0)^T \tilde{X}^0 \|_F^2.
\]

Summing up the above inequalities, we have

\[
f(X^N) \leq f(X^0) - \tilde{C} \cdot \sum_{k=1}^{N} \| \tilde{X}^k - X^k \|_F^2
\]

\[
- \frac{1}{8\theta + 1} \sum_{k=1}^{N} \| (\tilde{X}^k)^T \nabla f(\tilde{X}^k) - \nabla f(\tilde{X}^k)^T \tilde{X}^k \|_F^2.
\]

Noting that \( f(X^N) \geq f(X^*) \), we obtain

\[
\tilde{C} \cdot \sum_{k=1}^{N} \| \tilde{X}^k - X^k \|_F^2 \leq f(X^0) - f(X^*),
\]

which implies

\[
\min_{k=1,\ldots,N} \| \tilde{X}^k - X^k \|_F^2 \leq \frac{f(X^0) - f(X^*)}{N \cdot \tilde{C}}.
\]

This completes the proof. \( \square \)
4. Numerical experiments. In this section, we compare the numerical results of AG, GP [8], GR [8], GBB [16] and AFBB [10] methods. All experiment are performed in MATLAB R2014a under a Windows 7 operating system on an AMD A4-4335M APU at 1.90GHz and 2GB of RAM.

From [5] we know that it is usually difficult to obtain a good estimate of $\rho$, and $\rho^{-1}$ may be small, which would cause slow convergence. In our test, we use the alternate BB stepsize for $\tau$ to get good performance as suggested by [5]. Particularly, the update rule of $\tau$ can be described as follows:

$$
\tau = \begin{cases} 
\tau_{BB1}, & \text{mod}(k, 2) = 0, \\
\tau_{BB2}, & \text{mod}(k, 2) \neq 0,
\end{cases}
$$

where

$$
\tau_{BB1} = \frac{\langle J^{k-1}, J^{k-1} \rangle}{\langle J^{k-1}, K^{k-1} \rangle}, \quad \tau_{BB2} = \frac{\langle J^{k-1}, K^{k-1} \rangle}{\langle K^{k-1}, K^{k-1} \rangle},
$$

$$
J^{k-1} = X^k - X^{k-1}, \quad K^{k-1} = c(X^k) - c(X^{k-1}).
$$

Note that all the compared methods use the same stepsize rules as described above.

We use the same stopping criterion as in [8], i.e,

$$
\text{tol}_k^F = \frac{\|X^k - X^{k+1}\|_F}{\sqrt{n}} < \varepsilon_x, \quad (12)
$$

$$
\text{tol}_k^f = \frac{|f(X^k) - f(X^{k+1})|}{|f(X^k)| + 1} < \varepsilon_f, \quad (13)
$$

$$
\text{mean}([\text{tol}_{k-\text{min}(k,L)+1}^F, \ldots, \text{tol}_{k}^F]) < 10\varepsilon_x, \quad (14)
$$

$$
\text{mean}([\text{tol}_{k-\text{min}(k,L)+1}^f, \ldots, \text{tol}_{k}^f]) < 10\varepsilon_f, \quad (15)
$$

$$
\| (I_n - XX^T) \nabla f(X) \|_F < \varepsilon \| \nabla f(X^0) - X^0 \nabla f(X^0)^T X^0 \|_F. \quad (16)
$$

When one of the above criteria (12)-(16) or the number of iterations reaches maxiter, we terminate the algorithm. The parameters selection are $\varepsilon = 10^{-5}, \varepsilon_x = 10^{-8}, \varepsilon_f = 10^{-10}, L = 5,$ and maxiter= 5000.

Firstly, we consider some random generated problems in the form (2), where $A \in \mathbb{R}^{n \times n}$ and $G \in \mathbb{R}^{n \times p}$ are generated as follows:

$$
A = \frac{1}{2}(\bar{A} + \bar{A}^T), \quad G = \delta * QD.
$$

Here $Q_i = \frac{Q_i}{\|Q_i\|_2}, \bar{Q}_i = \text{rand}(n, 1), \ i = 1, \ldots, p,$ and $\bar{A} = \text{randn}(n)$ with $\text{rand}$ and $\text{randn}$ being MATLAB functions. The matrix $D \in \mathbb{R}^{p \times p}$ is diagonal with

$$
D_{jj} = \zeta^{j-1}(j = 1, \ldots, p),
$$

where we set $\zeta = 1.2$. The initial point is chosen by using the matrix QR decomposition function in MATLAB as $X^0 = q(r(\text{rand}(n, p))) \in \mathbb{R}^{n \times p}$.

Since the value of $m$ has a great influence on the performance of the AG method, we test it with different values of $m$. In particular, for $n = 500$, $p = 30$ and $\delta = 1$, we choose $m$ from $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15, 18, 20\}$. We see from Figure 1 that the AG method with $m = 2$ is faster than it with other values. So, we set $m = 2$ in our test.

Then, we compare AG with GBB, AFBB, GP and GR on problem (2) with $n = 1000$ and $\delta = 1$, and present the results in Table 2, where the notations are the same as those in Section 2. It can be seen that our AG method requires less
CPU time and number of iterations than GR and GP. Although our AG method takes more iterations than GBB and AFBB for a small value of $p$, it requires less CPU time. In addition, our AG method always outperforms GBB and AFBB when $p \geq 30$.

Table 3 presents results of the compared methods for the case $\delta = 10$ and $n = 1000$. Similarly as the above case, for most values of $p$, our AG method is better than other methods.

Secondly, we test Kohn-Sham total energy minimization problem with orthogonal constraints in the form (3). Our test results show that GP performs better than GR in the test, so we present the results of GP. We compare the numerical results of GP, GBB, AFBB, self consistent field (SCF) iteration in KSSOLV [18] and AG for several practical problems. The objective function and its gradient based on the MATLAB toolbox KSSOLV are employed. Due to the limited computer memory, we only test some medium scale concrete examples. We run SCF with maxiter=200 and other parameters taking their default values in KSSOLV, while GBB, AFBB and AG parameters selection are $\varepsilon = 10^{-5}, \varepsilon_x = 10^{-9}, \varepsilon_f = 10^{-13}$, maxiter=1000 and stopping rule is set as $\| (I_n - XX^T)H(X)X \|_F < \epsilon$ instead of (16). We set the same initial guess $X^0$ by using the function “genX0”, which is provided by KSSOLV. The test results are presented in Table 4. It can be seen that our AG method is comparable to other methods.

5. Conclusion. We proposed an alternate gradient (AG) method which alternately takes several gradient reflection steps followed by one gradient projection step. It was proved that any accumulation point of the iterations generated by our AG method satisfies the first-order optimal condition. Numerical results show that our alternate scheme is efficient.

Acknowledgments. The author would like to thank Dr. Bin Gao of ICTEAM institute for sharing his code GR and GP and thank Associate Professor Bo Jiang from Nanjing Normal University for sharing the code AFBB.
Table 2. Numerical results on problem (2) with $n = 1000$ and $\delta = 1$.

|     | cpu | iter | fun      | KKT violation | feasibility |
|-----|-----|------|----------|---------------|-------------|
| $p = 6$ |     |      |          |               |             |
| GBB | 9.474 | 734.2 | -131.5233622 | 2.6304E-04 | 1.7197E-15 |
| AFBB | 13.388 | 326.3 | -131.5249367 | 1.7790E-03 | 2.2349E-15 |
| GP  | 10.992 | 939.4 | -131.5225679 | 5.4438E-04 | 4.3864E-15 |
| GR  | 9.764 | 832.6 | -131.5214683 | 5.4198E-04 | 1.8122E-13 |
| AG  | 8.882 | 749.9 | -131.5221231 | 5.4445E-04 | 6.1051E-15 |
| $p = 10$ |     |      |          |               |             |
| GBB | 11.263 | 780.5 | -218.5897386 | 4.1310E-04 | 1.9883E-15 |
| AFBB | 18.575 | 527.3 | -218.5916071 | 3.2611E-03 | 2.9976E-15 |
| GP  | 15.173 | 1075.2 | -218.5856256 | 7.0932E-04 | 6.6001E-15 |
| GR  | 11.220 | 820.6 | -218.594843 | 7.0782E-04 | 2.3583E-13 |
| AG  | 9.986 | 706.5 | -218.5926144 | 7.0900E-04 | 8.0173E-15 |
| $p = 20$ |     |      |          |               |             |
| GBB | 14.344 | 825.5 | -478.6533636 | 6.3350E-04 | 1.8762E-15 |
| AFBB | 22.275 | 465.6 | -478.6490255 | 9.4387E-03 | 3.4080E-15 |
| GP  | 23.287 | 1251.6 | -478.6547359 | 1.1447E-03 | 1.2903E-13 |
| GR  | 15.833 | 915.2 | -478.6564912 | 1.1402E-03 | 6.0783E-13 |
| AG  | 15.426 | 872.9 | -478.6534441 | 1.1421E-03 | 1.6879E-14 |
| $p = 30$ |     |      |          |               |             |
| GBB | 31.84 | 1226.0 | -1245.746827 | 1.5488E-03 | 2.1434E-15 |
| AFBB | 48.08 | 661.4 | -1245.748650 | 1.8988E-02 | 5.0980E-15 |
| GP  | 21.86 | 784.0 | -1245.750942 | 8.4329E-05 | 1.5679E-14 |
| GR  | 17.06 | 656.8 | -1245.750942 | 7.1151E-05 | 8.0287E-13 |
| AG  | 16.24 | 602.0 | -1245.750942 | 6.7208E-05 | 1.5920E-14 |
| $p = 40$ |     |      |          |               |             |
| GBB | 61.73 | 2142.0 | -5639.853266 | 9.3744E-03 | 2.7954E-15 |
| AFBB | 87.39 | 853.6 | -5639.770839 | 1.2351E-01 | 5.1548E-15 |
| GP  | 29.81 | 861.2 | -5639.853800 | 7.2273E-05 | 1.9845E-14 |
| GR  | 24.26 | 693.2 | -5639.853800 | 5.9084E-05 | 7.8415E-13 |
| AG  | 21.82 | 654.0 | -5639.853800 | 5.7129E-05 | 1.9641E-14 |
| $p = 50$ |     |      |          |               |             |
| GBB | 160.670 | 4324.2 | -32893.22537 | 1.4305E-01 | 2.4661E-15 |
| AFBB | 195.783 | 902.0 | -32892.54444 | 6.1956E-01 | 7.0825E-15 |
| GP  | 10.488 | 234.8 | -32893.2292 | 1.3979E-01 | 2.3702E-14 |
| GR  | 10.261 | 253.2 | -32893.23767 | 1.3921E-01 | 7.1360E-13 |
| AG  | 8.226 | 189.8 | -32893.23383 | 1.3910E-01 | 1.6358E-13 |
| $p = 60$ |     |      |          |               |             |
| GBB | 181.231 | 4479.6 | -201185.9576 | 5.2503E-01 | 2.5522E-15 |
| AFBB | 220.369 | 889.4 | -201182.0639 | 4.6276E+00 | 7.5163E-15 |
| GP  | 3.832 | 71.0 | -201185.8551 | 8.6024E-01 | 2.5309E-14 |
| GR  | 5.578 | 122.2 | -201186.0137 | 7.7927E-01 | 6.1877E-13 |
| AG  | 3.139 | 63.5 | -201185.8558 | 8.6000E-01 | 1.6652E-13 |
Table 3. Numerical results on problem (2) with $n = 1000$ and $\delta = 10$.

| $p = 6$ | cpu | iter | fun   | KKT violation | feasibility |
|---------|-----|------|-------|---------------|-------------|
| GBB     | 2.350 | 193.7 | -162.4333197 | 5.1684E-04 | 1.5849E-15 |
| AFBB    | 4.323 | 168.4 | -162.3569464 | 8.9896E-02 | 1.0327E-14 |
| GP      | 9.625 | 833.8 | -162.4646836 | 6.9011E-04 | 4.5461E-15 |
| GR      | 7.432 | 653.8 | -162.4753778 | 6.8193E-04 | 2.0315E-13 |
| AG      | 6.443 | 572.0 | -162.4788103 | 6.8824E-04 | 5.2647E-15 |

| $p = 10$ | cpu | iter | fun   | KKT violation | feasibility |
|----------|-----|------|-------|---------------|-------------|
| GBB     | 2.412 | 212.8 | -162.445186  | 3.0936E-04 | 1.6845E-15 |
| AFBB    | 4.108 | 150.2 | -162.383351  | 9.9274E-02 | 2.3889E-15 |
| GP      | 10.028 | 892.1 | -162.445179  | 6.8857E-04 | 4.8457E-15 |
| GR      | 6.242 | 546.8 | -162.4768871 | 6.8015E-04 | 1.9881E-13 |
| AG      | 5.447 | 484.1 | -162.4667359 | 6.8719E-04 | 6.6787E-15 |

| $p = 20$ | cpu | iter | fun   | KKT violation | feasibility |
|----------|-----|------|-------|---------------|-------------|
| GBB     | 8.38  | 454.0 | -1504.964138 | 2.6761E-03 | 1.8293E-15 |
| AFBB    | 14.92 | 385.6 | -1504.918549 | 1.6780E-02 | 3.5068E-15 |
| GP      | 7.20  | 391.8 | -1504.918628 | 1.0776E-04 | 1.2022E-14 |
| GR      | 5.94  | 337.2 | -1504.918628 | 9.4064E-05 | 6.5906E-13 |
| AG      | 5.61  | 310.8 | -1504.918628 | 8.8531E-05 | 1.2883E-14 |

| $p = 30$ | cpu | iter | fun   | KKT violation | feasibility |
|----------|-----|------|-------|---------------|-------------|
| GBB     | 29.689 | 1243.9 | -8613.088591 | 2.0335E-02 | 2.2557E-15 |
| AFBB    | 43.338 | 565.2 | -8613.038104 | 1.4066E-01 | 5.1917E-15 |
| GP      | 10.754 | 401.0 | -8613.086436 | 3.6167E-02 | 1.5379E-14 |
| GR      | 6.938 | 275.4 | -8613.096615 | 3.6090E-02 | 8.1878E-13 |
| AG      | 6.811 | 259.9 | -8613.096568 | 3.6147E-02 | 8.1414E-14 |

| $p = 40$ | cpu | iter | fun   | KKT violation | feasibility |
|----------|-----|------|-------|---------------|-------------|
| GBB     | 66.861 | 2384.3 | -52514.12943 | 1.0842E-01 | 2.7336E-15 |
| AFBB    | 85.967 | 657.8 | -52513.74516 | 1.0337E+00 | 5.2309E-15 |
| GP      | 4.311 | 128.6 | -52514.11926 | 2.2352E-01 | 1.9552E-14 |
| GR      | 5.342 | 176.4 | -52514.09427 | 2.2260E-01 | 7.8898E-13 |
| AG      | 3.657 | 114.3 | -52514.08076 | 2.2395E-01 | 1.4507E-13 |

| $p = 50$ | cpu | iter | fun   | KKT violation | feasibility |
|----------|-----|------|-------|---------------|-------------|
| GBB     | 101.840 | 2667.2 | -323993.0909 | 5.2023E-01 | 2.4041E-15 |
| AFBB    | 131.354 | 735.0 | -323989.6253 | 4.9496E+00 | 7.1751E-15 |
| GP      | 2.297 | 45.4 | -323992.3883 | 1.3801E+00 | 2.1972E-14 |
| GR      | 2.527 | 59.2 | -323992.4399 | 1.3566E+00 | 8.8005E-13 |
| AG      | 1.867 | 39.8 | -323992.402  | 1.3720E+00 | 2.3366E-13 |
Table 4. Numerical results on Kohn-Sham total energy minimization problem.

|     | cpu  | iter | fun       | KKT violation | feasibility   |
|-----|------|------|-----------|---------------|---------------|
| cO2, n = 2103, p = 8 |      |      |           |               |               |
| SCF | 31.768 | 17  | -35.124396 | 1.5035E-06    | 6.5360E-15    |
| GBB | 32.818 | 53  | -35.124396 | 3.4202E-06    | 9.5236E-14    |
| AFBB| 35.728 | 61  | -35.124396 | 1.2646E-06    | 3.5113E-14    |
| GP  | 32.941 | 46  | -35.124396 | 9.5300E-06    | 3.8009E-15    |
| AG  | 28.840 | 44  | -35.124396 | 6.0064E-06    | 2.7404E-15    |
| c2H6, n = 2103, p = 7 |      |      |           |               |               |
| SCF | 28.294 | 18  | -14.420491 | 1.0304E-06    | 1.5356E-14    |
| GBB | 32.388 | 51  | -14.420491 | 9.7967E-06    | 8.0020E-14    |
| AFBB| 32.524 | 52  | -14.420491 | 4.7573E-06    | 1.2778E-14    |
| GP  | 32.879 | 50  | -14.420491 | 9.7401E-06    | 4.2016E-15    |
| AG  | 31.951 | 50  | -14.420491 | 5.9437E-06    | 4.2785E-15    |
| benzene, n = 8047, p = 15 |      |      |           |               |               |
| SCF | 601.321 | 118 | -37.225751 | 1.9756E-06    | 6.9301E-14    |
| GBB | 183.137 | 51  | -37.225751 | 7.4389E-06    | 8.7418E-14    |
| AFBB| 200.916 | 51  | -37.225751 | 2.0575E-06    | 1.7646E-13    |
| GP  | 207.944 | 50  | -37.225751 | 9.6828E-06    | 8.3590E-15    |
| AG  | 200.898 | 56  | -37.225751 | 4.9222E-06    | 9.8340E-15    |
| h2O, n = 2103, p = 4 |      |      |           |               |               |
| SCF | 20.570  | 26  | -16.440507 | 9.3011E-07    | 5.5665E-15    |
| GBB | 28.780  | 53  | -16.440507 | 9.4006E-06    | 1.8356E-14    |
| AFBB| 30.198  | 59  | -16.440507 | 4.8143E-07    | 1.9661E-14    |
| GP  | 26.768  | 51  | -16.440507 | 6.6455E-06    | 6.5909E-15    |
| AG  | 23.697  | 44  | -16.440507 | 8.0030E-06    | 4.8574E-14    |
| c12H26, n = 5709, p = 37 |      |      |           |               |               |
| SCF | 418.609 | 55  | -81.536092 | 3.9309E-06    | 3.8821E-14    |
| GBB | 283.230 | 66  | -81.536092 | 9.6402E-06    | 6.5371E-14    |
| AFBB| 289.654 | 61  | -81.536092 | 6.0986E-06    | 1.0709E-13    |
| GP  | 301.821 | 71  | -81.536092 | 5.9451E-06    | 1.4570E-14    |
| AG  | 261.392 | 60  | -81.536092 | 4.9262E-06    | 1.7237E-14    |
| sl2H4, n = 2103, p = 6 |      |      |           |               |               |
| SCF | 36.054  | 19  | -6.300975  | 1.5619E-06    | 8.8784E-15    |
| GBB | 43.816  | 70  | -6.300975  | 3.9777E-06    | 3.7868E-14    |
| AFBB| 44.630  | 69  | -6.300975  | 7.4737E-06    | 1.5903E-14    |
| GP  | 32.075  | 53  | -6.300975  | 9.6973E-06    | 3.9623E-14    |
| AG  | 36.636  | 62  | -6.300975  | 8.4565E-06    | 2.7022E-13    |
| nic, n = 251, p = 7 |      |      |           |               |               |
| SCF | 10.995  | 14  | -23.543530 | 1.2291E-06    | 3.2549E-15    |
| GBB | 10.030  | 45  | -23.543530 | 7.4993E-06    | 2.1795E-14    |
| AFBB| 9.038   | 45  | -23.543530 | 7.4993E-06    | 1.1159E-14    |
| GP  | 11.812  | 84  | -23.543530 | 1.0205E-06    | 1.9205E-15    |
| AG  | 11.162  | 82  | -23.543530 | 6.1287E-06    | 3.3939E-15    |
REFERENCES

[1] P. A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, Princeton, 2008.

[2] T. Abrudan, J. Eriksson and V. Koivunen, Steepest descent algorithms for optimization under unitary matrix constraint, IEEE Transactions on Signal Processing, 56 (2008), 1134–1147.

[3] J. Barzilai and J. M. Borwein, Two-point step size gradient methods, IMA Journal of Numerical Analysis, 8 (1988), 141–148.

[4] A. D’Aspremont, L. E. Ghaoui and J. G. R. G. Lanckriet, A direct formulation for sparse PCA using semidefinite programming, SIAM Review, 49 (2007), 434–448.

[5] Y. Dai and R. Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming, Numerische Mathematik, 100 (2005), 21–47.

[6] L. Eldén and H. Park, A procrustes problem on the Stiefel manifold, Numerische Mathematik, 82 (1999), 599–619.

[7] A. Edelman, T. A. Arias and S. T. Smith, The geometry of algorithms with orthogonality constraints, SIAM Journal on Matrix Analysis and Applications, 20 (1998), 303–353.

[8] B. Gao, X. Liu, X. Chen and Y. Yuan, A new first-order algorithmic framework for optimization problems with orthogonality constraints, SIAM Journal on Optimization, 28 (2018), 302–332.

[9] J. Hu, B. Jiang and X. Liu et al, A note on semidefinite programming relaxations for polynomial optimization over a single sphere, Science China Mathematics, 59 (2016), 1543–1560.

[10] B. Jiang and Y. Dai, A framework of constraint preserving update schemes for optimization on Stiefel manifold, Mathematical Programming, 153 (2015), 535–575.

[11] X. Liu, C. Hao and M. Cheng, A sequential subspace projection method for linear symmetric eigenvalue problem, Asia-Paciﬁc Journal of Operational Research, 30 (2013), 1340003.1–1340003.17.

[12] X. Liu, X. Wang and Z. Wen et al, On the convergence of the self-consistent field iteration in Kohn-Sham density functional theory, SIAM Journal on Matrix Analysis and Applications, 35 (2014), 546–558.

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

[14] A. Sameh and Z. Tong, The trace minimization method for the symmetric generalized eigenvalue problem, Journal of Computational and Applied Mathematics, 123 (2000), 155–175.

[15] M. Ulbrich, Z. Wen and C. Yang et al, A proximal gradient method for ensemble density functional theory, SIAM Journal on Scientiﬁc Computing, 37 (2015), A1975–A2002.

[16] Z. Wen and W. Yin, A feasible method for optimization with orthogonality constraints, Mathematical Programming, 142 (2013), 397–434.

[17] C. Yang, J. C. Meza and L. Wang, A constrained optimization algorithm for total energy minimization in electronic structure calculations, Journal of Computational Physics, 217 (2006), 709–721.

[18] C. Yang, J. C. Meza and B. Lee et al, KSSOLV-a MATLAB toolbox for solving the Kohn-Sham equations, ACM Transactions on Mathematical Software, 36 (2009), 1–35.

[19] H. Zhang and W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM Journal on Optimization, 14 (2004), 1043–1056.

[20] H. Zou, T. Hastie and R. Tibshirani, Sparse principal component analysis, Journal of Computational and Graphical Statistics, 15 (2006), 265–286.

[21] X. Zhang, J. Zhu and Z. Wen, Gradient type optimization methods for electronic structure calculations, SIAM Journal on Scientiﬁc Computing, 36 (2014), A265–A289.

Received November 2020; 1st revision December 2020; Final revision December 2020.

E-mail address: symsym2020@163.com
E-mail address: hyk@hebut.edu.cn