NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2015; **22**:283–298
Published online 30 July 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nla.1949

A sequential subspace projection method for extreme Z-eigenvalues of supersymmetric tensors

C. L. Hao[1], C. F. Cui[2] and Y. H. Dai[2,*,†]

[1]*Beijing University of Technology, Beijing 100124, P. R. China*
[2]*State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, P. R. China*

SUMMARY

Z-eigenvalues of tensors, especially extreme ones, are quite useful and are related to many problems, such as automatic control, quantum physics, and independent component analysis. For supersymmetric tensors, calculating the smallest/largest Z-eigenvalue is equivalent to solving a global minimization/maximization problem of a homogenous polynomial over the unit sphere. In this paper, we utilize the sequential subspace projection method (SSPM) to find extreme Z-eigenvalues and the corresponding Z-eigenvectors. The main idea of SSPM is to form a 2-dimensional subspace at the current point and then solve the original optimization problem in the subspace. SSPM benefits from the fact that the 2-dimensional subproblem can be solved by a direct method. Global convergence and linear convergence are established for supersymmetric tensors under certain assumptions. Preliminary numerical results over several testing problems show that SSPM is very promising. Besides, the globalization strategy of random phase can be easily incorporated into SSPM, which promotes the ability to find extreme Z-eigenvalues.

Received 30 December 2012; Revised 22 January 2014; Accepted 29 June 2014

KEY WORDS: extreme Z-eigenvalue; supersymmetric tensor; subspace; projection; global convergence; linear convergence

1. INTRODUCTION

Eigenvalues and eigenvectors of supersymmetric tensors were introduced by Qi [1] and Lim [2], independently. They form an important part in multilinear algebra and have found wide applications in magnetic resonance imaging [3], signal processing [4], quantum physics [5], independent component analysis [6, 7], etc.

This paper focuses on calculating Z-eigenvalues and Z-eigenvectors. Their properties were researched in [1, 2, 8, 9], and the number of E-eigenvalues of supersymmetric tensors was studied in [1, 9, 10]. Refer to [4, 11, 12] for more details.

There are many applications of Z-eigenvalues, especially the largest and smallest ones. As shown in [1], the largest magnitude Z-eigenvalue λ and its Z-eigenvector $\mathbf{x}$ form the best rank one approximation $\lambda\mathbf{x}^m$ of $\mathcal{A}$. The best rank one approximation of a supersymmetric tensor has many applications and has been studied in [13–17]. The smallest Z-eigenvalues have practical applications in determining positive definiteness of an even order supersymmetric tensor, which plays an important role in the diffusion tensor imaging [3] and the stability study of nonlinear autonomous system via Lyapunov's direct method in automatic control [18].

*Correspondence to: Y. H. Dai, State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, P. R. China.
†E-mail: dyh@lsec.cc.ac.cn

Consider an m-order n-dimensional real supersymmetric tensor $\mathcal{A}$

$$\mathcal{A} = (a_{i_1,\ldots,i_m}),\ a_{i_1,\ldots,i_m} \in \mathbb{R},\ 1 \leqslant i_1, \ldots, i_m \leqslant n,$$

which defines an m degree homogenous polynomial

$$\mathcal{A}\mathbf{x}^m := \sum_{i_1,\ldots,i_m=1}^{n} a_{i_1,i_2,\ldots,i_m} x_{i_1} x_{i_2} \cdots x_{i_m}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$. The n-dimensional column vector $\mathcal{A}\mathbf{x}^{m-1}$ is defined as

$$\left(\mathcal{A}\mathbf{x}^{m-1}\right)_i := \left(\sum_{i_2,\ldots,i_m=1}^{n} a_{i,i_2,\ldots,i_m} x_{i_2} \cdots x_{i_m} \right).$$

We say $\mathcal{A}$ is supersymmetric if its entries $a_{i_1,\ldots,i_m}$ are invariant under any permutation of their indices $i_1, \ldots, i_m$ [12]. $\mathcal{A}$ is said to be diagonal if all entries are zero except the ones with $i_1 = i_2 = \cdots = i_m$. All tensors, considered in this paper, are supersymmetric. $\mathcal{A}$ is called positive definite (positive semidefinite) if $\mathcal{A}\mathbf{x}^m > (\geqslant) 0$, for all $\mathbf{x} \in \mathbb{R}^n \backslash \{0\}$.

For a tensor $\mathcal{A}$, $\lambda \in \mathbb{R}$ is called a Z-eigenvalue of $\mathcal{A}$ [1], if there exists a vector $\mathbf{x} \in \mathbb{R}^n$ satisfying

$$\begin{aligned} \mathcal{A}\mathbf{x}^{m-1} &= \lambda\, \mathbf{x}, \\ \mathbf{x}^T\mathbf{x} &= 1, \end{aligned} \tag{1.1}$$

and $\mathbf{x}$ is called a Z-eigenvector of $\mathcal{A}$ associated with the Z-eigenvalue λ.

By the variational principle, any vector $\mathbf{x}$ satisfying (1.1) is a critical point of the polynomial optimization problem

$$\begin{aligned} \max \quad & \mathcal{A}\mathbf{x}^m \\ \text{s.t.} \quad & \mathbf{x}^T\mathbf{x} = 1, \end{aligned}$$

with λ being the corresponding multiplier.

From the geometrical point of view, a Z-eigenvalue of a tensor has similar properties with an eigenvalue of a matrix. For a symmetric positive definite matrix A, its largest eigenvalue is exactly the reciprocal of the square of shortest axis's length of the ellipsoid $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T A\mathbf{x} = 1\}$. In the tensor case, the distance from the origin to the set $\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid \mathcal{A}\mathbf{x}^m = 1\}$ is $\left(\frac{1}{\lambda_{\max}}\right)^{\frac{1}{m}}$, where $\lambda_{\max}$ is the largest Z-eigenvalue of $\mathcal{A}$.

Recently, some algorithms have been developed for calculating extreme Z-eigenvalues. Qi *et al.* [12] presented a direct method for the cases that $n = 2$ and $m = n = 3$. The power method was considered in [13] for computing the rank one approximation of higher-order tensors. The shifted power method [19] was guaranteed to converge to a tensor eigenpair. Han [20] proposed an unconstrained optimization approach for finding real eigenvalues of even order supersymmetric tensors. Nie *et al.* [17] studied semidefinite relaxations for the best rank one tensor approximation.

The purpose of this paper is to utilize the SSPM to calculate Z-eigenvalues of supersymmetric tensors. The framework of SSPM was firstly proposed by Dai [21] to develop algorithms for projection on an ellipsoid. The basic idea of SSPM is to, at each iteration, first construct a subspace by using the information at the current iteration and possibly the previous iterations and then solve the original problem in the subspace, which is an easy subproblem. The novel SSPM is employed to construct algorithms for linear eigenvalue problem [22]. In the tensor case, although the original problem (1.1) is difficult, the subproblem restricted to a 2-dimensional subspace is a 2-dimensional Z-eigenvalue problem, which can be solved by a direct method [12]. Hence, SSPM is a good choice for computing extreme Z-eigenvalues of supersymmetric tensors.

The rest of this paper is organized as follows. In the next section, we describe the framework of SSPM. The algorithm and its one variant with a random phase technique are presented in Section 3. Global convergence and linear convergence results are established in Section 4. In Section 5, we provide some numerical results for the proposed algorithms. Conclusions and discussions are drawn in the last section.

DOI: 10.1002/nla

2. SEQUENTIAL SUBSPACE PROJECTION METHOD

We shall describe the principle of SSPM in Subsection 2.1. Meanwhile, we address how to solve the subproblem in SSPM there. The geometrical explanations of Z-eigenvalues and SSPM are illuminated in Subsection 2.2.

2.1. Description of SSPM

Consider an n-dimensional supersymmetric tensor $\mathcal{A}$ with order m. The Z-eigenvalue λ and Z-eigenvector $\mathbf{x}$ of $\mathcal{A}$ satisfy (1.1). The problem of getting the largest Z-eigenvalue and its related Z-eigenvector is equivalent to solving the following global optimization problem:

$$\begin{aligned} \max \quad & \mathcal{A}\mathbf{x}^m \\ \text{s.t.} \quad & \mathbf{x}^T\mathbf{x} = 1. \end{aligned} \tag{2.2}$$

It is worth mentioning that the problem of finding the smallest Z-eigenvalue and the associated Z-eigenvector can be transformed into a global minimization problem.

The main idea of SSPM is that, at each iteration, a 2-dimensional subspace is constructed by some information of the current point, and a 2-dimensional subproblem can be derived by projecting the problem (2.2) on the subspace. SSPM benefits from the fact that the 2-dimensional subproblem can be solved by a direct method [12].

In details, at the k-th iteration, if $\mathbf{x}_k$ is not a Z-eigenvector of $\mathcal{A}$, we project the problem (2.2) on the 2-dimensional subspace

$$\mathcal{S}_k = \text{span}\{\mathbf{x}_k, \mathcal{A}\mathbf{x}_k^{m-1}\}.$$

Define the basis of $\mathcal{S}_k$ as $H_k = (\mathbf{p}_k, \mathbf{q}_k) \in \mathbb{R}^{n\times 2}$, where

$$\mathbf{p}_k = \frac{\mathbf{x}_k}{||\mathbf{x}_k||_2}, \quad \mathbf{z}_k = \mathcal{A}\mathbf{x}_k^{m-1} - \frac{\mathcal{A}\mathbf{x}_k^m}{\mathbf{x}_k^T\mathbf{x}_k}\mathbf{x}_k, \quad \mathbf{q}_k = \frac{\mathbf{z}_k}{||\mathbf{z}_k||_2}.$$

It is easy to see that $H_k^T H_k = I_2$, where I_2 is the 2-by-2 identity matrix, and that the subspace $\mathcal{S}_k = \{H_k\mathbf{l} : \mathbf{l} \in \mathbb{R}^2\}$. In this way, the 2-dimensional subproblem can be derived as

$$\begin{aligned} \max \ & \mathcal{A}\mathbf{x}^m \\ \text{s.t.} \ & \mathbf{x}^T\mathbf{x} = 1, \\ & \mathbf{x} \in \mathcal{S}_k := \{H_k\mathbf{l} : \mathbf{l} = (l_1, l_2)^T \in \mathbb{R}^2\}. \end{aligned} \tag{2.3}$$

Rewrite the object function as

$$\mathcal{A}\mathbf{x}^m = \mathcal{A}(H_k\mathbf{l})^m = \sum_{j=0}^{m} \binom{m}{j} \mathcal{A}\mathbf{p}_k^{m-j}\mathbf{q}_k^j l_1^{m-j} l_2^j := \mathcal{A}_k\mathbf{l}^m, \tag{2.4}$$

where $\binom{m}{j} = \frac{m!}{j!(m-j)!}$, and the constraints as

$$\mathbf{x}^T\mathbf{x} = \mathbf{l}^T H_k^T H_k\mathbf{l} = \mathbf{l}^T\mathbf{l} = 1.$$

The subproblem (2.3) can be rewritten as

$$\begin{aligned} \max \ & \mathcal{A}_k\mathbf{l}^m \\ \text{s.t.} \ & \mathbf{l}^T\mathbf{l} = 1. \end{aligned} \tag{2.5}$$

Notice that $\mathcal{A}_k$ has 2^m entries, which are determined by the following rule: if the number of 2 in $i_1, i_2, \ldots, i_m$ is j, then

$$(\mathcal{A}_k)_{i_1,i_2,\ldots,i_m} = \mathcal{A}\mathbf{p}_k^{m-j}\mathbf{q}_k^j, \ j = 0, \ldots, m. \tag{2.6}$$

The solution to (2.5), $(\tilde{\mathbf{l}}, \tilde{\lambda})$, can be obtained directly by Theorem 2.1, and then we set $\mathbf{x}_{k+1} = H_k\tilde{\mathbf{l}} = \tilde{l}_1\,\mathbf{p}_k + \tilde{l}_2\,\mathbf{q}_k$ and $\lambda_{k+1} = \tilde{\lambda}$.

The following theorem [12] provides a direct way to find the Z-eigenvalue of 2-dimensional supersymmetric tensors. It is used to calculate the largest or smallest Z-eigenvalue of $\mathcal{A}_k$ in SSPM.

DOI: 10.1002/nla

Theorem 2.1
([12]) Consider an m-order 2-dimensional supersymmetric tensor $\mathcal{T}$ with $\alpha_j = t_{i_1,i_2,\ldots,i_m}$ for $0 \leqslant j \leqslant m$, where $i_1 = \cdots = i_{m-j} = 1$, $i_{m-j+1} = \cdots = i_m = 2$. If $\alpha_1 := t_{1,\ldots,1,2} = 0$, then $\lambda = \alpha_0 = t_{1,\ldots,1}$ is the Z-eigenvalue of $\mathcal{T}$, and the associated Z-eigenvector is $\mathbf{x} = (1,0)^T$. The other Z-eigenvalues and Z-eigenvectors can be obtained by the following 1-dimensional equation of a:

$$\sum_{j=0}^{m-1} \binom{m-1}{j} \left[\alpha_j a^{m-j-1} - \alpha_{j+1} a^{m-j}\right] = 0. \tag{2.7}$$

More exactly, substituting a into the expressions below, the other Z-eigenvalues and Z-eigenvectors can be obtained by

$$\begin{aligned} x_1 &= \pm \frac{a}{\sqrt{1+a^2}}, \\ x_2 &= \pm \frac{1}{\sqrt{1+a^2}}, \\ \lambda &= \sum_{j=0}^{m} \binom{m}{j} \alpha_j x_1^{m-j} x_2^j. \end{aligned} \tag{2.8}$$

2.2. A geometrical view of sequential subspace projection method

In this subsection, we shall understand SSPM for even order positive symmetric tensors from the geometrical view of Z-eigenvalues. Letting $\mathbf{y} = \frac{\mathbf{x}}{\sqrt[m]{\lambda}}$, (1.1) can be transformed into

$$\begin{aligned} (\mathbf{y}^T\mathbf{y})^h &= \frac{1}{\lambda}, \\ \mathcal{A}\mathbf{y}^m &= 1. \end{aligned} \tag{2.9}$$

The problem (2.2) for the largest Z-eigenvalue can be transformed into the following optimization problem:

$$\begin{aligned} \min \quad & (\mathbf{y}^T\mathbf{y})^h \\ \text{s.t.} \quad & \mathcal{A}\mathbf{y}^m = 1. \end{aligned}$$

The coming theorem in [8] describes the geometrical properties of Z-eigenvalues and Z-eigenvectors, which supply the geometrical perspective of the newly proposed SSPM.

A general algebraic hypersurface can be represented by

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid \mathcal{A}\mathbf{x}^m = c\},$$

where $\mathbf{x} = (x_1, \ldots, x_n)^T$, $\mathcal{A}\mathbf{x}^m \in \mathbb{R}[x_1, \ldots, x_n]$ is a homogeneous polynomial of degree m with n variables, and $c \in \mathbb{R}$ is a real constant.

Theorem 2.2
([8]) We have the following conclusions on Z-eigenvalues and Z-eigenvectors of $\mathcal{A}$, Ω:

(a) Z-eigenvalue and Z-eigenvector of $\mathcal{A}$ and Ω exist;
(b) When m is even, the supersymmetric tensor $\mathcal{A}$ is positive definite (positive semidefinite) if and only if all the eigenvalues are positive (nonnegative);
(c) Assume that $c > 0$. Let the largest eigenvalue be denoted as $\lambda_{\max}$. If $\lambda_{\max} \leqslant 0$, then Ω is empty. Otherwise, Ω is not empty, and the distance from Ω to the origin is

$$\sigma_{\min} = \left(\frac{c}{\lambda_{\max}}\right)^{\frac{1}{m}},$$

and this distance occurs at the point $\mathbf{y} = \sigma_{\min}\mathbf{x}$, where $\mathbf{x}$ is an eigenvector associated with $\lambda_{\max}$;

DOI: 10.1002/nla

(d) Assume that $c > 0$. Denote the smallest eigenvalue as $\lambda_{\min}$. If $\mathcal{A}$ is positive definite, then the largest distance from Ω to the origin is

$$\sigma_{\max} = \left(\frac{c}{\lambda_{\min}}\right)^{\frac{1}{m}},$$

and this distance occurs at the point $\mathbf{y} = \sigma_{\max}\mathbf{x}$, where $\mathbf{x}$ is an eigenvector associated with $\lambda_{\min}$;

(e) Assume that $c > 0$. The algebraic hypersurface $\mathcal{Q}$ surrounds a bounded region if and only if m is even and all the eigenvalues are positive; namely, $\mathcal{A}$ is positive definite.

Consider the case (c) with $\lambda_{\max} > 0$, the problem of finding the largest Z-eigenvalue is equivalent to finding the distance d_0 from Ω to the origin. The idea of SSPM is to construct a sequence of 2-dimensional subspace S_k such that the distance from S_k to the origin is monotonically decreasing to d_0.

3. THE SEQUENTIAL SUBSPACE PROJECTION ALGORITHM

We first describe the sequential subspace projection algorithm in Subsection 3.1. Then a random phase technique is incorporated into SSPM to make the iterates escape from local minimizers /maximizers in Subsection 3.2.

3.1. The sequential subspace projection algorithm

At each iteration, a 2-dimensional subspace $\mathcal{S}_k$ is spanned by $\mathbf{x}_k$ and $\mathcal{A}\mathbf{x}_k^{m-1}$. The subproblem is derived by projecting (2.2) on $\mathcal{S}_k$ and solved by Theorem 2.1. The initial point is generated randomly. We stop the algorithm when $\mathcal{A}\mathbf{x}_k^{m-1}$ is parallel to $\mathbf{x}_k$, that is $(\mathcal{A}\mathbf{x}_k^{m-1})^T\mathbf{x}_k = \|\mathbf{x}_k\| \cdot \|\mathcal{A}\mathbf{x}_k^{m-1}\|$. In this paper, we use $\|\cdot\|$ to denote $\|\cdot\|_2$. The stopping criterion is set to be

$$\left|\frac{\mathcal{A}\mathbf{x}_k^m}{\|\mathbf{x}_k\| \cdot \|\mathcal{A}\mathbf{x}_k^{m-1}\|} - 1\right| \leqslant \epsilon, \tag{3.10}$$

where $\epsilon \geqslant 0$ is a tolerance parameter.

Algorithm 3.1
Given an initial point $\bar{\mathbf{x}}_0$, $\mathbf{x}_0 = \frac{\bar{\mathbf{x}}_0}{\|\bar{\mathbf{x}}_0\|}$, $\epsilon > 0$. Set $k = 0$.

Step 1. Let $\mathbf{p}_k = \mathbf{x}_k, \mathbf{z}_k = \mathcal{A}\mathbf{x}_k^{m-1} - \frac{\mathcal{A}\mathbf{x}_k^m}{\mathbf{x}_k^T\mathbf{x}_k}\mathbf{x}_k, \mathbf{q}_k = \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|}$. Compute $\mathcal{A}_k$ by (2.4).

Step 2. Solve the m-degree equation (2.7) and obtain the largest eigenvalue $\tilde{\lambda}$ and the associated eigenvector $\tilde{\mathbf{l}}$ with $\tilde{\lambda}$ by (2.8).

Step 3. Calculate $\mathbf{x}_{k+1} = \tilde{l}_1\,\mathbf{p}_k + \tilde{l}_2\,\mathbf{q}_k, \lambda_{k+1} = \tilde{\lambda}$.

Step 4. If (3.10) with k replaced by $k+1$ holds, stop and output $(\lambda_{k+1}, \mathbf{x}_{k+1})$; otherwise, let $k = k+1$ and go to Step 1.

Some remarks about Algorithm 3.1 are given in the following.

Remark 1
The aforementioned algorithm can also be employed to find the smallest Z-eigenvalue and the corresponding Z-eigenvector, if the smallest Z-eigenvalue of $\mathcal{A}_k$ is computed in Step 2.

Remark 2
The main storage cost of Algorithm 3.1 is n^m for the tensor $\mathcal{A}$. The main computational cost lies in the tensor-vector multiplications $\mathcal{A}\mathbf{x}^{m-1}$ and $\mathcal{A}_k$ in Step 1. For an m-order n-dimensional supersymmetric tensor $\mathcal{A}$, it costs $O(mn^m)$ operations to compute $\mathcal{A}\mathbf{x}^{m-1}$. Then the value of $\mathcal{A}\mathbf{x}^m$ can be obtained by multiplying the vectors $\mathcal{A}\mathbf{x}^{m-1}$ and $\mathbf{x}$, which takes only $2n-1$ operations. Computing $\mathcal{A}_k$ is more expensive, which requires $O(m^2n^m)$ operations. Compared with SSPM, the power method requires mainly to compute $\mathcal{A}\mathbf{x}^{m-1}$. Our numerical experiments with $m = 4$ in Section 5

DOI: 10.1002/nla

show that the computational cost of SSPM per iteration is about three to four times of the power method.

Remark 3
Up to now, SSPM is designed to handle fully populated supersymmetric tensors, the size of which is n^m. Hence, we can only solve the problems with $n \leqslant 100$ in a personal computer for the sake of storage. In the case of large mode size n, it is a good future work to consider how to make SSPM work well with the idea of the multigrid accelerated alternating least squares type algorithm [23].

3.2. Sequential subspace projection method with random phase

In our numerical experiments, we find that (3.1) may stick in some local minimizers/maximizers. In this subsection, SSPM with random phase (SSPMr) is proposed by introducing a random phase in order to escape from local minimizers/maximizers. In details, after some criterion is satisfied, we switch to construct the subspace by $\mathcal{S}_k = \text{span}\{\mathbf{x}_k, \mathbf{u}\}$, where $\mathbf{u}$ is a unit vector generated randomly, instead of $\mathcal{S}_k = \text{span}\{\mathbf{x}_k, \mathcal{A}\mathbf{x}_k^{m-1}\}$ in SSPM. Moreover, an upper bound is set for the total number of switching to the random phase.

Algorithm 3.2
Prefix an initial point $\bar{\mathbf{x}}_0$, $\mathbf{x}_0 = \frac{\bar{\mathbf{x}}_0}{\|\bar{\mathbf{x}}_0\|}$, $\epsilon > 0$, and set $k = 0$, $\lambda_k = \mathcal{A}\mathbf{x}_k^m$, $\mathbf{u} = \mathcal{A}\mathbf{x}_k^{m-1}$, $F = 1 - \frac{\lambda_k}{\|\mathbf{u}\|}$, $flagmax = 20$, $flag = 0$.

Step 1. Let $\mathbf{p}_k = \mathbf{x}_k, \mathbf{z}_k = \mathbf{u} - (\mathbf{u}^T\mathbf{x}_k)\mathbf{x}_k, \mathbf{q}_k = \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|}$. Compute $\mathcal{A}_k$ by (2.4).
Step 2. Solve the m-degree equation (2.7) and obtain the largest eigenvalue $\tilde{\lambda}$ and the associated eigenvector $\tilde{\mathbf{l}}$ with $\tilde{\lambda}$ by (2.8).
Step 3. Calculate $\mathbf{x}_{k+1} = \tilde{l}_1\,\mathbf{p}_k + \tilde{l}_2\,\mathbf{q}_k$, $\lambda_{k+1} = \tilde{\lambda}$.
Step 4. Compute $\mathbf{u} = \mathcal{A}\mathbf{x}_{k+1}^{m-1}$, $F = 1 - \frac{\lambda_{k+1}}{\|\mathbf{u}\|}$.
If $F > \epsilon$,
if $flag = 0$, $\mathbf{x}_k = \mathbf{x}_{k+1}$, $\lambda_k = \lambda_{k+1}$,
else if $\lambda_{k+1} - \lambda_k \geqslant 10^{-6}$, $flag = 0$, $\mathbf{x}_k = \mathbf{x}_{k+1}$, $\lambda_k = \lambda_{k+1}$,
else $flag = flag + 1$, $\mathbf{u} = rand(n, 1)$, $\frac{\mathbf{u}}{\|\mathbf{u}\|}$.
$k = k + 1$, go to step 1,
else
if $flag = 0$, set $flag = 1$, $\mathbf{u} = rand(n, 1)$ and $\mathbf{u} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$.
else if $0 < flag < flagmax$
if $\lambda_{k+1} - \lambda_k \geqslant 10^{-6}$, $flag = 0$, $\mathbf{x}_k = \mathbf{x}_{k+1}$, $\lambda_k = \lambda_{k+1}$.
else $flag = flag + 1$, $\mathbf{u} = rand(n, 1)$, $\mathbf{u} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$.
$k = k + 1$, go to step 1.
else stop and output $(\lambda_{k+1}, \mathbf{x}_{k+1})$.

4. CONVERGENCE ANALYSIS

In this section, we show the global convergence of Algorithm 3.1 and the linear convergence of the objective values $\{\mathcal{A}\mathbf{x}_k\}$ under some assumptions.

Noticing that at each iteration, the current point $\mathbf{x}_k$ belongs to the subspace $\mathcal{S}_k = \text{span}\{\mathbf{x}_k, \mathcal{A}\mathbf{x}_k^{m-1}\}$, we have $f(\mathbf{x}_{k+1}) = \max_{\mathbf{s}\in\mathcal{S}_k} f(\mathbf{s}) \geqslant f(\mathbf{x}_k)$; namely, Algorithm 3.1 is monotone increasing. For convenience, rewrite (2.2) as

$$\begin{aligned} \max \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & c(\mathbf{x}) = 0, \end{aligned} \tag{4.11}$$

where

$$f(\mathbf{x}) = \frac{1}{m}\mathcal{A}\mathbf{x}^m \quad \text{and} \quad c(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}^T\mathbf{x} - 1\right).$$

DOI: 10.1002/nla

The Lagrangian function is

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda\, c(\mathbf{x}),$$

and its gradient and Hessian are

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = \nabla f(\mathbf{x}) - \lambda\, \mathbf{x} \quad \text{and} \quad \nabla_{\mathbf{xx}} L(\mathbf{x}, \lambda) = \nabla^2 f(\mathbf{x}) - \lambda I_n.$$

We use $\nabla L(\mathbf{x}, \lambda)$ and $\nabla^2 L(\mathbf{x}, \lambda)$ to represent $\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda)$ and $\nabla_{\mathbf{xx}} L(\mathbf{x}, \lambda)$ unless specifically stated. Noting that at the KKT point $(\mathbf{x}^*, \lambda^*)$, it holds $\nabla L(\mathbf{x}^*, \lambda^*) = 0$. Therefore, $\nabla L(\mathbf{x}^*, \lambda^*)^T \mathbf{x} = \nabla f(\mathbf{x}^*)^T \mathbf{x}^* - \lambda^* = 0$; namely, $\lambda^* = \nabla f(\mathbf{x}^*)^T \mathbf{x}^*$. At the k-th iteration, we define $\lambda_k := \nabla f(\mathbf{x}_k)^T \mathbf{x}_k$ and

$$\mathbf{z}_k := \nabla f(\mathbf{x}_k) - \left(\nabla f(\mathbf{x}_k)^T \mathbf{x}_k\right) \mathbf{x}_k = \nabla f(\mathbf{x}_k) - \lambda_k \mathbf{x}_k. \tag{4.12}$$

It is easy to see that $\mathbf{z}_k = 0$ if and only if $\nabla L(\mathbf{x}_k, \lambda_k) = 0$. For the problem (4.11), we have $\nabla f(\mathbf{x}) = \mathcal{A}\mathbf{x}^{m-1}$. By definition, if $\mathbf{z}_k = 0$, $\mathbf{x}_k$ is a Z-eigenvector and $\lambda_k = \mathbf{x}_k^T \mathcal{A}\mathbf{x}_k^{m-1} = \mathcal{A}\mathbf{x}_k^m$ is the corresponding Z-eigenvalue.

Assumption 4.1
$\nabla f(\mathbf{x})$ is bounded and Lipschitz continuous on the unit sphere; namely, there exist positive constants L and M such that

$$\begin{aligned} \|\nabla f(\mathbf{x})\| &\leqslant M, && \forall\ \mathbf{x}^T\mathbf{x} = 1; \\ \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| &\leqslant L\|\mathbf{x} - \mathbf{y}\|, && \forall\ \mathbf{x}^T\mathbf{x} = \mathbf{y}^T\mathbf{y} = 1. \end{aligned}$$

These assumptions are trivial for polynomials. The global convergence of Algorithm 3.1 is established in the following lemma.

Lemma 4.2
Consider the problem (4.11). Under Assumption 4.1, for the sequence $\{\mathbf{x}_k\}$ generated by Algorithm 3.1, there exist positive constants c_1 and c_2 such that

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \geqslant \min\{c_1, c_2\|\mathbf{z}_k\|^2\}, \tag{4.13}$$

where $c_1 = \frac{3}{4}(M + L)$, $c_2 = \frac{1}{4}(M + L)^{-1}$. Furthermore,

$$\|\mathbf{z}^*\| := \lim_{k\to\infty} \|\mathbf{z}_k\| = 0. \tag{4.14}$$

That is to say, $\{\mathbf{x}_k\}$ converges to the Karush–Kuhn–Tucker (KKT) point of the problem (4.11) globally.

Proof
Noting that $\mathbf{z}_k$ lies in the subspace $\text{span}\{\mathbf{x}_k, \mathcal{A}\mathbf{x}_k^{m-1}\}$ and $\mathbf{z}_k^T\mathbf{x}_k = 0$, every feasible point of the problem (2.3) can be expressed by

$$\mathbf{x}_k(\alpha) = \alpha\mathbf{z}_k \mathbf{p} m \sqrt{1 - \alpha^2\|\mathbf{z}_k\|^2}\, \mathbf{x}_k,\ -\frac{1}{\|\mathbf{z}_k\|} \leqslant \alpha \leqslant \frac{1}{\|\mathbf{z}_k\|}.$$

It follows that

$$\mathbf{x}_k'(\alpha) = \mathbf{z}_k \mp \frac{\alpha\|\mathbf{z}_k\|^2}{\sqrt{1 - \alpha^2\|\mathbf{z}_k\|^2}}\mathbf{x}_k.$$

We shall study the gap between $f(\mathbf{x}_k(\alpha))$ and $f(\mathbf{x}_k)$:

$$\begin{aligned} f(\mathbf{x}_k(\alpha)) - f(\mathbf{x}_k) &= \int_0^\alpha \nabla f(\mathbf{x}_k(t))^T \mathbf{x}_k'(t)\, dt \\ &= \int_0^\alpha \nabla f(\mathbf{x}_k(t))^T \left[\mathbf{x}_k'(t) - \mathbf{x}_k'(0)\right] dt + \alpha \nabla f(\mathbf{x}_k(0))^T \mathbf{x}_k'(0) \\ &\quad + \int_0^\alpha \left(\nabla f(\mathbf{x}_k(t)) - \nabla f(\mathbf{x}_k(0))\right)^T \mathbf{x}_k'(0)\, dt \\ &= T_1 + \alpha\|\mathbf{z}_k\|^2 + T_2, \end{aligned} \tag{4.15}$$

DOI: 10.1002/nla

where the third equality is due to the definitions of $\nabla f(\mathbf{x}_k(0))$ and $\mathbf{x}'_k(0)$. Next, we estimate the term T_1 for $\alpha \in [0, \tilde{\alpha}]$, $\tilde{\alpha} = \frac{\sqrt{3}}{2\|\mathbf{z}_k\|}$.

$$\begin{aligned} T_1 &= \int_0^\alpha \nabla f(\mathbf{x}_k(t))^T [\mathbf{x}'_k(t) - \mathbf{x}'_k(0)]\, dt \\ &\geqslant -\int_0^\alpha \|\nabla f(\mathbf{x}_k(t))\| \cdot \|\mathbf{x}'_k(t) - \mathbf{x}'_k(0)\|\, dt \\ &\geqslant -M\|\mathbf{z}_k\|^2 \int_0^\alpha \frac{t}{\sqrt{1 - t^2\|\mathbf{z}_k\|^2}}\, dt \\ &\geqslant -M\alpha^2\|\mathbf{z}_k\|^2, \end{aligned} \tag{4.16}$$

where the last inequality follows from $\frac{1}{\sqrt{1-t^2\|\mathbf{z}_k\|^2}} \leqslant \frac{1}{\sqrt{1-\tilde{\alpha}^2\|\mathbf{z}_k\|^2}} = 2$ for $t \leqslant \alpha \leqslant \tilde{\alpha}$. For the third term T_2, by Assumption 4.1, we can derive

$$\begin{aligned} T_2 &= \int_0^\alpha (\nabla f(\mathbf{x}_k(t)) - \nabla f(\mathbf{x}_k(0)))^T \mathbf{x}'_k(0)\, dt \\ &\geqslant -\|\mathbf{z}_k\| \int_0^\alpha \|\nabla f(\mathbf{x}_k(t)) - \nabla f(\mathbf{x}_k(0))\|\, dt \\ &\geqslant -L\|\mathbf{z}_k\| \int_0^\alpha \|\mathbf{x}_k(t) - \mathbf{x}_k(0)\|\, dt \\ &= -L\|\mathbf{z}_k\| \int_0^\alpha \|t\mathbf{z}_k + \left(\pm\sqrt{1 - t^2\|\mathbf{z}_k\|^2} - 1\right)\mathbf{x}_k\|\, dt \\ &= -L\|\mathbf{z}_k\| \int_0^\alpha \sqrt{2 \pm 2\sqrt{1 - t^2\|\mathbf{z}_k\|^2}}\, dt \\ &\geqslant -L\alpha^2\|\mathbf{z}_k\|^2 \end{aligned} \tag{4.17}$$

where the last equality can be derived by $\mathbf{x}_k^T\mathbf{z}_k = 0$, $\|\mathbf{x}_k\| = 1$, and the last inequality is due to $\sqrt{2 \pm 2\sqrt{1 - t^2\|\mathbf{z}_k\|^2}} \leqslant 2t\|\mathbf{z}_k\|$ for $t \leqslant \alpha \leqslant \tilde{\alpha}$.

It follows from (4.15), (4.16), and (4.17) that

$$\begin{aligned} f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) &= \max_{\alpha^2 \leqslant 1/\|\mathbf{z}_k\|^2} (f(\mathbf{x}_k(\alpha)) - f(\mathbf{x}_k)) \\ &\geqslant \max_{0 \leqslant \alpha \leqslant \tilde{\alpha}} (f(\mathbf{x}_k(\alpha)) - f(\mathbf{x}_k)) \\ &\geqslant \max_{0 \leqslant \alpha \leqslant \tilde{\alpha}} (-M\alpha^2\|\mathbf{z}_k\|^2 - L\alpha^2\|\mathbf{z}_k\|^2 + \alpha\|\mathbf{z}_k\|^2) \\ &\geqslant \max_{0 \leqslant \alpha \leqslant \tilde{\alpha}} (-M\alpha^2 - L\alpha^2 + \alpha)\|\mathbf{z}_k\|^2. \end{aligned}$$

Denote $\hat{\alpha} = \frac{1}{2M+2L}$ as the maximizer of the quadratic function above. If $\tilde{\alpha}^2 \leqslant \hat{\alpha}^2$, that is $\|\mathbf{z}_k\| \geqslant \sqrt{3}(M + L)$, the maximum will be achieved at $\tilde{\alpha}$:

$$\begin{aligned} f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) &\geqslant (-M\tilde{\alpha}^2 - L\tilde{\alpha}^2 + \tilde{\alpha})\|\mathbf{z}_k\|^2 \\ &\geqslant \frac{\sqrt{3}}{4}\|\mathbf{z}_k\| \\ &\geqslant \frac{3}{4}(M + L). \end{aligned} \tag{4.18}$$

Otherwise, if $\|\mathbf{z}_k\| \leqslant \sqrt{3}(M + L)$, we can obtain

$$\begin{aligned} f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) &\geqslant (-M\hat{\alpha}^2 - L\hat{\alpha}^2 + \hat{\alpha})\|\mathbf{z}_k\|^2 \\ &= \frac{1}{4(M + L)}\|\mathbf{z}_k\|^2. \end{aligned} \tag{4.19}$$

DOI: 10.1002/nla

If the sequence $\{\mathbf{z}_k\}$ is infinite, the ascent can not be greater than a constant because $f(\mathbf{x})$ must be bounded on the unit sphere. This means that the relation (4.18) will not occur but the relation (4.19) holds for sufficiently large k. Thus, there exists an integer N_0 such that for all $k \geqslant N_0$,

$$\begin{aligned} f(\mathbf{x}_{k+1}) - f(\mathbf{x}_{N_0}) &\geqslant \sum_{i=N_0}^{k} (f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)) \\ &\geqslant \frac{1}{4(M+L)} \sum_{i=N_0}^{k} \|\mathbf{z}_k\|^2. \end{aligned}$$

As $f(\mathbf{x})$ must be bounded on the unit sphere, $\lim_{k\to\infty} \|\mathbf{z}_k\| = 0$; namely, the sequence is globally convergent. □

In order to obtain the linear convergence, we make the following assumption.

Assumption 4.3
Assume that the sequence $\{\mathbf{x}_k\}$ tends to a point $\mathbf{x}^*$ and that the second order sufficient condition for $\mathbf{x}^*$ being a strict local maximizer of (4.11) is satisfied; namely, the corresponding Lagrangian multiplier λ^* and Lagrangian function $L(\mathbf{x}^*, \lambda^*)$ satisfy

$$\mathbf{w}^T \nabla^2 L(\mathbf{x}^*, \lambda^*)\mathbf{w} < 0, \ \forall\ \mathbf{w} \in \nabla c(\mathbf{x}^*)^{\perp} \cap \{\mathbf{w} : c(\mathbf{w}) = 0\}. \tag{4.20}$$

Lemma 4.4
Suppose Assumptions 4.1 and 4.3 hold for the problem (4.11). The sequence $\{\mathbf{x}_k\}$ generated by Algorithm 3.1 satisfies

$$\lim_{k\to\infty} \frac{\|\mathbf{z}_k\|^2}{f(\mathbf{x}^*) - f(\mathbf{x}_k)} \geqslant 2\mu_{n-1}, \tag{4.21}$$

where $\mu_{n-1} > 0$ is the smallest eigenvalue of the matrix $-U^T \nabla^2 L(\mathbf{x}^*, \lambda^*)U$, $U \in \mathbb{R}^{n\times(n-1)}$ is the orthogonal complement matrix of $\nabla c(\mathbf{x}^*)$.

Proof
First, we deduce the fact that $f(\mathbf{x}^*) - f(\mathbf{x}_k) > 0$. By $c(\mathbf{x}_k) = c(\mathbf{x}^*) = 0$ and $\nabla L(\mathbf{x}^*, \lambda^*) = \mathbf{z}^* = 0$,

$$\begin{aligned} f(\mathbf{x}^*) - f(\mathbf{x}_k) &= L(\mathbf{x}^*, \lambda^*) - L(\mathbf{x}_k, \lambda_k) \\ &= -\frac{1}{2}(\mathbf{x}_k - \mathbf{x}^*)^T \nabla^2 L(\mathbf{x}^*, \lambda^*)(\mathbf{x}_k - \mathbf{x}^*) + o(\|\mathbf{x}_k - \mathbf{x}^*\|^2). \end{aligned} \tag{4.22}$$

For the problem (4.11), $\mathbf{x}_k^T \mathbf{x}_k = \mathbf{x}^{*T} \mathbf{x}^* = 1$, thus

$$-(\mathbf{x}_k - \mathbf{x}^*)^T \mathbf{x}^* = 1 - \mathbf{x}_k^T \mathbf{x}^* = \frac{1}{2}(\mathbf{x}_k - \mathbf{x}^*)^T (\mathbf{x}_k - \mathbf{x}^*), \tag{4.23}$$

which means that if we project $\mathbf{x}_k - \mathbf{x}^*$ on the orthogonal space of $\mathbf{x}^*$,

$$\mathbf{w}_k = \mathbf{x}_k - \mathbf{x}^* - \left((\mathbf{x}_k - \mathbf{x}^*)^T \mathbf{x}^*\right) \mathbf{x}^* = \mathbf{x}_k - \mathbf{x}^* + O(\|\mathbf{x}_k - \mathbf{x}^*\|^2). \tag{4.24}$$

It follows from the second order sufficient condition (4.20) that

$$\begin{aligned} & -\frac{1}{2}(\mathbf{x}_k - \mathbf{x}^*)^T \nabla^2 L(\mathbf{x}^*, \lambda^*)(\mathbf{x}_k - \mathbf{x}^*) \\ &= -\frac{1}{2}(\mathbf{w}_k + O(\|\mathbf{x}_k - \mathbf{x}^*\|^2))^T \nabla^2 L(\mathbf{x}^*, \lambda^*)(\mathbf{w}_k + O(\|\mathbf{x}_k - \mathbf{x}^*\|^2)) \\ &= -\frac{1}{2}\mathbf{w}_k^T \nabla^2 L(\mathbf{x}^*, \lambda^*)\mathbf{w}_k + o(\|\mathbf{x}_k - \mathbf{x}^*\|^2) \\ &> 0, \end{aligned}$$

which with (4.22) shows $f(\mathbf{x}^*) - f(\mathbf{x}_k) > 0$.

DOI: 10.1002/nla

Second, the approximation of $\mathbf{z}_k$ is required. It follows from $\mathbf{z}^* = 0$ in (4.14) that

$$\begin{aligned}\mathbf{z}_k &= \mathbf{z}_k - \mathbf{z}^* \\ &= \nabla f(\mathbf{x}_k) - \lambda_k \mathbf{x}_k - \nabla f(\mathbf{x}^*) + \lambda^* \mathbf{x}^* \\ &= \nabla^2 f(\mathbf{x}^*)(\mathbf{x}_k - \mathbf{x}^*) - \lambda^*(\mathbf{x}_k - \mathbf{x}^*) - (\lambda_k - \lambda^*)\mathbf{x}_k + O(\|\mathbf{x}_k - \mathbf{x}^*\|^2) \\ &= \nabla^2 L(\mathbf{x}^*, \lambda^*)(\mathbf{x}_k - \mathbf{x}^*) - (\lambda_k - \lambda^*)\mathbf{x}_k + O(\|\mathbf{x}_k - \mathbf{x}^*\|^2),\end{aligned} \tag{4.25}$$

where the last equality follows from the definition $\nabla^2 L(\mathbf{x}^*, \lambda^*) = \nabla^2 f(\mathbf{x}^*) - \lambda^* I_n$. Consider the term $\lambda_k - \lambda^*$, by the Taylor expansion,

$$\begin{aligned}\lambda_k - \lambda^* &= \nabla f(\mathbf{x}_k)^T \mathbf{x}_k - \nabla f(\mathbf{x}^*)^T \mathbf{x}^* \\ &= \left(\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^*)\right)^T \mathbf{x}_k + \nabla f(\mathbf{x}^*)^T (\mathbf{x}_k - \mathbf{x}^*) \\ &= (\mathbf{x}_k - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)\mathbf{x}_k + O(\|\mathbf{x}_k - \mathbf{x}^*\|^2) + \lambda^* \mathbf{x}^{*T}(\mathbf{x}_k - \mathbf{x}^*) \\ &= O(\|\mathbf{x}_k - \mathbf{x}^*\|^2),\end{aligned} \tag{4.26}$$

where the third equality is obtained by $\mathbf{z}^* = \nabla f(\mathbf{x}^*) - \lambda^* \mathbf{x}^* = 0$, and the last equality follows from $(\mathbf{x}_k - \mathbf{x}^*)^T \nabla^2 f(\mathbf{x}^*)\mathbf{x}_k \leqslant M(\mathbf{x}_k - \mathbf{x}^*)^T \mathbf{x}_k$, $(\mathbf{x}_k - \mathbf{x}^*)^T \mathbf{x}_k = \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}^*\|^2$ and (4.23). It follows from (4.24), (4.25), and (4.26),

$$\begin{aligned}\|\mathbf{z}_k\|^2 &= \|\nabla^2 L(\mathbf{x}^*, \lambda^*)(\mathbf{x}_k - \mathbf{x}^*) + O(\|\mathbf{x}_k - \mathbf{x}^*\|^2)\|^2 \\ &= \|\nabla^2 L(\mathbf{x}^*, \lambda^*)(\mathbf{x}_k - \mathbf{x}^*)\|^2 + o(\|\mathbf{x}_k - \mathbf{x}^*\|^2) \\ &= \|\nabla^2 L(\mathbf{x}^*, \lambda^*)\mathbf{w}\|^2 + o(\|\mathbf{x}_k - \mathbf{x}^*\|^2).\end{aligned} \tag{4.27}$$

Let $U \in \mathbb{R}^{n\times(n-1)}$ be the orthogonal complement of the vector $\nabla c(\mathbf{x}^*)$. For any $\mathbf{w}$ defined in (4.24), there exists $\mathbf{v} \in \mathbb{R}^{n-1}$ such that $\mathbf{w} = U\mathbf{v}$. It follows from (4.22) and (4.27) that

$$\begin{aligned}\frac{\|\mathbf{z}_k\|^2}{f(\mathbf{x}^*) - f(\mathbf{x}_k)} &= \frac{\|\nabla^2 L(\mathbf{x}^*, \lambda^*)(\mathbf{x}_k - \mathbf{x}^*)\|^2}{-\frac{1}{2}(\mathbf{x}_k - \mathbf{x}^*)^T \nabla^2 L(\mathbf{x}^*, \lambda^*)(\mathbf{x}_k - \mathbf{x}^*)} \\ &= \frac{\|\nabla^2 L(\mathbf{x}^*, \lambda^*)\mathbf{w}\|^2}{-\frac{1}{2}\mathbf{w}^T \nabla^2 L(\mathbf{x}^*, \lambda^*)\mathbf{w}} \\ &= \frac{2\|\nabla^2 L(\mathbf{x}^*, \lambda^*)U\mathbf{v}\|^2}{-\mathbf{v}^T U^T \nabla^2 L(\mathbf{x}^*, \lambda^*)U\mathbf{v}} \\ &= \frac{2\|U\|\|\nabla^2 L(\mathbf{x}^*, \lambda^*)U\mathbf{v}\|^2}{-\mathbf{v}^T U^T \nabla^2 L(\mathbf{x}^*, \lambda^*)U\mathbf{v}} \\ &\geqslant \frac{2\|U^T \nabla^2 L(\mathbf{x}^*, \lambda^*)U\mathbf{v}\|^2}{-\mathbf{v}^T U^T \nabla^2 L(\mathbf{x}^*, \lambda^*)U\mathbf{v}} \\ &\geqslant 2\mu_{n-1},\end{aligned}$$

where the fourth equality is derived from $\|B\|_2 = \sqrt{\|B^T B\|} = 1$ and the first inequality follows from the matrix compatibility. For the last inequality, suppose the Cholesky decomposition of the positive definite matrix $-U^T \nabla^2 L(\mathbf{x}^*, \lambda^*)U$ is $L^T L$. Denoting $\mathbf{y} = L\mathbf{x}$, we can obtain $\frac{\mathbf{x}^T P P \mathbf{x}}{\mathbf{x}^T P \mathbf{x}} = \frac{\mathbf{y}^T P \mathbf{y}}{\mathbf{y}^T \mathbf{y}} \geqslant \mu_{n-1}$. □

Theorem 4.5
If Assumptions 4.1 and 4.3 hold for the problem (4.11), the sequence $\{\mathcal{A}\mathbf{x}_k^m, \mathbf{x}_k\}$ generated by Algorithm 3.1 convergent to $(\lambda^*, \mathbf{x}^*)$ globally, where $(\lambda^*, \mathbf{x}^*)$ is some Z-eigenpair of $\mathcal{A}$. Furthermore, $\{f(\mathbf{x}_k)\}$ converges to $f(\mathbf{x}^*)$ linearly.

Proof
It follows from Assumption 4.3 that $\{\mathbf{x}_k\}$ is convergent to $\mathbf{x}^*$. We showed that $\mathbf{z}^* = \mathcal{A}(\mathbf{x}^*)^{m-1} - \lambda^* \mathbf{x}^* = 0$ in Lemma 4.2. Thus, $(\lambda^*, \mathbf{x}^*)$ is some Z-eigenpair of $\mathcal{A}$.

DOI: 10.1002/nla

Further, it follows from (4.13) and (4.21) that

$$\begin{aligned}\lim_{k\to\infty}\frac{f(\mathbf{x}^*)-f(\mathbf{x}_{k+1})}{f(\mathbf{x}^*)-f(\mathbf{x}_k)} &= 1-\lim_{k\to\infty}\frac{f(\mathbf{x}_{k+1})-f(\mathbf{x}_k)}{f(\mathbf{x}^*)-f(\mathbf{x}_k)}\\ &\leqslant 1-\frac{1}{4(M+L)}\lim_{k\to\infty}\frac{\|\mathbf{z}_k\|^2}{f(\mathbf{x}^*)-f(\mathbf{x}_k)}\\ &\leqslant 1-\frac{\mu_{n-1}}{2M+2L}\\ &< 1.\end{aligned} \tag{4.28}$$

This completes the proof. □

The aforementioned theorem is established under Assumption 4.3. If $\{\mathbf{x}_k\}$ is an infinite sequence, there must be some convergent subsequence $\{\mathbf{x}_{k_i}\}$, say $\{\mathbf{x}_{k_i}\}\to\mathbf{x}^*$ for some point $\mathbf{x}^*$ on the unit sphere. Therefore, the point $\mathbf{x}^*$ can also been proved to be a Z-eigenvector.

5. NUMERICAL RESULTS

In this section, we report some numerical results of the proposed SSPM and compare SSPM with the power method (PM). Our code was implemented in MATLAB (R2013a). All the experiments were preformed on a Dell desktop with Intel dual core E6750 CPU at 2.66 GHz and 2 GB of memory running Windows 7. The initial point $\bar{\mathbf{x}}_0$ is generated randomly, and the parameter ϵ in the stopping criterion is set to be 10^{-10}.

The solution to each testing problem is known in advance, and hence, we can tell whether SSPM and PM find the right solution or not. Moreover, we focus on 4-order tensors in the numerical experiments for convenience. In this case, Step 2 of Algorithm 3.1 involves the quartic equation

$$\alpha_1 a^4-(\alpha_0-3\alpha_2)a^3-3(\alpha_1-\alpha_3)a^2-(3\alpha_2-\alpha_4)a-\alpha_3=0,$$

where

$$\alpha_0=\mathcal{A}\mathbf{p}_k^4,\ \alpha_1=\mathcal{A}\mathbf{p}_k^3\mathbf{q}_k,\ \alpha_2=\mathcal{A}\mathbf{p}_k^2\mathbf{q}_k^2,\ \alpha_3=\mathcal{A}\mathbf{p}_k\mathbf{q}_k^3,\ \alpha_4=\mathcal{A}\mathbf{q}_k^4.$$

5.1. Testing diagonal tensors

The following proposition [1] gives all the Z-eigenvalues and the related Z-eigenvectors of diagonal tensors. Based on this proposition, we can choose diagonal tensors as testing problems.

Proposition 5.1
([1]) Suppose $\mathcal{A}$ is a diagonal tensor with diagonal elements $\{a_1,a_2,\ldots,a_n\}$. Let

$$J_1=\{i\ :\ a_i<0\},\quad J_2=\{i\ :\ a_i>0\}.$$

If at least one of J_1 and J_2 has more than one element, then $\mathcal{A}$ has more than n Z-eigenvalues. In this case, besides the n Z-eigenvalues, which are the diagonal elements of $\mathcal{A}$, for each $\bar{J}_k\subseteq J_k$ with $|\bar{J}_k|\geqslant 2$, $k=1,2$,

$$\lambda=(-1)^k\left[\sum_{i\in\bar{J}_k}\left(\frac{1}{|a_i|}\right)^{\frac{2}{m-2}}\right]^{-\frac{m-2}{2}}$$

is a Z-eigenvalue of $\mathcal{A}$, with a Z-eigenvector $\mathbf{x}$ defined by

$$x_i=\begin{cases}\left(\frac{\lambda}{a_i}\right)^{\frac{1}{m-2}}, & \text{for } i\in\bar{J}_k;\\ 0, & \text{otherwise.}\end{cases}$$

DOI: 10.1002/nla

Table I. Compare SSPM with PM for Example 5.1

	PM			SSPM			SSPM		
n	iter	CPU	λ	iter	CPU	$\lambda_{\max}$	iter	CPU	$\lambda_{\min}$
10	3	0.0012	100	3	0.0052	100	8	0.0132	3.4142
20	5	0.0203	200	4	0.0581	200	7	0.0983	2.7795
30	5	0.0930	300	4	0.2602	300	9	0.5677	2.5031
40	5	0.2705	400	4	0.8290	400	10	2.0087	2.3372
50	5	0.7583	500	4	2.5156	500	8	4.8226	2.2226
60	5	2.1273	600	4	6.5730	600	10	15.2800	2.1368
70	6	5.2405	700	5	17.0260	700	8	26.2312	2.0692
80	5	7.8705	800	4	23.5201	800	11	62.4543	2.0139

PM = power method; SSPM = sequential subspace projection method.

Proposition 5.2
Consider a 4-order diagonal tensor $\mathcal{A}$, $a_i > 0$, $i = 1, 2, \ldots, n$. Suppose that $a_1 \geqslant a_2 \geqslant \cdots \geqslant a_n > 0$. Then the largest Z-eigenvalue $\lambda_{\max} = a_1$, and the smallest Z-eigenvalue $\lambda_{\min} = \frac{1}{\sum_{i=1}^{n} \frac{1}{a_i}}$.

Example 5.1
Consider the diagonal tensor $\mathcal{A}$ with diagonal elements $\mathcal{A}_{i,i,i,i} = 10i$, $i = 1, 2, \ldots, n$ for $n = 10, 20, \cdots, 80$.

We record the numerical results in Table I, where 'iter' is the iteration number, 'CPU' is the CPU time (in seconds), 'λ' is the Z-eigenvalue returned by PM, and '$\lambda_{\max}$' ('$\lambda_{\min}$') is the largest (smallest) Z-eigenvalue returned by SSPM. For the sake of fairness, the three algorithms share the same initial point. From Table I, it can be seen that SSPM requires fewer iterations, but a little more CPU time. Noting that SSPM looks for an optimal solution in a 2-dimensional subspace while PM does in a 1-dimensional subspace, it is reasonable that SSPM is likely to require fewer iterations than PM. From the computational complexity analysis in Section 3.1, SSPM needs two tensor-vector multiplications while PM only needs one, so SSPM needs more computation per iteration. In this experiment, the iteration numbers required by PM and SSPM are very close, which can interpret why SSPM takes more total CPU time than PM.

5.2. *A numerical example that favors sequential subspace projection method*

In this subsection, we test a kind of tensor from [17], which shows that SSPM might perform much better than PM.

Example 5.2
[17] Consider 4-order n-dimensional supersymmetric tensors with the elements

$$\mathcal{A}_{i_1,\ldots,i_4} = \arctan\left((-1)^{i_1}\frac{i_1}{n}\right) + \cdots + \arctan\left((-1)^{i_4}\frac{i_4}{n}\right).$$

For this example, the largest magnitude Z-eigenvalue is the smallest one. Hence, PM returns the smallest Z-eigenvalue, and SSPM is called to calculate the smallest Z-eigenvalue. The results are reported in the following Table II. We can see that SSPM performs much better than PM in both iterations and CPU time. Specifically, SSPM can terminate in 10 iterations, while PM takes far more. Thus, SSPM uses less CPU time than PM in this test, although it takes more CPU time per iteration.

Before we proceed, several remarks are in order.

Remark 4
We can also use the Matlab build-in solver GloptiPoly3 to solve the subproblem (2.5). To compare GloptiPoly3 with the direct method, we test Example 5.1 for $n = 10, 20, \cdots, 80$. The two methods

DOI: 10.1002/nla

Table II. Compare the quality SSPM and PM for Example 5.2.

	PM			SSPM		
n	iter	CPU	λ	iter	CPU	$\lambda_{\min}$
5	23	0.0016	-2.357e+01	6	0.0028	-2.357e+01
15	97	0.0897	-1.650e+02	6	0.0281	-1.650e+02
25	160	0.8646	-4.353e+02	7	0.1847	-4.353e+02
35	218	4.1928	-8.342e+02	8	0.7868	-8.342e+02
45	285	15.2155	-1.361e+03	7	2.1085	-1.361e+03
55	348	53.6189	-2.018e+03	8	6.9728	-2.018e+03
65	421	171.6298	-2.803e+03	7	14.4386	-2.803e+03
75	503	411.9391	-3.716e+03	5	19.4202	-3.716e+03
85	544	739.4757	-4.758e+03	7	49.7785	-4.758e+03
95	608	1315.2707	-5.929e+03	7	70.3252	-5.929e+03

PM = power method; SSPM = sequential subspace projection method.

provide the same solution to the subproblem (2.5) and lead to the same sequence of iterates $\{\mathbf{x}_k\}$, but the CPU time is slightly different. We report the CPU time (in seconds) in Table III.

Table III. Compare direct method with GloptiPoly3 for the subproblem (2.5).

n	10	20	30	40	50	60	70	80
Root	0.005	0.058	0.260	0.829	2.517	6.573	17.026	23.520
Glo3	0.449	0.506	0.716	1.303	2.877	6.832	17.153	24.309

PM = power method; SSPM = sequential subspace projection method.

Remark 5

SSPM can also be applied to supersymmetric tensors with general m, although we test $m = 4$ in the experiments. By the direct method [12], the subproblem (2.5) leads to an m degree equation (2.7), which can be solved by Wu method [24]. Besides, GloptiPoly3 is also a choice to solve the subproblem directly.

Remark 6

PM outputs the largest magnitude Z-eigenvalue, but SSPM can find both the smallest one and the largest one. For example,

$$\mathcal{A} = \text{diag}\{-8, -7, -6, -5, 1, 2, 3, 4\}.$$

PM outputs -8. SSPM can find both -8 and 4. However, SSPM applies only to supersymmetric tensors while PM can also work for unsymmetric ones.

5.3. Efficiency of the global strategy of the random phase

To improve the performance of SSPM in finding extreme Z-eigenvalues, the globalization strategy of random phase is introduced in Algorithm 3.2 once the iterates get trapped in local minimizers/maximizers.

Let $\mathcal{A}$ be an m-th order n-dimensional supersymmetric tensor and $P = (p_{i,j})$ be an $n \times n$ real matrix. Define $\mathcal{B} = P^m\mathcal{A}$ as another m-th order n-dimensional tensor with entries [12],

$$b_{i_1,i_2,\ldots,i_m} = \sum_{j_1,j_2,\ldots,j_m=1}^{n} p_{i_1,j_1} p_{i_2,j_2} \cdots p_{i_m,j_m} a_{j_1,j_2,\ldots,j_m}.$$

Then $\mathcal{B}$ is also a supersymmetric tensor. If P is an orthogonal matrix, we say that $\mathcal{A}$ and $\mathcal{B}$ are orthogonally similar [12]. If $\mathcal{A}$ and $\mathcal{B}$ are orthogonally similar, $\mathcal{A}$ and $\mathcal{B}$ share the same Z-eigenvalues

DOI: 10.1002/nla

Table IV. Compare SSPM, SSPMr with PM for Example 5.3.

n	=	>	<	PM_{opt}	$SSPM_{opt}$	=	>	<	PM_{opt}	$SSPMr_{opt}$
6	93	0	7	41	45	41	0	59	41	100
8	83	1	16	42	49	43	0	57	42	99
10	86	0	14	50	61	50	0	50	50	98
12	94	0	6	65	69	65	0	35	65	100

PM = power method; SSPM = sequential subspace projection method; SSPMr = SSPM with random phase.

and related Z-eigenvectors [12]. Tensors that are orthogonally similar to diagonal ones are tested to illustrate the efficiency of this globalization strategy.

Example 5.3
Let $\mathcal{A}, \mathcal{B}$ be 4-order n-dimensional diagonal tensors, with $\mathcal{B} = P^m\mathcal{A}$ and $\mathcal{A}_{i,i,i,i} = i$, $i = 1, 2, \ldots, n$,

$$P = (I - 2\mathbf{w}_1\mathbf{w}_1^T)(I - 2\mathbf{w}_2\mathbf{w}_2^T)(I - 2\mathbf{w}_3\mathbf{w}_3^T),$$

where $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ are unit vectors generated randomly. The task is to find the largest Z-eigenvalue of $\mathcal{B}$.

We test Example 5.3 with different n. Each test is conducted 100 times with different initial points. The left half of Table IV compares PM with SSPM, and the right half with SSPMr. In this table, $= (>, <)$ stands for the number of the cases that PM finds the same (better, worse) Z-eigenvalue, and PM_{opt} ($SSPM_{opt}$, $SSPMr_{opt}$) represents the number of cases that PM (SSPM, SSPMr) finds the optimal solution.

From Table IV, it can be seen that both SSPM and SSPMr perform better than PM in that they can find the optimal solutions with higher probability. If all the three algorithms fail to return the largest Z-eigenvalue, SSPM and SSPMr always return a larger one than PM. Obviously, SSPMr has a higher successful rate than SSPM. Hence, the globalization strategy of the random phase does work.

5.4. Determining positive definiteness of tensors

The smallest Z-eigenvalue is useful for checking whether a supersymmetric tensor is positive definite or not [12]. In Algorithm 3.1, once some $\lambda_k < 0$ in some subspace is detected, we can stop the algorithm and claim that the tensor is not positive definite.

The original problem and the subproblem are

$$\begin{aligned} \lambda^{\min} = \min \quad & \mathcal{A}\mathbf{x}^m \\ \text{s.t.} \quad & \mathbf{x}^T\mathbf{x} = 1 \end{aligned}$$

and

$$\begin{aligned} \lambda_k^{\min} = \min \quad & \mathcal{A}_k\mathbf{l}^m \\ \text{s.t.} \quad & \mathbf{l}^T\mathbf{l} = 1, \end{aligned}$$

respectively. It is easy to see that $\lambda^{\min} \leqslant \lambda_k^{\min}$. Therefore, if there exists a $\lambda_j^{\min} < 0$ for some j, we can have $\lambda^{\min} < 0$ and claim that $\mathcal{A}$ is not positive definite.

We utilize Algorithm 3.1 to compute the smallest Z-eigenvalue of the diagonal tensor

$$\mathcal{A} = \text{diag}\{-0.002, -0.001, 1, 2, 3, 4, 6, 7, 8, 8.001\}$$

with 50 different random initial points. In our experiments, Algorithm 3.1 takes two iterations for 41 times, one iteration for six times, three iterations twice and four iterations once to detect the phenomenon that $\lambda_{\min}^j < 0$. Moreover, similar results have been obtained for the tensors that are orthogonally similar to $\mathcal{A}$.

DOI: 10.1002/nla

6. CONCLUSIONS AND DISCUSSIONS

A new algorithm, SSPM, for finding extreme Z-eigenvalues of supersymmetric tensors is proposed. It was motivated by SSPM in [21] and the direct method for Z-eigenvalue of tensor with $n = 2$ in [12]. Besides, we present SSPMr in which a random phase is incorporated so that the algorithm can escape from the local minimizers/maximizers with a high probability. Theoretically, the global convergence and local linear convergence have been established under some mild assumptions. Preliminary numerical results illustrated that SSPM takes few iterations than PM, but it needs more computational cost and slightly more storage cost per iteration. For some specific problems, SSPM can reduce the number of iterations and CPU time significantly compared with PM, as shown in Subsection 5.2.

ACKNOWLEDGEMENTS

The authors thank Dr. Bo Jiang for his useful discussions about the convergence analysis in Section 4. They are also grateful to the anonymous referees and the managing editor for their valuable comments, which improved the quality of this paper greatly. The work of the first author is supported by Chinese NSF grant (no. 11301016) and the Research Fund for the Doctoral Program of Higher Education of China (no. 20131103120029) and that of the third author is supported by the Chinese NSF grants (nos. 11331012 and 81173633) and the China National Funds for Distinguished Young Scientists (no. 11125107).

REFERENCES

1. Qi LQ. Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation* 2005; **40**:1302–1324.
2. Lim LH. Singular values and eigenvalues of tensors: a variational approach. In *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Addaptive Processing (CAMSAP'05)*, Vol. 1. IEEE Computer Society Press: Piscataway, New Jersey, 2005; 129–132.
3. Qi LQ, Yu GH, Wu EX. Higer order positive semidefinite diffusion tensor imaging. *SIAM Journal on Imaging Sciences* 2010; **3**(3):416–433.
4. Qi LQ, Teo KL. Multivariate polynomial minimization and its application in signal processing. *Journal of Global Optimization* 2003; **26**:419–433.
5. Wei TC, Goldbart PM. Geometric measure of entanglement and applications to bipartite and multipartite quantum states. *Physical Review A* 2003; **68**:042307.
6. Lathauwer LD, Comon P, Moor BD, Vandewalle J. High-order power method-application in independent component analysis. In *Proceeding of the International Symposium on Nonlinear Theory and its Applications (NOLTA'95)*: Las Vegas, Nevada, 1995; 91–96.
7. Cardoso JF. High-order contrasts for independent component analysis. *Neural computation* 1999; **11**:157–192.
8. Qi LQ. Rank and eigenvalues of a supersymmetric tensor, the multivariate homogeneous polynomial and the algebraic hypersurface it defines. *Journal of Symbolic Computation* 2006; **41**:1309–1327.
9. Qi LQ. Eigenvalues and invariants of tensors. *Journal of Mathematical Analysis and Applications* 2007; **325**(2): 1363–1377.
10. Ni G, Qi LQ, Wang F, Wang Y. The degree of the E-characteristic polynomial of an even order tensor. *Journal of Mathematical Analysis and Applications* 2007; **329**:1218–1229.
11. Qi LQ. *Eigenvalues of a Supersymmetric Tensor and Positive Definiteness of an Even Degree Multivariate Form*. Department of Applied Mathematics: The Hong Kong Polytechnic University, 2004.
12. Qi L Q, Wang F, Wang YJ. Z-eigenvalue methods for a global polynomial optimization problem. *Mathmatical Programming* 2009; **118**:301–316.
13. Lathauwer LD, Moor BD, Vandewalle J. On the best rank-1 and rank- $(R_1, R_2, ..., R_N)$ approximation of higher-order tensors. *SIAM Journal of Matrix Analysis and Applications* 2000; **21**:1324–1342.
14. Kofidis E, Regalia PA. On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM Journal of Matrix Analysis and Applications* 2002; **23**:863–884.
15. Zhang T, Golub GH. Rank-1 approximation of higer-order tensors. *SIAM Journal of Matrix Analysis and Applications* 2001; **23**:534–550.
16. Zhang XZ, Ling C, Qi LQ. The best rank-1 approximation of a symmetric tensor and related spherical optimization problems. *SIAM Journal on Matrix Analysis and Applications* 2012; **33**(3):806–821.
17. Nie JW, Wang L. Semidefinite relaxations for best rank-1 tensor approximations. *ArXiv* 1308.6562, 2013.
18. Ni Q, Qi LQ, Wang F. An eigenvalue method for testing positive definiteness of a multivariate form. *IEEE Transactions on Automatic Control* 2008; **53**(5):1096–1107.

DOI: 10.1002/nla

19. Kolda TG, Mayo JR. Shifted power method for computing tensor eigenpairs. *SIAM Journal of Matrix Analysis and Applications* 2011; **32**(4):1095–1124.
20. Han LX. An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors. *Numerical Algebra, Control and Optimization* 2013; **3**(3):583–599.
21. Dai Y H. Fast algorithm for projection on an ellipsoid. *SIAM Journal on Optimization* 2006; **16**:986–1006.
22. Liu X, Hao CL, Cheng MH. A sequential subspace projection method for linear eigenvalue problem. *Asia Pacific Journal of Operational Research* 2013; **30**(3):1340003.
23. Khoromskij B N, Khoromskaia V. Multigrid accelerated tensor approximation of function related multi-dimensional arrays. *SIAM Journal on Scientific Computing* 2009; **31**(4):3002–3026.
24. Wang DM, Xia BC, Li ZM. *Computer Algebra* (2nd edn). Tsinghua University Press: Beijing, 2007.

DOI: 10.1002/nla