Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 16, No. 1, pp. 87 – 99, 2020

AN INEXACT LEVENBERG-MARQUARDT METHOD FOR TENSOR EIGENVALUE COMPLEMENTARITY PROBLEM*

Huanhuan Li, Shouqiang Du†, Yingxiao Wang and Miao Chen

Abstract: In this paper, we consider the method for solving tensor eigenvalue complementarity problem due to its various applications in engineering and polynomial optimization. We formulate the tensor eigenvalue complementarity problem as semismooth equations involving an NCP function and propose an inexact Levenberg-Marquardt method to solve it. The inexact Levenberg-Marquardt method is shown to be both globally convergent and local Q-superlinear convergence. Finally, numerical experiments are reported to show the efficiency of proposed method.

Key words: *tensor eigenvalue complementarity problem, Levenberg-Marquardt method, convergence*

Mathematics Subject Classification: *15A18, 15A69, 90C33*

1 Introduction

A tensor is a multidimensional array, a real mth-order n-dimensional tensor $\mathcal{A}$ consists of n^m real entries $a_{i_1\ldots i_m}$, where $i_j \in I_n, j \in I_m, I_n := \{1,2,\ldots,n\}, I_m := \{1,2,\ldots,m\}$. Denote the set of all real mth-order n-dimensional tensors by $T_{m,n}$, then $T_{m,n}$ is a linear space of dimension n^m. Let $\mathcal{A} \in T_{m,n}$, if the entries $a_{i_1\ldots i_m}$ are invariant under any permutation of their indices, then $\mathcal{A}$ is called a symmetric tensor. $S_{m,n}$ denotes the set of all real symmetric mth-order n-dimensional tensors. We call an mth-order n-dimensional tensor $\mathcal{I}$ the mth identity tensor, if its entries satisfy

$$\delta_{i_1\ldots i_m} = \begin{cases} 1, \text{ if } i_1 = \cdots = i_m, \\ 0, \text{ otherwise.} \end{cases}$$

For $x \in R^n$, $\mathcal{A}x^{m-1}$ is a vector on R^n with its ith component given by

$$\left(\mathcal{A}x^{m-1}\right)_i := \sum_{i_2,\ldots,i_m=1}^{n} a_{ii_2\ldots i_m} x_{i_2}\ldots x_{i_m},$$

for $i \in I_n$. And $\mathcal{A}x^m$ is defined by $\mathcal{A}x^m = x^T\mathcal{A}x^{m-1}$. A tensor $\mathcal{A} \in S_{m,n}$ is called positive definite if $\mathcal{A}x^m > 0$ for all $x \neq 0$.

*This work is supported by National Natural Science Foundation of China (No.11671220) and Natural Science Foundation of Shandong Province (No.ZR2016AM29).

†Corresponding author.

In this paper, we consider the tensor eigenvalue complementarity problem (TEiCP) which is to find a scalar $\lambda \in R$ and a vector $x \in R^n \setminus \{0\}$ satisfying the conditions

$$x \geq 0,\ (\lambda\mathcal{B} - \mathcal{A})x^{m-1} \geq 0,\ \langle x, (\lambda\mathcal{B} - \mathcal{A})x^{m-1}\rangle = 0, \tag{1.1}$$

for $\mathcal{A}, \mathcal{B} \in T_{m,n}$. Especially, when $m = 2$, problem (1.1) is reduced to the classical eigenvalue complementarity problem, which has wide applications in mechanical systems(see [14,22]). In recent years, tensor eigenvalue problem has attracted extensive interests for the wide spread applications in engineering, polynomial optimization and related fields. According to Qi's definition in [23], the eigenvalue and eigenvector of tensor $\mathcal{A}$ were the scalar $\lambda \in C$ and nonzero vector $x \in C^n$ satisfying $\mathcal{A}x^{m-1} = \lambda x^{[m-1]}$, where $x^{[m-1]} = (x_1^{m-1}, \ldots, x_n^{m-1})^T$. When $\lambda \in R$ and $x \in R^n$, they were called H-eigenvalue and H-eigenvector, respectively. And Qi also proved that H-eigenvalues and Z-eigenvalues existed for an even-order real supersymmetric tensor. In addition, Chang, Pearson and Zhang introduced the definition of generalized tensor eigenpair in [3]. Let $\mathcal{A}, \mathcal{B} \in T_{m,n}$, where m is even and $\mathcal{B}$ is positive definite. Then $(\lambda, x) \in C \times \{C^n \setminus \{0\}\}$ is a generalized eigenpair of $(\mathcal{A}, \mathcal{B})$, which satisfying $\mathcal{A}x^{m-1} = \lambda\mathcal{B}x^{m-1}$. In [23], Qi studied the symmetric hyperdeterminant, eigenvalues and E-eigenvalues of a real supersymmetric tensor. Ye and Hu studied the solvability of the inverse eigenvalue problem for tensor in [26]. Hu, Huang and Qi introduced the tensor conic linear programming(TCLP) in [12]. They introduced an approximation method and reformulated the extreme Z-eigenvalue problem as a special tensor conic linear programming. Chen and Wang proposed a sum-of-squares algorithm to compute the minimal H-eigenvalues of tensors in [4]. Zhang and Qi established an explicit convergence rate of the Ng-Qi-Zhou method for positive tensors in [28]. Ling, He and Qi showed that TEiCP was solvable and they also presented two equivalent optimization reformulations of TGEiCP in [18]. They also showed that the TEiCP had at least one solution, under the assumption that $\mathcal{B}x^{m-1} \neq 0$ for all $x \in R_+^n \setminus \{0\}$. Song and Qi [24] introduced the concepts of Pareto H-eigenvalue and Pareto Z-eigenvalue. And they also studied the properties of Pareto eigenvalue and the relationship between the minimum Pareto H-eigenvalue of a symmetric tensor and polynomial optimization. Ng, Qi and Zhou proposed an iterative method for calculating the largest eigenvalue of an irreducible nonnegative tensor in [20]. Fan, Nie and Zhou formulated TEiCP as constrained polynomial optimization in [8]. Chen and Qi reformulated TEiCP as nonlinear equations and proposed a damped semismooth Newton method for solving it in [5]. Yu et al. investigated two monotone ascent spectral projected gradient (SPG) methods for TEiCP in [27]. Ling, He and Qi reformulated tensor higher-degree eigenvalue complementarity problem with symmetric tensor as a weakly coupled polynomial optimization problem in [19].

Z-eigenvalues of $\mathcal{A}$ has wide range of research in the literature, such as [23,24]. When the order of $\mathcal{A}$ was even, its Z-eigenvalues always existed and an even order supersymmetric tensor was positive definite if and only if all of its Z-eigenvalues were positive. Therefore, firstly, a natural question is that can we solve TEiCP (1.1) with m is even and $\mathcal{A} \in S_{m,n}$ by optimization method? As we all know, Levenberg-Marquardt method is one of the important optimization methods for solving optimization problems, see, e.g., [7,11,21]. Most recently, Fan and Pan analyzed the convergence rate of an inexact Levenberg-Marquardt method for nonlinear equations in [9]. In [16], Karas, Santos and Svaiter used Levenberg-Marquardt method to solve the nonlinear least-squares problem and gave a convergence analysis of the proposed method. In [17], Lqbal, Lqbal and Arif used the Levenberg-Marquardt method to solve the absolute value equations. Secondly, another question is that can we use the Levenberg-Marquardt method to solve TEiCP (1.1) with m is even and $\mathcal{A} \in S_{m,n}$ and

obtain more interesting properties? Motivated by these questions, in this paper, we study an inexact Levenberg-Marquardt method for TEiCP (1.1), where m is even and $\mathcal{A} \in S_{m,n}$.

The rest of this paper is organized as follows. In Section 2, we reformulate TEiCP (1.1) as nonsmooth equations by using Fischer-Burmeister function and present an inexact Levenberg-Marquardt method. We also analyse the convergence properties of this method. In Section 3, we report some numerical results to show the efficiency of the presented method. Finally, some conclusions are also given.

Through this paper, lowercase letters $(x, y, z, \dots)$ denote vectors, italic capitals (A, B, I) denote matrices, calligraphic capitals $(\mathcal{A}, \mathcal{B}, \mathcal{I})$ denote tensors. All the tensors discussed are real tensors.

2 Preliminaries and the Inexact Levenberg-Marquardt Method

In this section, firstly, we present properties of the transformed nonsmooth equations. Then an inexact Levenberg-Marquardt method for TEiCP is proposed. Finally, the convergence properties of the given method are also presented.

Now, we introduce the Fischer-Burmeister function [10,25] defined as

$$\phi(a, b) = (a + b) - \sqrt{a^2 + b^2}.$$

The Fischer-Burmeister function has been widely used in solving many optimization problems, such as constrained optimization problems, variational inequality problems and complementarity problems. By the Fischer-Burmeister function, problem (1.1) can be reformulated as nonsmooth equations

$$H(z) = H(x, y, \lambda) = \begin{pmatrix} \Phi(x, y) \\ (\lambda\mathcal{B} - \mathcal{A})x^{m-1} - y \\ x^T x - 1 \end{pmatrix} = 0, \tag{2.1}$$

where

$$\Phi(x, y) = \begin{pmatrix} \phi(x_1, y_1) \\ \phi(x_2, y_2) \\ \vdots \\ \phi(x_n, y_n) \end{pmatrix}, \tag{2.2}$$

$z = (x, y, \lambda) \in R^n \times R^n \times R$ and $\phi(x_i, y_i) = (x_i + y_i) - \sqrt{x_i^2 + y_i^2}$, for $i = 1, 2, \dots, n$. Now, we will show that the functions defined by (2.1) and (2.2) are all strongly semismooth functions. This property is very important for the convergence analysis of the inexact Levenberg-Marquardt method.

Theorem 2.1. *Let H be defined by (2.1). Then H is strongly semismooth at z.*

Proof. By Theorem 1 in [25], we know that Φ is strongly semismooth at (x, y). Note that for any $z = (x, y, \lambda) \in R^n \times R^n \times R$, $(\lambda\mathcal{B} - \mathcal{A})x^{m-1} - y$ is continuously differentiable. Moreover, $x^T x - 1$ is continuously differentiable at x. Therefore, H is strongly semismooth at z. This proves the theorem. □

We define the merit function associated with H by

$$\Psi(z) = \frac{1}{2}\|H(z)\|^2.$$

Now, we know that problem (1.1) is equivalent to the following unconstrained optimization problem

$$\min \Psi(z)$$

with zero optimal value. We denote the generalized Jacobian of H at z by $\partial H(z)$. Further, we consider the properties of Ψ, where Ψ is continuously differentiable at z.

Theorem 2.2. *The function Ψ is continuously differentiable at z and its gradient is given by*

$$\nabla\Psi(z) = Q^T H(z),$$

where $Q \in \partial H(z)$.

The generalized Jacobian of H at $z = (x, y, \lambda) \in R^n \times R^n \times R$ is defined by

$$\partial H(z) = \begin{pmatrix} D_a(z) & D_b(z) & 0 \\ (m-1)(\lambda\mathcal{B} - \mathcal{A})x^{m-2} & -I & \mathcal{B}x^{m-1} \\ 2x^T & 0 & 0 \end{pmatrix},$$

where

$$D_a(z) := diag\{a_1(z), \ldots, a_n(z)\},$$
$$D_b(z) := diag\{b_1(z), \ldots, b_n(z)\},$$

and

$$a_i(z) = 1 - \frac{x_i}{\sqrt{x_i^2 + y_i^2}},$$
$$b_i(z) = 1 - \frac{y_i}{\sqrt{x_i^2 + y_i^2}},$$

for $(x_i, y_i) \neq (0, 0), i = 1, 2, \ldots, n$. For $(x_i, y_i) = (0, 0)$, we define $(a_i(z), b_i(z)) = (1-\zeta, 1-\varsigma)$, where (ζ, ς) satisfies $\|(\zeta, \varsigma)\| \leq 1, i = 1, 2, \ldots, n$. And for a tensor $\mathcal{T} = (t_{i_1 \ldots i_m}) \in T_{m,n}$ and a vector $x \in R^n$, $\mathcal{T}x^{m-2}$ is a matrix in $R^{n\times n}$ with its (i, j)-th component defined by

$$(\mathcal{T}x^{m-2})_{i,j} = \sum_{i_3,\ldots,i_m=1}^{n} t_{iji_3\ldots i_m} x_{i_3} \ldots x_{i_m}.$$

We are in the position to present the inexact Levenberg-Marquardt method.

Algorithm 1: An inexact Levenberg-Marquardt method for TEiCP

Step 0: Given a starting vector $z_0 \in R^n$ and some scales $p > 2, 0 < \beta < \frac{1}{2}, \rho > 0, \varepsilon \geq 0$. Set $k := 0$.
Step 1: If $\|H(z_k)\| \leq \varepsilon$, stop. Otherwise, choose $Q_k \in \partial H(z_k)$.
Step 2: Find an approximate solution d_k satisfying the equation

$$((Q_k)^T Q_k + \mu_k I)d_k = -\nabla\Psi(z_k) + r_k, \tag{2.3}$$

where $\mu_k \geq 0$ is the inexact Levenberg-Marquardt parameter and r_k is the vector of residuals. If the condition

$$\nabla\Psi(z_k)^T d_k \leq -\rho\|d_k\|^p$$

is not satisfied, set

$$d_k = -\nabla\Psi(z_k).$$

Step 3: Find the smallest integer $i_k \in \{0, 1, 2, \ldots\}$ such that $t_k = 2^{-i_k}$ and

$$\Psi(z_k + t_k d_k) \leq \Psi(z_k) + \beta t_k \nabla\Psi(z_k)^T d_k.$$

Step 4: Set $z_{k+1} = z_k + t_k d_k$, $k := k + 1$, go to Step 1.

In Algorithm 1, d_k can be computed by inexact equation (2.3). In the implementation of inexact Levenberg-Marquardt method, we can set $\|r_k\| \leq \alpha_k\|\nabla\Psi(z_k)\|$, where $\{\alpha_k\}$ is a sequence and $\alpha_k \in (0, 1)$, $\alpha_k \to 0 (k \to \infty)$. To ensure global convergence, a line search is used to minimize the natural merit function Ψ. If the search direction d_k generated by (2.3) is not a good descent direction, we can set $d_k = -\nabla\Psi(z_k)$. In the following of this section, we analyze the convergence properties of the given method. Next, we present the following global convergence theorem.

Theorem 2.3. *Suppose that $\{\mu_k\}$ is bounded and $\{r_k\}$ satisfies*

$$\|r_k\| \leq \alpha_k\|\nabla\Psi(z_k)\|,$$

where $\{\alpha_k\}$ is a sequence of numbers with $0 < \alpha_k < 1$ and $\alpha_k \to 0$ as $k \to \infty$. Then each accumulation point of $\{z_k\}$ is a stationary point of Ψ.

Proof. Let $\{z_k\}_K$ is a subsequence of $\{z_k\}$ and $\{z_k\}_K \to z^*$, $k = 1, 2, \ldots$. Suppose that $d_k = -\nabla\Psi(z_k)$ for all $k \in K$, where K is an infinite set. Then we get the result of this theorem by Proposition 1.16 in [2]. If d_k is always computed by (2.3), we can get this theorem by Theorem 12 in [11]. This completes the proof. □

In the following of this section, we discuss the local rate convergence of the inexact Levenberg-Marquardt method. Here, we assume that $\{r_k\}$ satisfies $\|r_k\| \leq \alpha_k\|\nabla\Psi(z_k)\|$, where $\{\alpha_k\}$ is a sequence of numbers with $0 < \alpha_k < 1$ and $\alpha_k \to 0$ as $k \to \infty$. Then we get the following theorem.

Theorem 2.4. *Suppose that z^* is one of the accumulation point of $\{z_k\}$ and $2\mathcal{B}x^m \cdot D_b(z^*) \neq 0$, $\{\mu_k\} \to 0$. Then $z_k \to z^*$ Q-superlinearly.*

Proof. From Theorem 2.1, $2\mathcal{B}x^m \cdot D_b(z^*) \neq 0$ and $\{\mu_k\} \to 0$, we can get this theorem by Theorem 14 in [11]. □

3 Numerical Experiments

We implemented Algorithm 1 with the code in Matlab Version R2014a with Tensor Toolbox Version 2.6 [1]. All numerical experiments were done at a laptop with an Intel (R) Core (TM) i5-2520M CPU (2.50GHz) and RAM of 4.00GB. In order to verify the effectiveness of Algorithm 1, we compare Algorithm 1 with SNM, SSNM, SPP and SPA method presented in [5,6,13,18] and take the examples from [8,13,15,18]. In the implementation of Algorithm 1, we set $\rho = 10^{-6}, p = 2.1, \beta = 10^{-4}$, $\mu_k = 10^{-6}$ and maximum step $N_{max} = 300$.

Example 1 ([13]). Consider $\mathcal{A} \in S_{4,3}$, which is defined by

$$a_{1222} = 1, a_{1333} = 1,$$

$$a_{2111} = 1, a_{3111} = 1,$$

and the other $a_{i_1 i_2 i_3 i_4} = 0$.

We use Algorithm 1 to compute Pareto Z-eigenvalues of $\mathcal{A}$ with the initial point $z_0 = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]^T$ and $\varepsilon = 10^{-5}$. We get Pareto Z-eigenvalue 0.5566 and Pareto Z-eigenvector $[0.8002, 0.4240, 0.4240]^T$, which happens to be as the same as the results given in [13]. The detailed numerical results are presented in Table 1. We also give Figure 1 to show the iterations related to Pareto Z-eigenvalues. It is clearly to see that Algorithm 1 is also efficient.

Table 1: Numerical results for *Example 1.*

Alg.	λ	Eigenvector	Its.	Time(sec.)
Algorithm 1	0.5566	(0.8002,0.4240,0.4240)	5	0.0682
SNM	0.5566	(0.8002,0.4240,0.4240)	5	0.1328
SSNM	0.5566	(0.8002,0.4240,0.4240)	5	0.5437
SPP	0.5566	(0.8004,0.4239,0.4239)	9	0.2817
SPA	0.5566	(0.8004,0.4239,0.4239)	229	2.0233

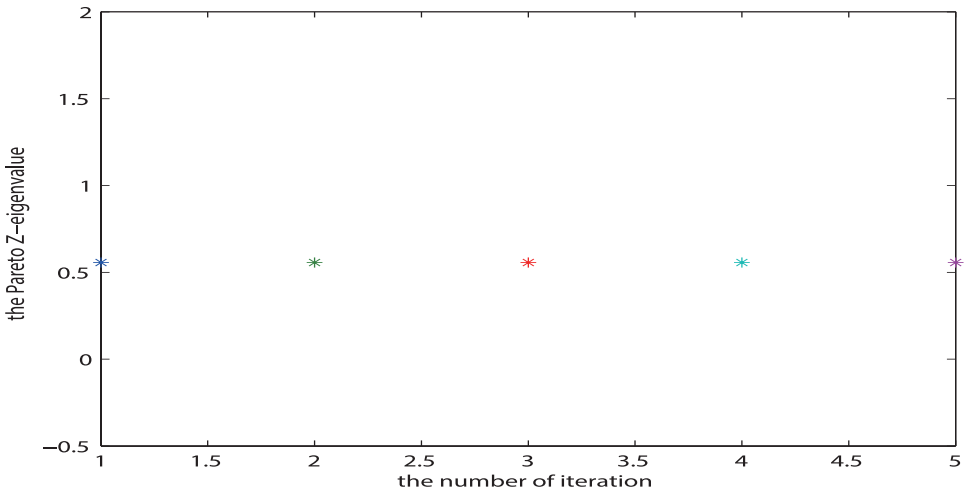

Figure 1: Pareto Z-eigenvalue of Algorithm 1 for *Example 1.*

Example 2. ([13,15]) Consider $\mathcal{A} \in S_{4,3}$, which is defined by

$$a_{1111} = 0.2883,\ a_{1112} = -0.0031,\ a_{1113} = 0.1973,\ a_{1122} = -0.2485,$$

$$a_{1223} = 0.1862,\ a_{1133} = 0.3847,\ \ a_{1222} = 0.2972,\ \ a_{1123} = -0.2939,$$

$$a_{1233} = 0.0919,\ a_{1333} = -0.3619,\ a_{2222} = 0.1241,\ a_{2223} = -0.3420,$$

$$a_{2233} = 0.2127,\ a_{2333} = 0.2727,\ a_{3333} = -0.3054.$$

We use Algorithm 1 to compute Pareto Z-eigenvalues of $\mathcal{A}$ with the initial point $z_0 = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]^T$ and set $\varepsilon = 10^{-5}$. The numerical results are reported in Table 2. We also give Figure 2 to show the iterations related to Pareto Z-eigenvalues.

Table 2: Numerical results for *Example 2.*

Alg.	λ	Eigenvector	Its.	Time(sec.)
Algorithm 1	0.3633	(0.2676,0.6447,0.7160)	6	0.0873
SPP	0.3632	(0.2648,0.6445,0.7162)	7	0.1120
SPA	0.3632	(0.2771,0.6461,0.7112)	260	2.0137
SNM	0.2682	(0.6099,0.4362,0.6616)	5	0.0832
SSNM	0.2682	(0.6099,0.4362,0.6616)	6	0.0860

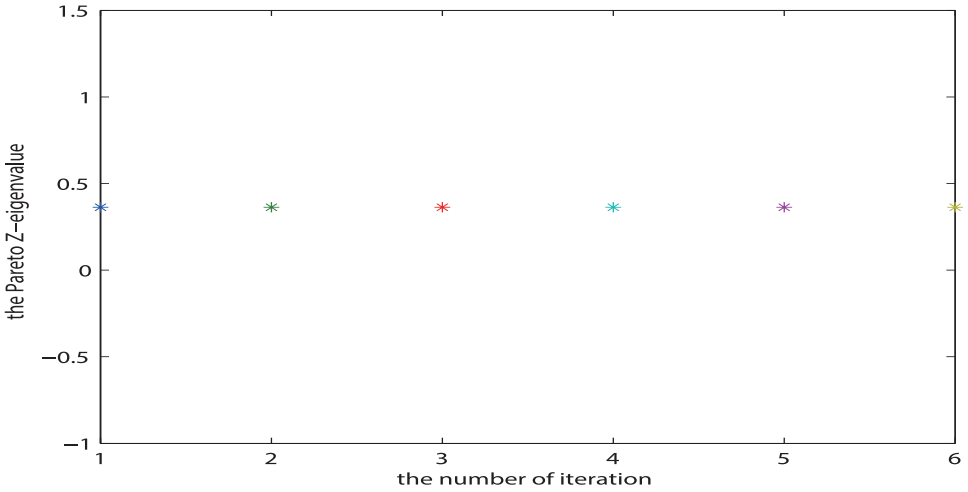

Figure 2: Pareto Z-eigenvalue of Algorithm 1 for *Example 2.*

Example 3. ([8]) Consider $\mathcal{A}$, $\mathcal{B} \in S_{4,3}$, which are defined by

$$\mathcal{A}(:,:,1,1) = \begin{pmatrix} 0.6229 & 0.2644 & 0.3567 \\ 0.2644 & 0.0475 & 0.7367 \\ 0.3567 & 0.7367 & 0.1259 \end{pmatrix}, \mathcal{A}(:,:,1,2) = \begin{pmatrix} 0.7563 & 0.5878 & 0.5406 \\ 0.5878 & 0.1379 & 0.0715 \\ 0.5406 & 0.0715 & 0.3725 \end{pmatrix},$$

$$\mathcal{A}(:,:,1,3) = \begin{pmatrix} 0.0657 & 0.4918 & 0.9312 \\ 0.4918 & 0.7788 & 0.9045 \\ 0.9312 & 0.9045 & 0.8711 \end{pmatrix}, \mathcal{A}(:,:,2,1) = \begin{pmatrix} 0.7563 & 0.5878 & 0.5406 \\ 0.5878 & 0.1379 & 0.0715 \\ 0.5406 & 0.0715 & 0.3725 \end{pmatrix},$$

$$\mathcal{A}(:,:,2,2) = \begin{pmatrix} 0.7689 & 0.3941 & 0.6034 \\ 0.3941 & 0.3577 & 0.3465 \\ 0.6034 & 0.3465 & 0.4516 \end{pmatrix}, \mathcal{A}(:,:,2,3) = \begin{pmatrix} 0.8077 & 0.4910 & 0.2953 \\ 0.4910 & 0.5054 & 0.5556 \\ 0.2953 & 0.5556 & 0.9608 \end{pmatrix},$$

$$\mathcal{A}(:,:,3,1) = \begin{pmatrix} 0.0657 & 0.4918 & 0.9312 \\ 0.4918 & 0.7788 & 0.9045 \\ 0.9312 & 0.9045 & 0.8711 \end{pmatrix}, \mathcal{A}(:,:,3,2) = \begin{pmatrix} 0.8077 & 0.4910 & 0.2953 \\ 0.4910 & 0.5054 & 0.5556 \\ 0.2953 & 0.5556 & 0.9608 \end{pmatrix},$$

$$\mathcal{A}(:,:,3,3)=\begin{pmatrix}0.7581 & 0.7205 & 0.9044\\ 0.7205 & 0.0782 & 0.7240\\ 0.9044 & 0.7240 & 0.3492\end{pmatrix},\mathcal{B}(:,:,1,1)=\begin{pmatrix}0.6954 & 0.4018 & 0.1406\\ 0.4018 & 0.9957 & 0.0483\\ 0.1406 & 0.0483 & 0.0988\end{pmatrix},$$

$$\mathcal{B}(:,:,1,2)=\begin{pmatrix}0.6730 & 0.5351 & 0.4473\\ 0.5351 & 0.2853 & 0.3071\\ 0.4473 & 0.3071 & 0.9665\end{pmatrix},\mathcal{B}(:,:,1,3)=\begin{pmatrix}0.7585 & 0.6433 & 0.2306\\ 0.6433 & 0.8986 & 0.3427\\ 0.2306 & 0.3427 & 0.5390\end{pmatrix},$$

$$\mathcal{B}(:,:,2,1)=\begin{pmatrix}0.6730 & 0.5351 & 0.4473\\ 0.5351 & 0.2853 & 0.3071\\ 0.4473 & 0.3071 & 0.9665\end{pmatrix},\mathcal{B}(:,:,2,2)=\begin{pmatrix}0.3608 & 0.3914 & 0.5230\\ 0.3914 & 0.6822 & 0.5516\\ 0.5230 & 0.5516 & 0.7091\end{pmatrix},$$

$$\mathcal{B}(:,:,2,3)=\begin{pmatrix}0.4632 & 0.2043 & 0.2823\\ 0.2043 & 0.7288 & 0.7400\\ 0.2823 & 0.7400 & 0.9369\end{pmatrix},\mathcal{B}(:,:,3,1)=\begin{pmatrix}0.7585 & 0.6433 & 0.2306\\ 0.6433 & 0.8986 & 0.3427\\ 0.2306 & 0.3427 & 0.5390\end{pmatrix},$$

$$\mathcal{B}(:,:,3,2)=\begin{pmatrix}0.4632 & 0.2043 & 0.2823\\ 0.2043 & 0.7288 & 0.7400\\ 0.2823 & 0.7400 & 0.9369\end{pmatrix},\mathcal{B}(:,:,3,3)=\begin{pmatrix}0.8200 & 0.5914 & 0.4983\\ 0.5914 & 0.0762 & 0.2854\\ 0.4983 & 0.2854 & 0.1266\end{pmatrix}.$$

We use Algorithm 1 to compute eigenpair of $(\mathcal{A},\mathcal{B})$ with a random vector uniformly distributed in $(0,1)$ and we also set $\varepsilon=10^{-5}$. We get the eigenvalue 1.5477 and the eigenvector $(0.2406,0.1751,0.9547)^T$ after 34 iterations. We also give Figure 3 to show the iterations related to eigenvalues.

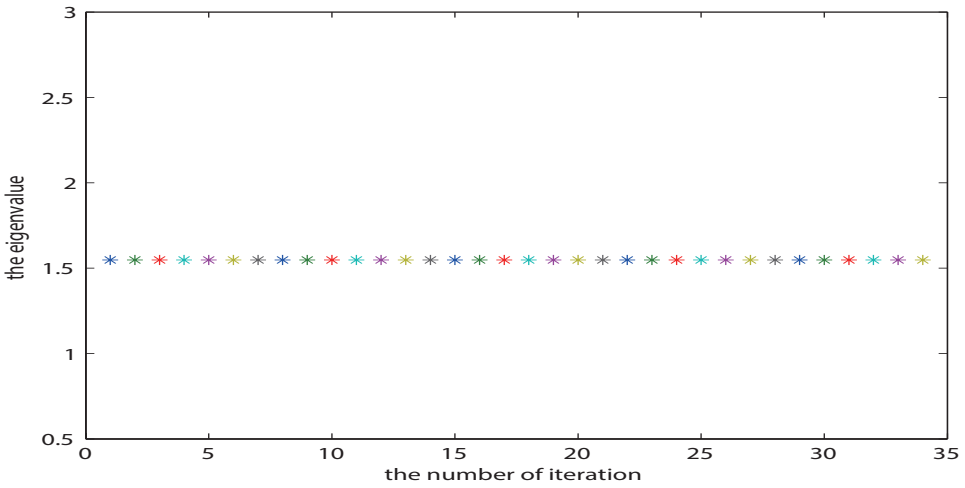

Figure 3: Eigenvalue of Algorithm 1 for *Example 3.*

Example 4. ([8]) Consider $\mathcal{A},\mathcal{B}\in S_{4,2}$, which are defined by

$$\mathcal{A}(:,:,1,1)=\begin{pmatrix}0.8147 & 0.5164\\ 0.5164 & 0.9134\end{pmatrix},\mathcal{A}(:,:,1,2)=\begin{pmatrix}0.4218 & 0.8540\\ 0.8540 & 0.9595\end{pmatrix},$$

$$\mathcal{A}(:,:,2,1)=\begin{pmatrix}0.4218 & 0.8540\\ 0.8540 & 0.9595\end{pmatrix},\mathcal{A}(:,:,2,2)=\begin{pmatrix}0.6787 & 0.7504\\ 0.7504 & 0.3922\end{pmatrix},$$

$$\mathcal{B}(:,:,1,1) = \begin{pmatrix} 1.6324 & 1.1880 \\ 1.1880 & 1.5469 \end{pmatrix}, \mathcal{B}(:,:,1,2) = \begin{pmatrix} 1.6557 & 1.4424 \\ 1.4424 & 1.9340 \end{pmatrix},$$

$$\mathcal{B}(:,:,2,1) = \begin{pmatrix} 1.6557 & 1.4424 \\ 1.4424 & 1.9340 \end{pmatrix}, \mathcal{A}(:,:,2,2) = \begin{pmatrix} 1.6555 & 1.4386 \\ 1.4386 & 1.0318 \end{pmatrix}.$$

We consider to compute eigenpairs of $(\mathcal{A}, \mathcal{B})$ with Algorithm 1 by difference starting points. The first set of starting points are all a vector of ones and the second set of starting points are random vectors uniformly distributed in $(0, 1)$. We set $\varepsilon := \{5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}\}$. The numerical results with the two different initial points are proposed in Tables 3 and 4, respectively. From the numerical results, it is clearly to see that Algorithm 1 is also efficient.

Table 3: Numerical results with starting point $(1, \ldots, 1)^T$ for *Example 4.*

ε	λ	Eigenvector	Its.
5.0e-03	0.4841	$(0.3693, 0.9293)^T$	15
1.0e-03	0.4849	$(0.3679, 0.9299)^T$	17
5.0e-04	0.4847	$(0.3669, 0.9302)^T$	19

Table 4: Numerical results with random starting point for *Example 4.*

ε	λ	Eigenvector	Its.
5.0e-03	0.4850	$(0.3701, 0.9290)^T$	14
5.0e-03	0.4839	$(0.3691, 0.9294)^T$	14
5.0e-03	0.4845	$(0.3701, 0.9290)^T$	15
1.0e-03	0.4992	$(1.0000, 0.0000)^T$	7
1.0e-03	0.4989	$(1.0000, 0.0000)^T$	15
1.0e-03	0.4847	$(0.3660, 0.9306)^T$	12
5.0e-04	0.4992	$(1.0000, 0.0000)^T$	16
5.0e-04	0.4992	$(1.0000, 0.0000)^T$	15
5.0e-04	0.4849	$(0.3672, 0.9301)^T$	18

Example 5. ([18]) Consider $\mathcal{A}$, $\mathcal{B} \in S_{4,3}$, which are defined by

$$\mathcal{A}(:,:,1,1) = \begin{pmatrix} 0.4468 & 0.4086 & 0.5764 \\ 0.4086 & 0.8176 & 0.5867 \\ 0.5764 & 0.5867 & 0.8116 \end{pmatrix}, \mathcal{A}(:,:,1,2) = \begin{pmatrix} 0.2373 & 0.5028 & 0.7260 \\ 0.5028 & 0.5211 & 0.4278 \\ 0.7260 & 0.4278 & 0.6791 \end{pmatrix},$$

$$\mathcal{A}(:,:,1,3) = \begin{pmatrix} 0.0424 & 0.0841 & 0.6220 \\ 0.0841 & 0.8181 & 0.4837 \\ 0.6220 & 0.4837 & 0.6596 \end{pmatrix}, \mathcal{A}(:,:,2,1) = \begin{pmatrix} 0.2373 & 0.5028 & 0.7260 \\ 0.5028 & 0.5211 & 0.4278 \\ 0.7260 & 0.4278 & 0.6791 \end{pmatrix},$$

$$\mathcal{A}(:,:,2,2) = \begin{pmatrix} 0.3354 & 0.7005 & 0.3154 \\ 0.7005 & 0.1068 & 0.7164 \\ 0.3154 & 0.7164 & 0.7150 \end{pmatrix}, \mathcal{A}(:,:,2,3) = \begin{pmatrix} 0.1734 & 0.5972 & 0.6791 \\ 0.5972 & 0.0605 & 0.4080 \\ 0.6791 & 0.4080 & 0.6569 \end{pmatrix},$$

$$\mathcal{A}(:,:,3,1)=\begin{pmatrix}0.0424 & 0.0841 & 0.6220\\ 0.0841 & 0.8181 & 0.4837\\ 0.6220 & 0.4837 & 0.6596\end{pmatrix},\mathcal{A}(:,:,3,2)=\begin{pmatrix}0.1734 & 0.5972 & 0.6791\\ 0.5972 & 0.0605 & 0.4080\\ 0.6791 & 0.4080 & 0.6569\end{pmatrix},$$

$$\mathcal{A}(:,:,3,3)=\begin{pmatrix}0.4897 & 0.6299 & 0.6104\\ 0.6299 & 0.0527 & 0.5803\\ 0.6104 & 0.5823 & 0.5479\end{pmatrix},\mathcal{B}(:,:,1,1)=\begin{pmatrix}2.5328 & 2.6133 & 2.7630\\ 2.6133 & 2.5502 & 2.4151\\ 2.7630 & 2.4151 & 2.3012\end{pmatrix},$$

$$\mathcal{B}(:,:,1,2)=\begin{pmatrix}2.3955 & 2.2026 & 2.8921\\ 2.2026 & 2.8852 & 2.5060\\ 2.8921 & 2.5060 & 2.2619\end{pmatrix},\mathcal{B}(:,:,1,3)=\begin{pmatrix}2.1586 & 2.8867 & 2.7372\\ 2.8867 & 2.4538 & 2.2579\\ 2.7372 & 2.2579 & 2.1332\end{pmatrix},$$

$$\mathcal{B}(:,:,2,1)=\begin{pmatrix}2.3955 & 2.2026 & 2.8921\\ 2.2026 & 2.8852 & 2.5060\\ 2.8921 & 2.5060 & 2.2619\end{pmatrix},\mathcal{B}(:,:,2,2)=\begin{pmatrix}2.9037 & 2.7948 & 2.5391\\ 2.7948 & 2.1978 & 2.2653\\ 2.5391 & 2.2653 & 2.4799\end{pmatrix},$$

$$\mathcal{B}(:,:,2,3)=\begin{pmatrix}2.6280 & 2.1537 & 2.2689\\ 2.2026 & 2.9841 & 2.2698\\ 2.8921 & 2.2698 & 2.1981\end{pmatrix},\mathcal{B}(:,:,3,1)=\begin{pmatrix}2.1586 & 2.8867 & 2.7372\\ 2.8867 & 2.4538 & 2.2579\\ 2.7372 & 2.2579 & 2.1332\end{pmatrix},$$

$$\mathcal{B}(:,:,3,2)=\begin{pmatrix}2.6280 & 2.1537 & 2.2689\\ 2.2026 & 2.9841 & 2.2698\\ 2.8921 & 2.2698 & 2.1981\end{pmatrix},\mathcal{B}(:,:,3,3)=\begin{pmatrix}2.9427 & 2.3596 & 2.7611\\ 2.3596 & 2.7011 & 2.6822\\ 2.7611 & 2.6822 & 2.6665\end{pmatrix}.$$

We use Algorithm 1 to compute eigenpairs of $(\mathcal{A},\mathcal{B})$ with an arbitrary vector uniformly distributed in $(0,1)$ and set $\varepsilon := \{5\times10^{-3}, 10^{-3}, 5\times10^{-4}\}$ respectively. The numerical results with different random starting points are reported in Table 5. From the numerical results, we can see that Algorithm 1 using less iterations.

Table 5: Numerical results with random starting points for *Example 5.*

ε	λ	Eigenvector	Its.
5.0e-03	0.2173	$(0.0704, 0.0000, 0.9975)^T$	5
5.0e-03	0.2170	$(0.0702, 0.0000, 0.9975)^T$	5
5.0e-03	0.2175	$(0.0685, 0.0046, 0.9977)^T$	9
1.0e-03	0.2171	$(0.0698, 0.0019, 0.9976)^T$	15
1.0e-03	0.2169	$(0.0703, 0.0005, 0.9975)^T$	6
1.0e-03	0.2170	$(0.0703, 0.0000, 0.9975)^T$	8
5.0e-04	0.2169	$(0.0702, 0.0007, 0.9975)^T$	9
5.0e-04	0.2169	$(0.0702, 0.0011, 0.9975)^T$	7
5.0e-04	0.2170	$(0.0700, 0.0013, 0.9975)^T$	8

4 Conclusion

In this paper, we reformulate the tensor eigenvalue complementarity problem into nonsmooth equations by Fischer-Burmeister function and present an inexact Levenberg-Marquardt method to solve it. The numerical results show the efficiency of the proposed method.

Acknowledgements

The authors are in debt to two anonymous referees for numerous insightful comments and suggestions, which have greatly improved the paper.

References

[1] B.W. Bader, T.G. Kolda, et al., MATLAB Tensor Toolbox Version 2.6, Available online, February 2015. http://www.sandia.gov/tgkolda/TensorToolbox/

[2] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[3] K.C. Chang, K. Pearson and T. Zhang, On eigenvalue problems of real symmetric tensors, *J. Math. Anal. Appl.* 350 (2009) 416–422.

[4] H. Chen and Y. Wang, On computing minimal H-eigenvalue of sign-structured tensors, *Front. Math. China* 12 (2017) 1289–1302.

[5] Z. Chen and L. Qi, A semismooth Newton method for tensor eigenvalue complementarity problem, *Comput. Optim. Appl.* 65 (2016) 109–126.

[6] Z. Chen, Q. Yang and L. Ye, Generalized eigenvalue complementarity problem for tensors, *Pac. J. Optim.* 13 (2017) 527–545.

[7] H. Dan, N. Yamashita and M. Fukushima, Convergence properties of the inexact Levenberg-Marquardt method under local error bound conditions, *Optim. Methods Softw.* 17 (2002) 605–626.

[8] J. Fan, J. Nie and A. Zhou, Tensor eigenvalue complementarity problems, *Math. Program.* 170 (2018) 507–539.

[9] J. Fan and J. Pan, On the convergence rate of the inexact Levenberg-Marquardt method, *J. Ind. Manag. Optim.* 7 (2017) 199–210.

[10] A. Fischer, A special Newton-type optimization method, *Optimization* 24 (1992) 269–284.

[11] F. Facchinei and C. Kanzow, A nonsmooth inexact Newton method for the solution of large-scale nonlinear complementarity problems, *Math. Program.* 76 (1997) 493–512.

[12] S. Hu, Z. Huang and L. Qi, Finding the extreme Z-eigenvalues of tensors via a sequential semidefinite programming method, *Numer. Linear Algebra Appl.* 20 (2013) 972–984.

[13] W. Hu, L. Lu, C. Yin and G. Yu, A smoothing Newton method for tensor eigenvalue complementarity problems, *Pac. J. Optim.* 13 (2017) 243–253.

[14] J.J. Júdice, H.D. Sherali and I.M. Ribeiro, The eigenvalue complementarity problem, *Comput. Optim. Appl.* 37 (2007) 139–156.

[15] E. Kofidis and P.A. Regalia, On the rank-1 approximation of higher-order supersymmetric tensors, *SIAM J. Matrix Anal. Appl.* 23 (2002) 863–884.

[16] E.W. Karas, S.A. Santos and B. F. Svaiter, Algebraic rules for computing the regularization parameter of the Levenberg-Marquardt method. *Comput. Optim. Appl.* 65 (2016) 723-751.

[17] J. Lqbal, A. Lqbal and M. Arif, Levenberg-Marquardt method for solving systems of absolute value equations, *J. Comput. Appl. Math.* 282 (2015) 134–138.

[18] C. Ling, H. He and L. Qi, On the cone eigenvalue complementarity problem for higher-order tensors, *Comput. Optim. Appl.* 63 (2016) 143-168.

[19] C. Ling, H. He and L. Qi, Higher-degree eigenvalue complementarity problems for tensors, *Comput. Optim. Appl.* 64 (2015) 149–176.

[20] M. Ng, L. Qi and G. Zhou, Finding the largest eigenvalue of a nonnegative tensor, *SIAM J. Matrix Anal. Appl.* 31 (2009) 1090–1099.

[21] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, New York, 1999.

[22] M. Queiroz, J.J. Júdice and J.C. Humes, The symmetric eigenvalue complementarity problem, *Math. Comp.* 73 (2004) 1849–1863.

[23] L. Qi, Eigenvalues of a real supersymmetric tensor, *J. Symbolic Comput.* 40 (2005) 1302–1324.

[24] Y. Song and L. Qi, Eigenvalue analysis of constrained minimization problem for homogeneous polynomial, *J. Global Optim.* 64 (2016) 563–575.

[25] D. Sun and L. Qi, On NCP-Functions, *Comput. Optim. Appl.* 13 (1999) 201–220.

[26] K. Ye and S. Hu, Inverse eigenvalue problems for tensors, *Commun. Math. Sci.* 15 (2017) 1627–1649.

[27] G. Yu, Y. Song, Y. Xu and Z. Yu, Spectral projected gradient methods for generalized tensor eigenvalue complementarity problems, *Numer. Algorithms* 80 (2019) 1181–1201.

[28] L. Zhang and L. Qi, Linear convergence of an algorithm for computing the largest eigenvalue of a nonnegative tensor, *Numer. Linear Algebra Appl.* 19 (2012) 830–841.

Manuscript received 15 November 2017
revised 11 April 2018
accepted for publication 17 July 2018

Huanhuan Li
School of Mathematics and Statistics
Qingdao University, Qingdao, 266071, China
E-mail address: 630391052@qq.com

Shouqiang Du
School of Mathematics and Statistics
Qingdao University, Qingdao, 266071, China
E-mail address: sqdu@qdu.edu.cn

Yingxiao Wang
School of Mathematics and Statistics
Qingdao University, Qingdao, 266071, China
E-mail address: 772989436@qq.com

Miao Chen
School of Mathematics and Statistics
Qingdao University, Qingdao, 266071, China
E-mail address: 1551244396@qq.com