Linear Algebra and its Applications 439 (2013) 3713–3733

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

The solution methods for the largest eigenvalue (singular value) of nonnegative tensors and convergence analysis

Zhongming Chen [a], Liqun Qi [b,1], Qingzhi Yang [a,*,2], Yuning Yang [a]

[a] School of Mathematical Sciences and LPMC, Nankai University, Tianjin 300071, PR China
[b] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong

ARTICLE INFO

Article history:
Received 19 July 2012
Accepted 18 September 2013
Available online 18 October 2013
Submitted by E. Tyrtyshnikov

MSC:
74B99
15A18
15A69

Keywords:
Nonnegative tensor
Eigenvalue
Singular value
Perturbation
Algorithm
Convergence
Complexity

ABSTRACT

In this paper we study two solution methods for finding the largest eigenvalue (singular value) of general square (rectangular) nonnegative tensors. For a positive tensor, one can find the largest eigenvalue (singular value) based on the properties of the positive tensor and the power-type method. While for a general nonnegative tensor, we use a series of decreasing positive perturbations of the original tensor and repeatedly recall power-type method for finding the largest eigenvalue (singular value) of a positive tensor with an inexact strategy. We prove the convergence of the method for the general nonnegative tensor. Under a certain assumption, the computing complexity of the method is established. Motivated by the interior-point method for the convex optimization, we put forward a one-step inner iteration power-type method, whose convergence is also established under certain assumption. Additionally, by using embedding technique, we show the relationship between the singular values of the rectangular tensor and the eigenvalues of related square tensor, which suggests another way for finding the largest singular value of nonnegative rectangular tensor besides direct power-type method for this problem. Finally, numerical examples of our algorithms are reported, which demonstrate the

* Corresponding author.
E-mail addresses: czm183015@mail.nankai.edu.cn (Z. Chen), maqilq@polyu.edu.hk (L. Qi), qz-yang@nankai.edu.cn (Q. Yang), nk0310145@gmail.com (Y. Yang).

[1] This author's work is supported by the Hong Kong Grant Council (Grant No. PolyU 501909, 502510, 502111 and 501212).
[2] This author's work is supported by the National Natural Science Foundation of China (Grant No. 11271206), Doctoral fund of Chinese Ministry of Education (Grant No. 20120031110024).

0024-3795/$ – see front matter
http://dx.doi.org/10.1016/j.laa.2013.09.027

convergence behaviors of our methods and show that the algorithms presented are promising.

1. Introduction

Let R be the real field. A real order m dimension n tensor $\mathcal{A}$ consists of n^m real entries:

$$a_{i_1\cdots i_m} \in R,$$

where $i_j = 1,\ldots,n$ for $j = 1,\ldots,m$. $\mathcal{A}$ is called nonnegative if $a_{i_1\cdots i_m} \geqslant 0$ and $\mathcal{A}$ is called symmetric as its entries $a_{i_1\cdots i_m}$ are invariant under any permutation of their indices. Like the square matrix, this class of tensors can be regarded as "square" tensors.

If there are a complex number λ and a nonzero complex vector x that are solutions of the following homogeneous polynomial equations:

$$\mathcal{A}x^{m-1} = \lambda x^{[m-1]},$$

then λ is called the eigenvalue of $\mathcal{A}$ and x the eigenvector of $\mathcal{A}$ associated with λ, where $\mathcal{A}x^{m-1}$ and $x^{[m-1]}$ are vectors, whose ith components are

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

respectively. This definition was first introduced by Qi [21] where he assumed that $\mathcal{A}$ is an order m dimension n symmetric tensor and m is even. If λ and x are restricted in the real field, then (λ, x) is called H-eigenpair. Independently, Lim [15] gave such a definition but restricted x to be a real vector and λ to be a real number. Here we follow the definition due to Chang et al., where they gave the general definition as above [1]. However, we mention that the ideas of eigenvalues of tensors may have been raised earlier. For example, the largest eigenvalue of a symmetric tensor is also a stationary point of the best rank-one approximation in the sense of the Frobenius norm [5].

Unlike matrices, eigenvalue problems for high-order tensors are nonlinear. As far as we know, applications of eigenvalues of high-order tensors include best rank-one approximation in data analysis [8,22], higher-order Markov chains [19], pagerank [16], hypergraph [6,9,14] and positive definiteness of even-order multivariate forms in automatic control [18].

Recently, eigenvalue problems for high-order tensors have gained special attention in the realm of numerical multilinear algebra. In particular, Chang, Person, Zhang [1] generalized the Perron–Frobenius theorem (see [11,24]) from the nonnegative irreducible matrix to the nonnegative irreducible tensor. Yang and Yang [26,27] generalized the weak Perron–Frobenius theorem to general nonnegative tensors. Later, the singular values of a real rectangular tensor was systematically studied by Chang, Qi and Zhou [3], who extended the Perron–Frobenius theorem to the nonnegative rectangular tensors. Yang and Yang [28] showed that the weak Perron–Frobenius theorem still keeps valid for the nonnegative rectangular tensors. Ng, Qi, and Zhou [19] proposed an iterative method for finding the largest eigenvalue of a nonnegative irreducible tensor, which is an extension of the Collatz method for computing the spectral radius of an irreducible matrix. Their method was called the NQZ method. Another method based on the NQZ method was proposed by Liu, Zhou and Ibrahim [17]. Pearson [20] introduced the essentially positive tensors and she conjectured that the NQZ method would converge if the tensor is essentially positive with even order. Chang, Pearson and Zhang [2] defined the primitive tensors, which include the class of essentially positive tensors as the special case, and they established the convergence of the NQZ method for primitive tensors. Zhang and Qi [33] established the linear convergence of the NQZ method for essentially positive tensors. Hu, Huang and Qi [13]

obtained the global R-linear convergence of the modified version of the NQZ method for weakly irreducible nonnegative tensors which were introduced by Friedland, Gaubert and Han in [10]. On the other hand, the method for finding the largest singular value was put forward in [3]. This method was called the CQZ method. Zhou, Caccetta and Qi [31] established the convergence of a modified version of the CQZ method for any irreducible nonnegative rectangular tensor. Yang and Yang [28] established its convergence by introducing the notion of primitive rectangular tensors. Recently, Ragnarsson and Van Loan [23] built the relationship between singular values of general tensors and eigenvalues of corresponding symmetric embedding tensors, which implies that one can solve the largest singular value of a nonnegative rectangular tensor with the help of the method for finding the largest eigenvalue of a nonnegative square tensor.

In this paper, we focus on finding the largest eigenvalue (singular value) of general nonnegative tensors. The numerical results show neither the NQZ method nor the CQZ method works for some nonnegative tensors. To the end, we propose the inexact power-type methods for finding the largest eigenvalue or singular value by using a series of decreasing positive perturbations of the original nonnegative tensor. For each positive perturbed tensor, it is not necessary to get the exact spectral radius, it only needs find the solution with certain precision, which can be realized in practice. The convergence of the inexact power-type algorithm is established for general nonnegative square tensors. And the computing complexity of the algorithm is given under certain assumption. Besides, we propose another algorithm for finding the spectral radius of a nonnegative square tensor. As we have known, some interior-point methods only execute one inner iteration. Motivated by this idea, we present the one-step inner iteration power-type method, which is easier to perform. The convergence of this algorithm is proven under weak positivity condition. Since it has been known that there are closed connections between the singular values of a tensor and the eigenvalues of its symmetric embedding tensor, for a rectangular or general tensor, we can transform the singular values problem to the eigenvalues problem by constructing a new related square tensor. However, the NQZ method does not work for the new constructed square tensor in many cases, while our inexact power-type method does well. Moreover, the numerical results show that it is more efficient to find the largest singular value in this way than to do it directly, such as done in [3].

This paper is organized as follows. We first review some preliminaries which are useful for further analysis in Section 2. In Section 3, we recall some theorems and algorithms for finding the largest eigenvalue and singular value and give the inexact power-type method in the case of square and rectangular tensor. The one-step inner iteration power-type method is put forward in Section 4. In Section 5, we illustrate that one can convert the singular value problem to the eigenvalue problem by constructing a new related square tensor. In Section 6, some numerical results are reported.

We first add a comment on the notation that is used in the sequel. Vectors are written as lowercase letters $(x, y, \ldots)$, matrices correspond to italic capitals $(A, B, \ldots)$, and tensors are written as calligraphic capitals $(\mathcal{A}, \mathcal{B}, \ldots)$. The entry with row index i and column index j in a matrix A, i.e. $(A)_{ij}$ is symbolized by a_{ij} (also $(\mathcal{A})_{i_1\cdots i_p, j_1\cdots j_q} = a_{i_1\cdots i_p, j_1\cdots j_q}$). The symbol $|\cdot|$ used on a matrix A (or tensor $\mathcal{A}$) means that $(|A|)_{ij} = |a_{ij}|$ (or $(|\mathcal{A}|)_{i_1\cdots i_p, j_1\cdots j_q} = |a_{i_1\cdots i_p, j_1\cdots j_q}|$). R^n_+ (R^n_{++}) denotes the cone $\{x \in R^n \mid x_i \geqslant (>)\, 0,\ i = 1, \ldots, n\}$. The symbol $A \geqslant (>, \leqslant, <) B$ means that $a_{ij} \geqslant (>, \leqslant, <)\, b_{ij}$ for every i, j and it is the same for rectangular tensors.

2. Preliminaries

First let's recall the notion of the rectangular tensor. Assume that p, q, m and n are positive integers, and $m, n \geqslant 2$. A real rectangular tensor $\mathcal{A}$ consists of $m^p n^q$ real entries

$$a_{i_1\cdots i_p j_1\cdots j_q} \in R,$$

where $i_k = 1, \ldots, m$, $k = 1, \ldots, p$, and $j_k = 1, \ldots, n$, $k = 1, \ldots, q$. We also call $\mathcal{A}$ a real (p, q)th order $m \times n$ dimensional rectangular tensor, or simply a real rectangular tensor. When $p = q = 1$, $\mathcal{A}$ reduces to a real $m \times n$ rectangular matrix.

The definition of singular values of a real rectangular tensor can be seen in [3]. Let $\mathcal{A}x^{p-1}y^q$ be a vector in R^m such that

$$\left(\mathcal{A}x^{p-1}y^{q}\right)_i = \sum_{i_2,\ldots,i_p=1}^{m}\ \sum_{j_1,\ldots,j_q=1}^{n} a_{ii_2\cdots i_p j_1\cdots j_q} x_{i_2}\cdots x_{i_p} y_{j_1}\cdots y_{j_q},$$

where $i = 1,\ldots,m$. Similarly, let $\mathcal{A}x^p y^{q-1}$ be a vector in R^n such that

$$\left(\mathcal{A}x^{p}y^{q-1}\right)_j = \sum_{i_1,\ldots,i_p=1}^{m}\ \sum_{j_2,\ldots,j_q=1}^{n} a_{i_1\cdots i_p j j_2\cdots j_q} x_{i_1}\cdots x_{i_p} y_{j_2}\cdots y_{j_q},$$

where $j = 1,\ldots,n$. Denote $M = p+q$. If there are a number $\lambda \in C$, vectors $x \in C^m \setminus \{0\}$ and $y \in C^n \setminus \{0\}$ such that

$$\begin{cases} \mathcal{A}x^{p-1}y^{q} = \lambda x^{[M-1]}, \\ \mathcal{A}x^{p}y^{q-1} = \lambda y^{[M-1]}, \end{cases}$$

then λ is called the singular value of $\mathcal{A}$ and (x, y) are the left and right eigenvectors of $\mathcal{A}$, associated with λ, respectively. If $\lambda \in R$, $x \in R^m$ and $y \in R^n$, then we say that λ is an H-singular value of $\mathcal{A}$, and (x, y) are the left and right H-eigenvectors associated with λ. If a singular value is not an H-singular value, we call it an N-singular value of $\mathcal{A}$.

In the following analysis, the notions below may be used.

Definition 2.1. (See [27].) The spectral radius of tensor $\mathcal{A}$ is defined as

$$\rho(\mathcal{A}) = \max\left\{|\lambda|\colon\ \lambda \text{ is an eigenvalue of } \mathcal{A}\right\}.$$

Since the tensor considered is nonnegative, Chang et al. [1] proved the existence of eigenvalue and Yang and Yang [27] showed that $\rho(\mathcal{A})$ is an eigenvalue. At this point, we see that Definition 2.1 is well defined. Similarly, one may define the largest singular value of the rectangular tensors. We call $r\rho(\mathcal{A}) = \max\{|\lambda|\colon\ \lambda \text{ is the singular value of } \mathcal{A}\}$ the like-spectral radius of $\mathcal{A}$. Yang and Yang [28] proved the existence of singular value and showed that $r\rho(\mathcal{A})$ is a singular value when $\mathcal{A}$ is nonnegative. At this point, this definition is well defined.

Definition 2.2. (See Definition 2.1 of [1].) A tensor $\mathcal{C} = (c_{i_1\cdots i_m})$ of order m dimension n is called reducible, if there exists a nonempty proper index subset $I \subset \{1,\ldots,n\}$ such that

$$c_{i_1\cdots i_m} = 0, \quad \forall i_1 \in I,\ \forall i_2,\ldots,i_m \notin I.$$

If $\mathcal{C}$ is not reducible, then we call $\mathcal{C}$ irreducible.

The irreducibility of a rectangular tensor is defined as follow:

Definition 2.3. (See Definition 1 of [3].) A nonnegative rectangular tensor $\mathcal{A}$ is called irreducible if all the square tensors $\mathcal{A}(\cdot, f_j^q)$, $j = 1,\ldots,n$, and $\mathcal{A}(e_i^p, \cdot)$, $i = 1,\ldots,m$, are irreducible in the sense of the following definition.

Let $P = \{(x_1,\ldots,x_n) \mid x_i \geqslant 0\}$ be a positive cone in R^n. For a nonnegative tensor $\mathcal{A}$, we define its associated nonlinear map $T_{\mathcal{A}} : P \to P$ by $T_{\mathcal{A}}(x) = (\mathcal{A}x^{m-1})^{[\frac{1}{m-1}]}$.

Definition 2.4. (See Definition 2.6 of [2].) A nonnegative irreducible m-order n-dimensional tensor $\mathcal{A}$ is called primitive if $T_{\mathcal{A}}$ does not have a nontrivial invariant set S on ∂P. ($\{0\}$ is the trivial invariant set.)

Definition 2.5. (See Definition 3.1 of [20].) A tensor $\mathcal{C} = (c_{i_1\cdots i_m})$ of order m dimension n is called essentially positive, if $\mathcal{C}x^{m-1} > 0$ for any nonzero $x \geqslant 0$.

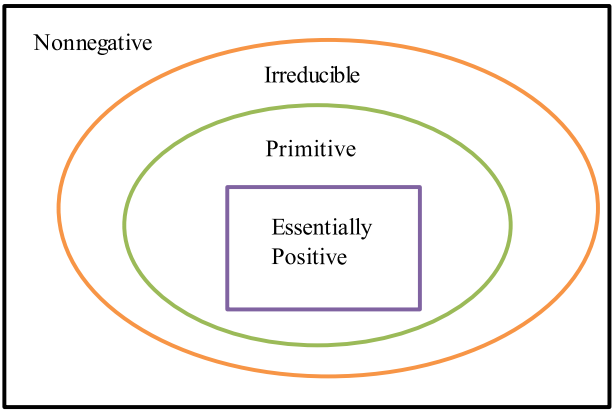

Fig. 1. Relationships of the three classes of nonnegative tensors.

For the essential positive tensor, we have the following proposition.

Proposition 2.1. *(See Theorem 3.2 of [20].) A nonnegative m-order n-dimensional tensor $\mathcal{A}$ is essential positive if and only if $a_{ij\cdots j} > 0$ for any $i, j \in \{1, 2, \ldots, n\}$.*

The relationships between these three classes of nonnegative tensors can be summarized in Fig. 1. For a more detailed discussion, one can refer to [13].

3. Inexact power-type algorithm and convergence analysis

In this section, we will give an inexact power-type method for finding the largest eigenvalue (singular value), and then establish the convergence of the method for general nonnegative square (rectangular) tensors. In addition, under certain assumption, we give the computing complexity of the algorithm for the square case.

3.1. The square tensor case

In this subsection, our main contribution is to propose a convergent algorithm for a general nonnegative square tensor by using inexact techniques, which greatly expands the scope of application of the NQZ method proposed in [19].

First we recall the power-type algorithm (the NQZ method) for an irreducible nonnegative square tensor in [19].

Algorithm 3.1.

Step 1. Choose $x^{(0)} > 0$, $x^{(0)} \in R^n$. Let $y^{(0)} = \mathcal{A}(x^{(0)})^{m-1}$ and set $k = 0$.
Step 2. Compute

$$x^{(k+1)} = \left(y^k\right)^{\left[\frac{1}{m-1}\right]} \Big/ \left\|\left(y^k\right)^{\left[\frac{1}{m-1}\right]}\right\|, \qquad y^{(k+1)} = \mathcal{A}\left(x^{(k+1)}\right)^{m-1},$$

$$\overline{\lambda}_{k+1} = \max_i \frac{y_i^{(k+1)}}{(x_i^{(k+1)})^{m-1}}, \qquad \underline{\lambda}_{k+1} = \min_i \frac{y_i^{(k+1)}}{(x_i^{(k+1)})^{m-1}}.$$

Step 3. If $\overline{\lambda}_{k+1} = \underline{\lambda}_{k+1}$, stop. Otherwise, replace k by $k+1$ and go to Step 2.

If the iteration stops with some k, then we find the largest eigenvalue $\rho(\mathcal{A}) = \overline{\lambda}_k = \underline{\lambda}_k$ and the associated eigenvector $x^* = x^{(k)}$. In fact,

$$\mathcal{A}(x^*)^{m-1} = \mathcal{A}(x^{(k)})^{m-1} = \rho(\mathcal{A})(x^{(k)})^{[m-1]} = \rho(\mathcal{A})(x^*)^{[m-1]}.$$

Otherwise, the following Lemma 3.1 proved the convergence of $\{\overline{\lambda}_k\}$ ($\{\underline{\lambda}_k\}$, respectively) under the condition that $\mathcal{A}$ is primitive.

Lemma 3.1. *(See Propositions 5.1 and 5.2 of [2].) If $\mathcal{A}$ is primitive, then $\{\overline{\lambda}_k\}$ is monotonically decreasing and $\{\underline{\lambda}_k\}$ is monotonically increasing and both of the sequences converge to $\rho(\mathcal{A})$.*

What's more, the linear convergence could also be derived if the condition becomes stronger.

Lemma 3.2. *(See [29,33].) If $\mathcal{A}$ is essentially positive, then both $\{\overline{\lambda}_k\}$ and $\{\underline{\lambda}_k\}$ converge linearly to $\rho(\mathcal{A})$. In detail, for $k = 0, 1, \ldots,$*

$$0 \leqslant \overline{\lambda}_k - \rho(\mathcal{A}) \leqslant \overline{\lambda}_{k+1} - \underline{\lambda}_{k+1} \leqslant \alpha(\overline{\lambda}_k - \underline{\lambda}_k),$$
$$0 \leqslant \rho(\mathcal{A}) - \underline{\lambda}_k \leqslant \overline{\lambda}_{k+1} - \underline{\lambda}_{k+1} \leqslant \alpha(\overline{\lambda}_k - \underline{\lambda}_k),$$

where $\alpha = 1 - \beta/\overline{R}$, $\beta = \min_{i,\, j\in\{1,2,\ldots,n\}} a_{ij\cdots j}$, $\overline{R} = \max_i \sum_{i_2,\ldots,i_m}^{n} a_{ii_2\cdots i_m}$.

It is well known that the algorithm for computing the largest eigenvalue of nonnegative square tensor fails to converge for general situation. The nonnegative weak irreducibility or essential positivity of the tensor must be satisfied to guarantee the convergence or linear convergence of the algorithm. It is natural to study the method for solving the largest eigenvalue (singular value) of a general nonnegative tensor.

Let's recall a proposition, which is the base of our following inexact algorithm.

Theorem 3.1. *(See Theorem 2.3 of [27].) Suppose $\mathcal{A} \geqslant 0$. Let $\{\mathcal{A}_k\}$ be a positive tensor sequence and it converges to $\mathcal{A}$, then $\rho(\mathcal{A}_k) \to \rho(\mathcal{A})$.*

In their preprint [34], Zhou et al. presented an algorithm based on this proposition to calculate the largest eigenvalue of a nonnegative square tensor. However, there are unsuitable requirements in their algorithm, i.e., they assumed that for every fixed $\mathcal{A}_k$, $\rho(\mathcal{A}_k)$ must be exactly calculated. Of course, it is impossible to get this exact solution in general within finite iterations and is also not necessary actually, just like in many perturbation-class algorithms. For practical purpose, we propose an inexact perturbation-type method and establish its convergence. In our method, the initial point in every inner loop is specified as the terminal point in the former inner loop with some precision.

For any given tolerance $\epsilon > 0$, it is easy to know from Lemma 3.2 that a required solution can be obtained within $\lceil \ln(\frac{\epsilon}{\overline{\lambda}_0 - \underline{\lambda}_0}) / \ln\alpha \rceil + 1$ steps.

Let $\mathcal{E}$ be the all-ones m-order n-dimensional tensor. Motivated by the inexact technique such as in [12,30], we present an inexact power-type algorithm for finding the largest eigenvalue of $\mathcal{A}$ in the following.

Algorithm 3.2.

Step 1. Take a positive sequence $\{\epsilon_k\}$ such that $\sum_{k=1}^{\infty} \epsilon_k < \infty$. Given a $\theta \in (0, 1)$, set $\tau_1 = \theta$, $\mathcal{A}_1 = \mathcal{A} + \tau_1\mathcal{E}$. Choose $x^{(0)} > 0, x^{(0)} \in R^n$. Let $y^{(0)} = \mathcal{A}_1(x^{(0)})^{m-1}$. Let $y_1^{(0)} = y^{(0)}$. $l = 1$.

Step 2. Compute

$$x_l^{(k)} = \left(y_l^{(k-1)}\right)^{[\frac{1}{m-1}]} / \left\|\left(y_l^{(k-1)}\right)^{[\frac{1}{m-1}]}\right\|, \qquad y_l^{(k)} = \mathcal{A}_l\left(x_l^{(k)}\right)^{m-1},$$
$$\overline{\lambda}_k^l = \max_i \frac{(y_l^{(k)})_i}{(x_l^{(k)})_i^{m-1}}, \qquad \underline{\lambda}_k^l = \min_i \frac{(y_l^{(k)})_i}{(x_l^{(k)})_i^{m-1}},$$

for $k = 1, 2, \ldots$, until $\overline{\lambda}_k^l - \underline{\lambda}_k^l < \epsilon_l$, denote this k as $k(l)$ and

$$x_{l+1}^{(0)} = x_l^{(k(l))}, \qquad \overline{\lambda}^l = \overline{\lambda}_{k(l)}^l, \qquad \underline{\lambda}^l = \underline{\lambda}_{k(l)}^l.$$

Step 3. $l = l + 1$, $\tau_l = \theta\tau_l$, $\mathcal{A}_l = \mathcal{A} + \tau_l\mathcal{E}$. Set $y_l^{(0)} = \mathcal{A}_l(x_l^{(0)})^{m-1}$. Go to Step 2.

We can see that Algorithm 3.2 can be complemented in practice as Lemma 3.3 shows.

Lemma 3.3. *In Algorithm 3.2, the inner iteration in Step 2 terminates in a finite number of steps.*

Proof. Since $\mathcal{A}_l$ is a positive tensor, it is primitive. By Lemma 3.1, this proposition holds. □

The convergence of Algorithm 3.2 could also be ensured for any nonnegative tensor.

Theorem 3.2. *Suppose $\mathcal{A}$ is an m-order n-dimensional nonnegative tensor. The two sequences generated by Algorithm 3.2, $\{\overline{\lambda}^l\}$ and $\{\underline{\lambda}^l\}$, converge to $\rho(A)$ respectively.*

Proof. Since $\mathcal{A}_{l+1} < \mathcal{A}_l$, we have $\rho(\mathcal{A}_{l+1}) \leqslant \rho(\mathcal{A}_l)$ due to Lemma 3.3 in [27]. So we get the following by the algorithm

$$\overline{\lambda}^l \geqslant \rho(\mathcal{A}_l) \geqslant \rho(\mathcal{A}_{l+1}) \geqslant \overline{\lambda}^{l+1} - \epsilon_{l+1},$$

from which one has

$$\overline{\lambda}^{l+1} \leqslant \overline{\lambda}^l + \epsilon_{l+1}.$$

Since both $\{\overline{\lambda}^l\}$ and $\{\epsilon_l\}$ are nonnegative sequences and $\sum_{k=1}^{\infty} \epsilon_k < \infty$, it follows that $\{\overline{\lambda}^l\}$ is convergent. From Theorem 3.1 we get that

$$\lim_{l\to\infty} \overline{\lambda}^l = \lim_{l\to\infty} \rho(\mathcal{A}_l) = \rho(\mathcal{A}).$$

Similarly one has that $\lim_{l\to\infty} \underline{\lambda}^l = \rho(\mathcal{A})$. □

In order to give the complexity analysis of Algorithm 3.2, we need the following assumption which holds when $m = 2$.

Assumption 3.1. Let $\mathcal{A}$ be an m-order n-dimensional nonnegative tensor, $\mathcal{A}_k = \mathcal{A} + \tau_k\mathcal{E}$ and $|\rho(\mathcal{A}_k) - \rho(\mathcal{A})| \leqslant C\tau_k$, where C is a positive constant.

In the next we give three examples which satisfy the assumption.

Example 3.1. Let $\mathcal{A} = \mathcal{O}_{3\times3\times3}$ be a 3-order 3-dimensional zero tensor, then $\mathcal{A}_k = \tau_k\mathcal{E}$. It is easy to see that $\rho(\mathcal{A}) = 0$, $\rho(\mathcal{A}_k) = 9\tau_k$. So

$$\left|\rho(\mathcal{A}_k) - \rho(\mathcal{A})\right| \leqslant 9\tau_k.$$

Example 3.2. Let $\mathcal{A}$ be an m-order n-dimensional diagonal tensor such that $a_{i\cdots i} = i$, $i = 1, \ldots, n$ and others are zero. It is easy to see that $\rho(\mathcal{A}) = n$, $\rho(\mathcal{A}_k) = n + n^{m-1}\tau_k$. So

$$\left|\rho(\mathcal{A}_k) - \rho(\mathcal{A})\right| \leqslant n^{m-1}\tau_k.$$

Example 3.3. Let $\mathcal{A} = \epsilon_{3\times3\times3}$ be a 3-order 3-dimensional all-ones tensor, then $\mathcal{A}_k = (1 + \tau_k)\mathcal{E}$. It is easy to see that $\rho(\mathcal{A}) = 9$, $\rho(\mathcal{A}_k) = 9 + 9\tau_k$. So

$$\left|\rho(\mathcal{A}_k) - \rho(\mathcal{A})\right| \leqslant 9\tau_k.$$

From the examples above we see that a tensor which satisfies Assumption 3.1 may not be irreducible or essentially positive. It deserves further discussion for sufficient conditions ensuring Assumption 3.1.

Below we will establish the computing complexity of Algorithm 3.2 with a specific sequence $\{\epsilon_k\}$ under Assumption 3.1.

Theorem 3.3. *In Algorithm 3.2, set $\epsilon_k = c\theta^k$, where $c > 0$ is a constant. Then for any given $\epsilon > 0$, one may get a required approximate solution by using Algorithm 3.2 within K iteration steps, where $K = (\lceil \ln(\frac{\epsilon}{c+C})/\ln\theta\rceil + 1)\lceil \ln(\frac{c\theta}{c+(1-\theta)\rho(\mathcal{E})})/\ln d_2\rceil$. In particular, if we choose $c = \rho(\mathcal{E})$, then $K = (\lceil \ln(\frac{\epsilon}{\rho(\mathcal{E})+C})/\ln\theta\rceil + 1)\lceil \ln(\frac{\theta}{2-\theta})/\ln d_2\rceil$.*

Proof. It has been known from Lemma 3.2 that for a fixed ϵ_{l+1}, one can get the approximate solution with required precision within $K_{l+1} = \lceil \ln(\frac{\epsilon_{l+1}}{\overline{\lambda}_0^{l+1} - \underline{\lambda}_0^{l+1}})/\ln\alpha_{l+1}\rceil + 1$ steps. Since $\epsilon_k = c\theta^k$,

$$0 \leqslant \overline{\lambda}^l_{k(l)} - \underline{\lambda}^l_{k(l)} \leqslant \epsilon_l = c\theta^l.$$

In Algorithm 3.2, we choose $x^{(0)}_{l+1} = x^{(k(l))}_l$, and

$$\overline{\lambda}_0^{l+1} = \max_i \frac{[\mathcal{A}_{l+1}(x^{(0)}_{l+1})^{m-1}]_i}{(x^{(0)}_{l+1})_i^{m-1}}, \qquad \underline{\lambda}_0^{l+1} = \min_i \frac{[\mathcal{A}_{l+1}(x^{(0)}_{l+1})^{m-1}]_i}{(x^{(0)}_{l+1})_i^{m-1}}.$$

However,

$$\mathcal{A}_l = \mathcal{A} + \tau_1\mathcal{E}, \quad \tau_l = \theta^l,$$

so

$$\mathcal{A}_{l+1} = \mathcal{A}_l - \theta^l(1-\theta)\mathcal{E} \leqslant \mathcal{A}_l.$$

It is obvious that $x^{(k)}_l$ is positive for any l, k since $\mathcal{A}_l > 0$ and $x^{(0)} > 0$. So for any $i \in \{1, 2, \ldots, n\}$, we have

$$\frac{[\mathcal{A}_{l+1}(x^{(0)}_{l+1})^{m-1}]_i}{(x^{(0)}_{l+1})_i^{m-1}} \leqslant \frac{[\mathcal{A}_l(x^{(0)}_{l+1})^{m-1}]_i}{(x^{(0)}_{l+1})_i^{m-1}} = \frac{[\mathcal{A}_l(x^{(k(l))}_l)^{m-1}]_i}{(x^{(k(l))}_l)_i^{m-1}},$$

from which we can derive

$$\overline{\lambda}_0^{l+1} \leqslant \overline{\lambda}^l_{k(l)}, \qquad \underline{\lambda}_0^{l+1} \leqslant \underline{\lambda}^l_{k(l)}.$$

On the other hand,

$$\begin{aligned}
\underline{\lambda}_0^{l+1} &= \min_i \frac{[\mathcal{A}_{l+1}(x^{(0)}_{l+1})^{m-1}]_i}{(x^{(0)}_{l+1})_i^{m-1}} \\
&\geqslant \min_i \frac{[\mathcal{A}_l(x^{(0)}_{l+1})^{m-1}]_i}{(x^{(0)}_{l+1})_i^{m-1}} - \max_i \frac{\theta^l(1-\theta)[\mathcal{E}(x^{(0)}_{l+1})^{m-1}]_i}{(x^{(0)}_{l+1})_i^{m-1}} \\
&\geqslant \underline{\lambda}^l_{k(l)} - \theta^l(1-\theta)\rho(\mathcal{E}),
\end{aligned}$$

where $\rho(\mathcal{E})$ is the spectral radius of tensor $\mathcal{E}$. Thereby one obtains

$$\begin{aligned}
\overline{\lambda}_0^{l+1} - \underline{\lambda}_0^{l+1} &\leqslant \overline{\lambda}^l_{k(l)} - \underline{\lambda}^l_{k(l)} + \theta^l(1-\theta)\rho(\mathcal{E}) \\
&\leqslant c\theta^l + \theta^l(1-\theta)\rho(\mathcal{E}).
\end{aligned}$$

So

$$\ln\left(\frac{\epsilon_{l+1}}{\overline{\lambda}_0^{l+1}-\underline{\lambda}_0^{l+1}}\right)\geqslant\ln\left(\frac{c\theta}{c+(1-\theta)\rho(\mathcal{E})}\right).$$

In particular, if we choose $c=\rho(\mathcal{E})$, then we have

$$\ln\left(\frac{\epsilon_{l+1}}{\overline{\lambda}_0^{l+1}-\underline{\lambda}_0^{l+1}}\right)\geqslant\ln\left(\frac{\theta}{2-\theta}\right).$$

By the definition of α_l and the expression of A_l we have that

$$\alpha_l=1-\frac{\beta+\theta^l}{\overline{R}+n^{m-1}\theta^l}.$$

Denote

$$d_1=\min\left\{1-\frac{\beta+\theta}{\overline{R}+n^{m-1}\theta},1-\frac{\beta+\theta^L}{\overline{R}+n^{m-1}\theta^L}\right\},$$
$$d_2=\max\left\{1-\frac{\beta+\theta}{\overline{R}+n^{m-1}\theta},1-\frac{\beta+\theta^L}{\overline{R}+n^{m-1}\theta^L}\right\},$$

where $L=\lceil\ln(\frac{\epsilon}{c+C})/\ln\theta\rceil+1$ is an upper bound of the outer iteration. Thereby one obtains

$$d_1\leqslant\alpha_l\leqslant d_2\quad\Rightarrow\quad\frac{1}{d_2}\leqslant\frac{1}{\alpha_l}\leqslant\frac{1}{d_1}.$$

Thus we arrive at $K_{l+1}\leqslant\ln(\frac{c\theta}{c+(1-\theta)\rho(\mathcal{E})})/\ln d_2$, an upper bound of the iterations at the lth step. Since we know the outer iterations does not exceed $L=\lceil\ln(\frac{\epsilon}{c+C})/\ln\theta\rceil+1$, the total iterations are bounded by

$$LK_{l+1}\leqslant\left(\left\lceil\ln\left(\frac{\epsilon}{c+C}\right)\Big/\ln\theta\right\rceil+1\right)\left\lceil\ln\left(\frac{c\theta}{c+(1-\theta)\rho(\mathcal{E})}\right)\Big/\ln d_2\right\rceil.$$

In particular, if we choose $c=\rho(\mathcal{E})$, then the total iterations are bounded by $(\lceil\ln(\frac{\epsilon}{\rho(\mathcal{E})+C})/\ln\theta\rceil+1)\times\lceil\ln(\frac{\theta}{2-\theta})/\ln d_2\rceil$.

In the following we establish the upper bound of outer iterations. By the monotonicity of the spectral radius and Assumption 3.1, we have that

$$0\leqslant\overline{\lambda}_l-\rho(A)\leqslant\overline{\lambda}_l-\rho(A_l)+C\tau_l\leqslant\epsilon_l+C\tau_l.$$

So we can get a required approximate solution provided $\epsilon_l+C\tau_l<\epsilon$. Because $\epsilon_l=c\theta^l$, $\tau_l=\theta^l$, we get from Lemma 3.2 that the outer iterations do not exceed $L=\lceil\ln(\frac{\epsilon}{c+C})/\ln\theta\rceil+1$. □

Corollary 3.1. *In Algorithm 3.2, set $\epsilon_k=\rho(\mathcal{E})\theta^k$. Then for any given $\epsilon>0$, one may get a required approximate solution by using Algorithm 3.2 within $(\lceil\ln(\frac{\epsilon}{\rho(\mathcal{E})+C})/\ln\theta\rceil+1)\lceil\ln(\frac{\theta}{2-\theta})/\ln d_2\rceil mn^m$ operations.*

3.2. The rectangular tensor case

For real rectangular tensors, they arise from the strong ellipticity condition problem in solid mechanics [25] and the entanglement problem in quantum physics [7]. In [3], Chang et al. systematically discussed properties of singular values of such rectangular tensors and proposed Algorithm 3.3 to find the largest singular value of a nonnegative rectangular tensor. And the convergence of this algorithm is established under primitivity assumption in [28]. However, many examples show that this algorithm does not work for a general rectangular tensor, such as Example 6.8 and Example 6.9.

In this subsection, our main contribution is to propose an always convergent algorithm for the largest singular values for a general nonnegative rectangular tensor by using inexact techniques.

First, let's recall the algorithm presented in [3,31]. Suppose that $\mathcal{A}$ is a real (p,q)th order $m\times n$ dimensional rectangular tensor and $M=p+q$.

Algorithm 3.3.

Step 1. Given $x^{(0)}\in R^m_{++}$, $y^{(0)}\in R^n_{++}$, let $\xi^{(0)}=\mathcal{A}(x^{(0)})^{p-1}(y^{(0)})^q$, $\eta^{(0)}=\mathcal{A}(x^{(0)})^p(y^{(0)})^{q-1}$ and set $k=0$.

Step 2. Compute

$$x^{(k+1)}=\left(\xi^k\right)^{[\frac{1}{M-1}]}/\left\|\left(\xi^k,\eta^k\right)^{[\frac{1}{M-1}]}\right\|,\qquad y^{(k+1)}=\left(\eta^k\right)^{[\frac{1}{M-1}]}/\left\|\left(\xi^k,\eta^k\right)^{[\frac{1}{M-1}]}\right\|,$$
$$\xi^{(k+1)}=\mathcal{A}\left(x^{(k+1)}\right)^{p-1}\left(y^{(k+1)}\right)^q,\qquad \eta^{(k+1)}=\mathcal{A}\left(x^{(k+1)}\right)^{p}\left(y^{(k+1)}\right)^{q-1}.$$

Let

$$\overline{\lambda}_{k+1}=\max_{i,j}\left\{\frac{(\xi^{(k+1)})_i}{(x^{(k+1)})_i^{M-1}},\frac{(\eta^{(k+1)})_j}{(y^{(k+1)})_j^{M-1}}\right\},$$
$$\underline{\lambda}_{k+1}=\min_{i,j}\left\{\frac{(\xi^{(k+1)})_i}{(x^{(k+1)})_i^{M-1}},\frac{(\eta^{(k+1)})_j}{(y^{(k+1)})_j^{M-1}}\right\}.$$

Step 3. If $\overline{\lambda}_{k+1}=\underline{\lambda}_{k+1}$, stop. Otherwise, replace k by $k+1$ and go to Step 2.

Before giving the rectangular version of power-type algorithm, we list two important properties of the largest singular value $r\rho(\mathcal{A})$, which can be useful in further analysis. One can refer to [28].

Lemma 3.4. *If $\mathcal{A},\mathcal{B}$ are two tensors with the same order and dimension, $0\leqslant\mathcal{A}\leqslant\mathcal{B}$. Then $r\rho(\mathcal{A})\leqslant r\rho(\mathcal{B})$. Furthermore, if $\mathcal{A}$ is irreducible and $\mathcal{A}\neq\mathcal{B}$, then $r\rho(\mathcal{A})<r\rho(\mathcal{B})$.*

Lemma 3.5. *Assume $\{\mathcal{A}_k\}$ is a positive tensor sequence with common order and dimension, and $\mathcal{A}_k\to\mathcal{A}$ as $k\to\infty$. Then $\lim_{k\to\infty}r\rho(\mathcal{A}_k)=r\rho(\mathcal{A})$.*

Now, for a general nonnegative rectangular tensor, we can also present an inexact power-type algorithm based on Algorithm 3.3 and Lemma 3.5.

Let δ be the all-ones (p,q)th order $m\times n$ dimensional rectangular tensor.

Algorithm 3.4.

Step 1. Take a positive sequence $\{\epsilon_k\}$ such that $\sum_{k=1}^{\infty}\epsilon_k<\infty$. Given a $\theta\in(0,1)$, set $\tau_1=\theta$, $\mathcal{A}_1=\mathcal{A}+\tau_1\delta$. Choose $x^{(0)}\in R^m_{++}$, $y^{(0)}\in R^n_{++}$. Let $\xi^{(0)}=\mathcal{A}_1(x^{(0)})^{p-1}(y^{(0)})^q$, $\eta^{(0)}=\mathcal{A}_1(x^{(0)})^p(y^{(0)})^{q-1}$. Let $\xi_1^{(0)}=\xi^{(0)}$, $\eta_1^{(0)}=\eta^{(0)}$. $l=1$.

Step 2. Compute

$$x_l^{(k)}=\left(\xi_l^{(k-1)}\right)^{[\frac{1}{M-1}]}/\left\|\left(\xi_l^{(k-1)},\eta_l^{(k-1)}\right)^{[\frac{1}{M-1}]}\right\|,$$
$$x_l^{(k)}=\left(\eta_l^{(k-1)}\right)^{[\frac{1}{M-1}]}/\left\|\left(\xi_l^{(k-1)},\eta_l^{(k-1)}\right)^{[\frac{1}{M-1}]}\right\|,$$
$$\xi_l^{(k)}=\mathcal{A}_l\left(x_l^{(k)}\right)^{p-1}\left(y_l^{(k)}\right)^q,\qquad \eta_l^{(k)}=\mathcal{A}_l\left(x_l^{(k)}\right)^{p}\left(y_l^{(k)}\right)^{q-1},$$
$$\overline{\lambda}_k^l=\max_{i,j}\left\{\frac{(\xi_l^{(k)})_i}{(x_l^{(k)})_i^{M-1}},\frac{(\eta_l^{(k)})_j}{(y_l^{(k)})_j^{M-1}}\right\},$$

$$\underline{\lambda}_k^l = \min_{i,j}\left\{\frac{(\xi_l^{(k)})_i}{(x_l^{(k)})_i^{M-1}}, \frac{(\eta_l^{(k)})_j}{(y_l^{(k)})_j^{M-1}}\right\},$$

for $k = 1, 2, \ldots$, until $\overline{\lambda}_k^l - \underline{\lambda}_k^l < \epsilon_l$, denote this k as $k(l)$ and

$$x_{l+1}^{(0)} = x_l^{(k(l))}, \qquad y_{l+1}^{(0)} = y_l^{(k(l))}, \qquad \overline{\lambda}^l = \overline{\lambda}_{k(l)}^l, \qquad \underline{\lambda}^l = \underline{\lambda}_{k(l)}^l.$$

Step 3. $l = l+1$, $\tau_l = \theta\tau_l$, $\mathcal{A}_l = \mathcal{A} + \tau_l\delta$. Set $\xi_l^{(0)} = \mathcal{A}_l(x_l^{(0)})^{p-1}(y_l^{(0)})^q$, $\eta_l^{(0)} = \mathcal{A}_l(x_l^{(0)})^p(y_l^{(0)})^{q-1}$. Go to Step 2.

Like the square case, Algorithm 3.4 can be complemented in practice as Lemma 3.6 shows.

Lemma 3.6. *The inner iteration Step* 2 *in the algorithm will terminate in a finite number of steps.*

Similarly, the convergence of Algorithm 3.4 could also be derived.

Theorem 3.4. *The sequences* $\overline{\lambda}_k$ *and* $\underline{\lambda}_k$ *generated by Algorithm* 3.4 *converge to* $r\rho(\mathcal{A})$, *respectively.*

Proof. Because $\mathcal{A}_{l+1} < \mathcal{A}_l$, we have $r\rho(\mathcal{A}_{l+1}) < r\rho(\mathcal{A}_l)$ due to the monotonicity of the spectral radius. So we get the following by the algorithm

$$\overline{\lambda}^l \geqslant r\rho(\mathcal{A}_l) \geqslant r\rho(\mathcal{A}_{l+1}) \geqslant \overline{\lambda}^{l+1} - \epsilon_{l+1},$$

from which one has

$$\overline{\lambda}^{l+1} \leqslant \overline{\lambda}^l + \epsilon_{l+1}.$$

Since both $\{\overline{\lambda}^l\}$ and $\{\epsilon_l\}$ are all nonnegative sequences and $\sum_{k=1}^{\infty}\epsilon_k < \infty$, it follows that $\{\overline{\lambda}^l\}$ is convergent. From Lemma 3.5 we get that

$$\lim_{l\to\infty}\overline{\lambda}^l = \lim_{l\to\infty} r\rho(\mathcal{A}_l) = r\rho(\mathcal{A}).$$

Similarly one has that $\lim_{l\to\infty}\underline{\lambda}^l = r\rho(\mathcal{A})$. □

Remark. Recently, Zhang [32] established the linear convergence of Algorithm 3.3 under reasonable assumptions. So like the square case, we can also give the computing complexity of Algorithm 3.4. We just omit it here.

4. One-step inner iteration power-type algorithm

In this section we present a one-step inner iteration power-type algorithm and establish its convergence. In Algorithms 3.2 and 3.4, it may cost too much time to achieve the required precision in every inner loop. Next, we offer an algorithm with only one-step inner loop.

We first state our algorithm as follows.

Algorithm 4.5.

Step 1. Given a $\theta \in (0,1)$, set $\tau_1 = \theta$, $\mathcal{A}_1 = \mathcal{A} + \tau_1\mathcal{E}$. Set $x^{(0)} = e$, $y^{(0)} = \mathcal{A}_1(x^{(0)})^{m-1}$. Denote $x_1^{(0)} = x^{(0)}$, $y_1^{(0)} = y^{(0)}$. $l = 1$.

Step 2. Compute

$$x_l^{(1)} = \left(y_l^{(0)}\right)^{[\frac{1}{m-1}]} \Big/ \left\|\left(y_l^{(0)}\right)^{[\frac{1}{m-1}]}\right\|, \qquad y_l^{(1)} = \mathcal{A}_l\left(x_l^{(1)}\right)^{m-1},$$

$$\overline{\lambda}_1^l = \max_i \frac{(y_l^{(1)})_i}{(x_l^{(1)})_i^{m-1}}, \qquad \underline{\lambda}_1^l = \min_i \frac{(y_l^{(1)})_i}{(x_l^{(1)})_i^{m-1}}.$$

Step 3. $l = l+1$, $x_l^{(0)} = x_l^{(1)}$, $\tau_l = \theta\tau_l$, $\mathcal{A}_l = \mathcal{A} + \tau_l\mathcal{E}$, $y_l^{(0)} = \mathcal{A}_l(x_l^{(0)})^{m-1}$. Go to Step 2.

Remark. It is easy to see that in Algorithm 3.2 if the inner iteration is performed once without the precision requirement, then the parameter τ_k is decreased, it reduces to Algorithm 4.5.

Theorem 4.5. *If $\mathcal{A}$ is nonnegative, then Algorithm 4.5 produces the value of the spectral radius $\rho(\mathcal{A})$, or generates two convergent sequences $\{\overline{\lambda}_1^l\}$ and $\{\underline{\lambda}_1^l\}$. Furthermore, let $\overline{\lambda} = \lim_{l\to\infty} \overline{\lambda}_1^l$ and $\underline{\lambda} = \lim_{l\to\infty} \underline{\lambda}_1^l$. Then $\overline{\lambda}$ and $\underline{\lambda}$ are an upper bound and a lower bound of $\rho(\mathcal{A})$, respectively. If $\overline{\lambda} = \underline{\lambda}$, then $\overline{\lambda} = \underline{\lambda} = \rho(\mathcal{A})$.*

Proof. Similarly as in the proof of Theorem 3.3, we denote

$$\overline{\lambda}_0^{l+1} = \max_i \frac{[\mathcal{A}_{l+1}(x_{l+1}^{(0)})^{m-1}]_i}{(x_{l+1}^{(0)})_i^{m-1}}, \qquad \underline{\lambda}_0^{l+1} = \min_i \frac{[\mathcal{A}_{l+1}(x_{l+1}^{(0)})^{m-1}]_i}{(x_{l+1}^{(0)})_i^{m-1}}.$$

Then we have

$$\overline{\lambda}_0^{l+1} \leqslant \overline{\lambda}_1^l, \qquad \underline{\lambda}_0^{l+1} \leqslant \underline{\lambda}_1^l,$$
$$\underline{\lambda}_0^{l+1} \geqslant \underline{\lambda}_1^l - \theta^l(1-\theta)\rho(\mathcal{E}).$$

Thanks to Lemma 3.1, $\{\overline{\lambda}_k^{l+1}\}$ is monotonically decreasing, and $\{\underline{\lambda}_k^{l+1}\}$ is monotonically increasing for tensor $\mathcal{A}_{l+1}$. So

$$\overline{\lambda}_1^l \geqslant \overline{\lambda}_0^{l+1} \geqslant \overline{\lambda}_1^{l+1} \geqslant \rho(\mathcal{A}_{l+1}) \geqslant \underline{\lambda}_1^{l+1} \geqslant \underline{\lambda}_0^{l+1} \geqslant \underline{\lambda}_1^l - \theta^l(1-\theta)\rho(\mathcal{E}).$$

It follows that $\{\overline{\lambda}_1^l\}$ is nonincreasing and $\{\underline{\lambda}_1^l\}$ is weak increasing. Since $\sum_{l=1}^{\infty} \theta^l(1-\theta)\rho(\mathcal{E}) = \theta\rho(\mathcal{E}) < \infty$, $\{\overline{\lambda}_1^l\}$ and $\{\underline{\lambda}_1^l\}$ converge respectively. What's more, if $\overline{\lambda} = \underline{\lambda}$, then $\overline{\lambda} = \underline{\lambda} = \rho(\mathcal{A})$ holds by Theorem 3.1. □

Moreover, if $\mathcal{A}$ is essential positive, we show that the two sequences $\{\overline{\lambda}_1^l\}$ and $\{\underline{\lambda}_1^l\}$ converge to $\rho(\mathcal{A})$, respectively.

Theorem 4.6. *If $\mathcal{A}$ is essential positive, then Algorithm 4.5 produces the value of the spectral radius $\rho(\mathcal{A})$, or generates two sequences $\{\overline{\lambda}_1^l\}$ and $\{\underline{\lambda}_1^l\}$ which converge to $\rho(\mathcal{A})$, respectively.*

Proof. Similarly as in the proof of Theorem 3.3, we denote

$$\overline{\lambda}_0^{l+1} = \max_i \frac{[\mathcal{A}_{l+1}(x_{l+1}^{(0)})^{m-1}]_i}{(x_{l+1}^{(0)})_i^{m-1}}, \qquad \underline{\lambda}_0^{l+1} = \min_i \frac{[\mathcal{A}_{l+1}(x_{l+1}^{(0)})^{m-1}]_i}{(x_{l+1}^{(0)})_i^{m-1}}.$$

Then we have

$$\overline{\lambda}_0^{l+1} \leqslant \overline{\lambda}_1^l, \qquad \underline{\lambda}_0^{l+1} \leqslant \underline{\lambda}_1^l,$$
$$\overline{\lambda}_0^{l+1} \geqslant \overline{\lambda}_1^l - \theta^l(1-\theta)\rho(\mathcal{E}),$$
$$\underline{\lambda}_0^{l+1} \geqslant \underline{\lambda}_1^l - \theta^l(1-\theta)\rho(\mathcal{E}).$$

Thanks to Lemma 3.2, $\{\overline{\lambda}_k^{l+1}\}$ and $\{\underline{\lambda}_k^{l+1}\}$ converge linearly to $\rho(\mathcal{A}_{l+1})$ for tensor $\mathcal{A}_{l+1}$. So

$$0 \leqslant \overline{\lambda}_1^{l+1} - \rho(\mathcal{A}_{l+1}) \leqslant \overline{\lambda}_1^{l+1} - \underline{\lambda}_1^{l+1} \leqslant \alpha_{l+1}\left(\overline{\lambda}_0^{l+1} - \underline{\lambda}_0^{l+1}\right), \tag{4.1}$$
$$0 \leqslant \rho(\mathcal{A}_{l+1}) - \underline{\lambda}_1^{l+1} \leqslant \overline{\lambda}_1^{l+1} - \underline{\lambda}_1^{l+1} \leqslant \alpha_{l+1}\left(\overline{\lambda}_0^{l+1} - \underline{\lambda}_0^{l+1}\right). \tag{4.2}$$

Denote

$$d = \max\left\{1 - \frac{\beta}{\overline{R}}, 1 - \frac{\beta+\theta}{\overline{R} + n^{m-1}\theta}\right\}.$$

By Proposition 2.1, we have $\beta = \min_{i,j\in\{1,2,\ldots,n\}} a_{ij\cdots j} > 0$ and $0 < d < 1$, so

$$\begin{aligned}
0 \leqslant \bar{\lambda}_1^{l+1} - \underline{\lambda}_1^{l+1} &\leqslant \alpha_{l+1}\left(\bar{\lambda}_0^{l+1} - \underline{\lambda}_0^{l+1}\right) \\
&\leqslant d\left(\bar{\lambda}_0^{l+1} - \underline{\lambda}_0^{l+1}\right) \\
&\leqslant d\left(\bar{\lambda}_1^{l} - \underline{\lambda}_1^{l}\right) + d\theta^l(1-\theta)\rho(\mathcal{E}) \\
&\cdots \\
&\leqslant d^l\left(\bar{\lambda}_1^{1} - \underline{\lambda}_1^{1}\right) + \left(d\theta^l + d^2\theta^{l-1} + \cdots + d^l\theta\right)(1-\theta)\rho(\mathcal{E}) \\
&= \begin{cases} d^l(\bar{\lambda}_1^1 - \underline{\lambda}_1^1) + l\theta^{l+1}(1-\theta)\rho(\mathcal{E}) & \text{if } d = \theta, \\ d^l(\bar{\lambda}_1^1 - \underline{\lambda}_1^1) + \frac{\theta^l - d^l}{\theta - d} d\theta(1-\theta)\rho(\mathcal{E}) & \text{if } d \neq \theta. \end{cases}
\end{aligned} \tag{4.3}$$

From (4.3), we could derive that when l tends to ∞, $\bar{\lambda}_1^{l+1} - \underline{\lambda}_1^{l+1}$ tends to zero. By Theorems 4.5 and 3.1, we have

$$\lim_{l\to\infty} \bar{\lambda}_1^l = \lim_{l\to\infty} \underline{\lambda}_1^l = \rho(\mathcal{A}). \qquad \square$$

As in the previous section, we may give the one-step inner iteration power-type algorithm for the nonnegative rectangular tensor. Since it is similar to the square situation, we omit it here. Our numerical experiments in Section 6 show that Algorithm 4.5 performs well for many nonnegative square tensors which do not satisfy convergence assumption in Theorem 4.6. So we guess that this algorithm converges under weaker conditions.

5. Singular values and reformulation

In this section, we consider the singular values for the rectangular tensors and the general tensors. By using the symmetric embedding technique, we establish the connection between singular values of the rectangular/general tensors and eigenvalues of related symmetric embedding tensors. Then we can compute the largest singular value of nonnegative rectangular/general tensors by using the algorithms for the largest eigenvalue of nonnegative square tensors.

5.1. Singular values of a rectangular tensor

In this subsection, we present a way to convert the singular value problem of a rectangular tensor to an eigenvalue problem of a related square tensor. In [4], Chang and Zhang introduce the lifting operator $\mathcal{T}_{\mathcal{A}}$ of a rectangular tensor $\mathcal{A}$, and they establish the relation between the singular values of original tensor and the eigenvalues of lifting square tensor. However, there is a minor mistake in the definition of $\mathcal{T}_{\mathcal{A}}$ which is weakly symmetric. We here define another lifting operator $\mathcal{C}_{\mathcal{A}}$ of a rectangular tensor $\mathcal{A}$ to correct the error in [4].

For a (p,q) order (m,n) dimensional tensor $\mathcal{A}$, the lifting $\mathcal{C}_{\mathcal{A}} = (c_{k_1,k_2,\ldots,k_M})$ as an $M = p+q$ order $N = m+n$ dimensional tensor is defined by:

$$c_{k_1,k_2,\ldots,k_M} = \begin{cases} a_{k_1,\ldots,k_p,k_{p+1}-m,\ldots,k_M-m} & \text{if } \begin{cases} 1 \leqslant k_1,\ldots,k_p \leqslant m, \\ m+1 \leqslant k_{p+1},\ldots,k_M \leqslant N, \end{cases} \\ a_{k_{q+1},\ldots,k_M,k_1-m,\ldots,k_q-m} & \text{if } \begin{cases} m+1 \leqslant k_1,\ldots,k_q \leqslant N, \\ 1 \leqslant k_{q+1},\ldots,k_M \leqslant m, \end{cases} \\ 0 & \text{otherwise.} \end{cases}$$

When $p = q = 1$, $\mathcal{A}$ is a matrix. From the definition, we can see that $\mathcal{C}_A = [0\ A;\ A^T\ 0]$.

Let

$$z = \begin{pmatrix} x \\ y \end{pmatrix} \in R^N = R^m \times R^n.$$

Then by definition,

$$\mathcal{C}_{\mathcal{A}} z^{M-1} = \begin{pmatrix} \mathcal{A}x^{p-1}y^{q} \\ \mathcal{A}x^{p}y^{q-1} \end{pmatrix}.$$

Note that $\mathcal{C}_{\mathcal{A}}$ may not be weakly symmetric (see [4]) since

$$\mathcal{C}_{\mathcal{A}} z^{M} = 2\mathcal{A}x^{p}y^{q}.$$

Theorem 5.7. *For a (p, q) order (m, n) dimensional tensor $\mathcal{A}$, if σ is a singular value of $\mathcal{A}$ with the singular vectors x, y, then $z = (x^T, y^T)$ is the eigenvector of $\mathcal{C}_{\mathcal{A}}$ associated with eigenvalue σ. Moreover, if $z = (x^T, y^T)$ is the eigenvector of $\mathcal{C}_{\mathcal{A}}$ associated with eigenvalue $\lambda \neq 0$, then λ is $\mathcal{A}$'s singular value with the singular vectors x, y.*

Proof. If σ is a singular value of $\mathcal{A}$ with the singular vectors x, y, then $z = (x^T, y^T) \neq 0$ and by definition, we have

$$\mathcal{C}_{\mathcal{A}} z^{M-1} = \begin{pmatrix} \mathcal{A}x^{p-1}y^{q} \\ \mathcal{A}x^{p}y^{q-1} \end{pmatrix} = \sigma \begin{pmatrix} x^{[M-1]} \\ y^{[M-1]} \end{pmatrix} = \sigma z^{[M-1]}.$$

So the first part of the theorem holds.

On the other hand, if $z = (x^T, y^T)$ is the eigenvector of $\mathcal{C}_{\mathcal{A}}$ associated with eigenvalue $\lambda \neq 0$, we have

$$\begin{pmatrix} \mathcal{A}x^{p-1}y^{q} \\ \mathcal{A}x^{p}y^{q-1} \end{pmatrix} = \mathcal{C}_{\mathcal{A}} z^{M-1} = \lambda z^{[M-1]} = \lambda \begin{pmatrix} x^{[M-1]} \\ y^{[M-1]} \end{pmatrix}. \tag{5.1}$$

Since $\lambda \neq 0$, then neither x nor y can be zero. Suppose $x = 0$. Then y must be zero by (5.1), which conflicts with that z is an eigenvector. So the second part of the theorem holds. □

Thus based on this result, one can calculate the singular values of a rectangular tensor $\mathcal{A}$ by calculating the eigenvalues of its lifting square tensor $\mathcal{C}_{\mathcal{A}}$. In particular, if $\mathcal{A}$ is nonnegative, then the spectral radius of $\mathcal{C}_{\mathcal{A}}$ is the largest singular value of $\mathcal{A}$.

5.2. Singular values of a general tensor

For a general tensor $\mathcal{A} \in R^{n_1 \times n_2 \times \cdots \times n_d} (d \geqslant 2)$, $n_i \neq n_j$ $(i \neq j)$, $i, j \in \{1, 2, \ldots, d\}$, Lim [15] defined its singular values and singular vectors.

Definition 5.6. The scalar $\sigma \in R$ is a singular value of a general tensor $\mathcal{A} \in R^{n_1 \times \cdots \times n_d}$ if there are vectors $u_k \in R^{n_k} \setminus \{0\}$ such that

$$\begin{cases} \mathcal{A}u_2 u_3 \cdots u_d = \sigma u_1^{[d-1]}, \\ \mathcal{A}u_1 u_3 \cdots u_d = \sigma u_2^{[d-1]}, \\ \quad \cdots \\ \mathcal{A}u_1 u_2 \cdots u_{d-1} = \sigma u_d^{[d-1]}. \end{cases} \tag{5.2}$$

The vector u_k is the mode-k singular vector associated with σ.

Remark. There is a little difference between this definition and the singular values of a matrix when $d = 2$. The singular values in matrix are all positive while this definition does not have this restriction.

As known, there are some connections between the singular values of a matrix A and the eigenvalues of its symmetric embedding $\mathbf{sym}(A) = ([0\ A; A^T\ 0])$. In particular, the set of all the positive eigenvalues of $\mathbf{sym}(A)$ is the same as the set of all the singular values of A. The top and bottom

halves of $\mathbf{sym}(A)$'s eigenvectors are singular vectors for A. In the following, we draw into the symmetric embedding $\mathbf{sym}(\mathcal{A})$ for a general tensor $\mathcal{A}$. And we give the relations between the singular values of $\mathcal{A}$ and the eigenvalues of $\mathbf{sym}(\mathcal{A})$ in the following.

In [23], Ragnarsson and Van Loan put forward the concept of block tensors and symmetric embeddings, and they derive the relations between the *singular values* of a general tensor and the *eigenvalues* of its symmetric embedding. But the singular values and eigenvalues defined in [23] are called *Z-singular values* and *Z-eigenvalues*, which require that the vectors are unit with Euclidean norm, and the powers of the eigenvectors at the right-hand side of Eqs. (5.2) are one. This definition was first given by Lim and Qi in [15,21]. However, the singular values and eigenvalues in our definition are called *H-singular values* and *H-eigenvalues* which require the equations are homogeneous. Our contribution here is to give the relations between the H-singular values of $\mathcal{A}$ and the H-eigenvalues of $\mathbf{sym}(\mathcal{A})$. In this subsection, the singular value and eigenvalue mean H-singular value and H-eigenvalue.

First we draw into the tensor transposition and symmetric embedding. For details, the readers are referred to [23].

Definition 5.7. If $\mathcal{A} \in R^{n_1\times\cdots\times n_d}$ and $\boldsymbol{p} = [p_1, \ldots, p_d]$ is a permutation of $[1, \ldots, d]$, then $\mathcal{A}^{\langle p\rangle} \in R^{n_{p_1}\times\cdots\times n_{p_d}}$ denotes the $\boldsymbol{p}$-transpose of $\mathcal{A}$ defined by

$$\mathcal{A}^{\langle p\rangle}(j_{p_1}, \ldots, j_{p_d}) = \mathcal{A}(j_1, \ldots, j_d),$$

where $1 \leqslant j_k \leqslant n_k$ for $k = 1, \ldots, d$.

Definition 5.8. If $\mathcal{A} \in R^{n_1\times\cdots\times n_d}$, then its symmetric embedding

$$\mathcal{C} = sym(\mathcal{A}) \in R^{N\times\cdots\times N}, \quad N = n_1 + \cdots + n_d,$$

is a block tensor defined by the partitioning $[1, \ldots, N] = [\boldsymbol{r}_1|\cdots|\boldsymbol{r}_d]$ where

$$\boldsymbol{r}_k = \big[(1 + n_1 + \cdots + n_{k-1}), \ldots, (n_1 + \cdots + n_k)\big], \quad k = 1, \ldots, d.$$

The $\boldsymbol{i}$th block of $\mathcal{C}$ is given by

$$\mathcal{C}_{\boldsymbol{i}} = \begin{cases} A^{\langle \boldsymbol{i}\rangle}, & \text{if } \boldsymbol{i} \text{ is a permutation of } [1, \ldots, d], \\ 0, & \text{otherwise,} \end{cases}$$

for all $\boldsymbol{i}$ that satisfy $\mathbf{1} \leqslant \boldsymbol{i} \leqslant \boldsymbol{d}$.

Note that $\mathcal{C}_i$ is of form $n_{i_1} \times n_{i_2} \cdots \times n_{i_d}$ and $sym(\mathcal{A})$ is symmetric by Lemma 2.1 in [23]. However, when $d > 2$, $sym(\mathcal{A})$ could be reducible.

Theorem 5.8. *For a general tensor $\mathcal{A} \in R^{n_1\times\cdots\times n_d}$ $(d > 2)$, $\mathcal{C} = sym(\mathcal{A})$ is reducible.*

Proof. When $d > 2$, we can choose $I = r_1 \cup r_2$. Then $\forall i_2, \ldots, i_d \in I^C$, there must exist $m, n \in \{2, 3, \ldots, d\}$, $k \in \{3, 4, \ldots, d\}$ such that $i_m, i_n \in r_k$ by the principle of drawer. So by the definition of $sym(\mathcal{A})$, $\forall i_1 \in I$, $i_2, \ldots, i_d \in I^C$, $C_{i_1 i_2 \cdots i_d} = 0$ which means $\mathcal{C} = sym(\mathcal{A})$ is reducible. □

Note that when $d = 2$, it is obvious that the symmetric embedding $\mathbf{sym}(A) = ([0\ \ A; A^T\ \ 0])$ is irreducible if A is positive. However, $sym(\mathcal{A})$ is not primitive even if $\mathcal{A}$ is positive.

Example 5.4. Suppose $A = [1\ 1] \in R^{1\times 2}$, then

$$sym(A) = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

which is irreducible but not primitive.

Theorem 5.9. *For a general tensor $\mathcal{A} \in R^{n_1 \times \cdots \times n_d} (d \geqslant 2)$, if σ is a singular value of $\mathcal{A}$ with the mode-k singular vector u_k $(k = 1, \ldots, d)$, then $x = (u_1^T, \ldots, u_d^T)$ is the eigenvector of $sym(\mathcal{A})$ associated with eigenvalue $(d-1)!\sigma$. Moreover, if $x = (u_1^T, \ldots, u_d^T)$ is the eigenvector of $sym(\mathcal{A})$ associated with eigenvalue $\lambda \neq 0$, then $\frac{\lambda}{(d-1)!}$ is $\mathcal{A}$'s singular value with the mode-k singular vector u_k $(k = 1, \ldots, d)$.*

Proof. From the definition of $sym(\mathcal{A})$, for any $x = (u_1^T, \ldots, u_d^T)$ we have

$$\begin{aligned} sym(\mathcal{A})x^d &= \sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{N}} \mathcal{C}(\boldsymbol{i}) x(i_1) \cdots x(i_d) \\ &= \sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{n}} \mathcal{A}^{(\boldsymbol{i})} u_1(i_1) \cdots u_d(i_d) \\ &= d!(\mathcal{A}u_1 u_2 \cdots u_d). \end{aligned}$$

Since $sym(\mathcal{A})$ is symmetric, it follows that

$$sym(\mathcal{A})x^{d-1} = (d-1)! \begin{pmatrix} \mathcal{A}u_2 u_3 \cdots u_d \\ \mathcal{A}u_1 u_3 \cdots u_d \\ \cdots \\ \mathcal{A}u_1 u_2 \cdots u_{d-1} \end{pmatrix}. \tag{5.3}$$

If σ is a singular value of $\mathcal{A}$ with the mode-k singular vector u_k $(k = 1, \ldots, d)$, then $x = (u_1^T, \ldots, u_d^T) \neq 0$ and it satisfies

$$\begin{aligned} sym(\mathcal{A})x^{d-1} &= (d-1)! \begin{pmatrix} \mathcal{A}u_2 u_3 \cdots u_d \\ \mathcal{A}u_1 u_3 \cdots u_d \\ \cdots \\ \mathcal{A}u_1 u_2 \cdots u_{d-1} \end{pmatrix} \\ &= (d-1)!\sigma \begin{pmatrix} u_1^{[d-1]} \\ u_2^{[d-1]} \\ \cdots \\ u_d^{[d-1]} \end{pmatrix} \\ &= (d-1)!\sigma x^{[d-1]}. \end{aligned}$$

So the first part of the theorem holds.

On the other hand, if $x = (u_1^T, \ldots, u_d^T)$ is the eigenvector of $sym(\mathcal{A})$ associated with eigenvalue $\lambda \neq 0$, then by (5.3), we can derive

$$\begin{pmatrix} \mathcal{A}u_2 u_3 \cdots u_d \\ \mathcal{A}u_1 u_3 \cdots u_d \\ \cdots \\ \mathcal{A}u_1 u_2 \cdots u_{d-1} \end{pmatrix} = \frac{\lambda}{(d-1)!} x^{[d-1]} = \frac{\lambda}{(d-1)!} \begin{pmatrix} u_1^{[d-1]} \\ u_2^{[d-1]} \\ \cdots \\ u_d^{[d-1]} \end{pmatrix}. \tag{5.4}$$

We will complete our proof as long as we show that for any $k \in \{1, \ldots, d\}$, $u_k \neq 0$. Suppose that there is a $k_0 \in \{1, \ldots, d\}$ such that $u_{k_0} = 0$. Then by (5.4) one can obtain for any $k \in \{1, \ldots, d\}$, $u_k = 0$ since $\lambda \neq 0$. It conflicts with that x is an eigenvector. So the second part of the theorem holds. □

Corollary 5.2. *Assume that $\mathcal{A} \in R^{n_1 \times \cdots \times n_d} (d \geqslant 2)$ is nonnegative and $\rho(sym(\mathcal{A}))$ is the spectral radius of $sym(\mathcal{A})$. Then $\frac{\rho(sym(\mathcal{A}))}{(d-1)!}$ is the largest singular value of $\mathcal{A}$.*

Remark. As far as we know, few methods were proposed to find the singular values for a general tensor. By Theorem 5.9, we get the relation between singular values of a general tensor and eigenvalues

Table 1
Numerical results of Algorithm 3.2 for Examples 6.5–6.7.

Example	ite_{out}	λ	*vol*	x
Example 6.5	10	0.3627	4.1234e−05	$(0.5598, 0.8286, 0.0000)^T$
Example 6.6	9	0.9875	2.1977e−06	$(0.0000, 0.0000, 0.0000, 0.0000, 1.0000)^T$
Example 6.7	10	0.4907	9.8460e−06	$(0.0000, 0.8163, 0.0000, 0.0001, 0.0000, 0.0000, 0.0000, 0.0000, 0.5777, 0.0000)^T$

of its symmetric embedding, which suggests a new way to find singular values of a general tensor, especially for the largest singular value. However, since $sym(\mathcal{A})$ is not primitive by Theorem 5.8, the convergence condition of Collatz's power-type algorithm for computing the spectral radius of a square tensor does not hold, which means that original power-type Algorithm 3.1 may not apply. Fortunately, our Algorithms 3.2 and 4.5 would be efficient in this case. By Theorems 3.2 and 4.6, they converge to the largest singular value for a general nonnegative tensor.

6. Numerical results

In this section, we present some preliminary results to test our methods proposed in Section 3 and Section 4 for finding the spectral radius (largest singular value) of the nonnegative tensors. Here, we mainly focus on the examples which cannot be solved by the NQZ method proposed in [19]. All codes were written by Matlab R2012b and all the numerical experiments were done on a laptop with Intel Core i5-2430M CPU 2.4 GHz and 1.58 GB memory.

In the implementation of Algorithms 3.2 and 3.4, we set the parameters $\tau_k = 10^{-k}$ and $\epsilon_k = 0.1 \times 0.3^{k-1}$. And the initial points are all-ones vectors. We use the following criterion:

$$\frac{\|\lambda^{k-1} - \lambda^k\|}{\lambda^k} \leqslant 10^{-4},$$

where λ^k denotes the largest eigenvalue obtained in the kth outer iteration.

First, we give three examples which cannot be solved by the NQZ method (Algorithm 3.1) within 3000 iterations.

Example 6.5. The 3-order 3-dimensional tensor $\mathcal{A}$ is given by $a(2,1,1) = 0.7943$, $a(1,2,2) = 0.1656$, $a(2,1,3) = 0.6542$, and zero elsewhere.

Example 6.6. The 3-order 5-dimensional tensor $\mathcal{B}$ is given by $b(5,3,1) = 0.1660$, $b(2,2,2) = 0.5583$, $b(1,1,3) = 0.1668$, $b(5,5,4) = 0.7105$, $b(5,5,5) = 0.9875$, and zero elsewhere.

Example 6.7. The 3-order 10-dimensional tensor $\mathcal{C}$ is given by $c(6,8,1) = 0.9065$, $c(9,2,2) = 0.2458$, $c(4,4,3) = 0.8551$, $c(8,7,4) = 0.5242$, $c(9,2,5) = 0.4325$, $c(2,2,6) = 0.4034$, $c(4,4,7) = 0.7353$, $c(4,4,8) = 0.0458$, $c(2,9,9) = 0.9797$, $c(4,8,10) = 0.8819$, and zero elsewhere.

It is worth mentioning that all the three examples are generated randomly with a certain level of sparsity. The elements are randomly distributed in $(0,1)$. From the viewpoint of numerical experiment, we find that the NQZ method probably does not work when the sparsity of a tensor is near to $1/n^{m-1}$, where m and n denote the order and dimension of a tensor, respectively. The numerical results are listed in Table 1 and Fig. 2 shows the courses of $\overline{\lambda}$ and $\underline{\lambda}$ with the outer iteration for Examples 6.5–6.7. In the table, ite_{out} denotes the outer iteration number of Algorithm 3.2. λ and x denote the largest eigenvalue and corresponding eigenvector at the final iteration, respectively. And *vol* means the l^∞ norm of the violation of the solution, i.e. $vol = \|Ax^{m-1} - \lambda x^{[m-1]}\|_\infty$.

Second, as an application, we use our algorithms to find the largest H-eigenvalue for a class of m-uniform hypergraph which can be divided into two connective parts. For simplicity, each connective part is a chain with one point at the joint. For example, a 3-uniform hypergraph $V = \{1, 2, \ldots, 8\}$ and

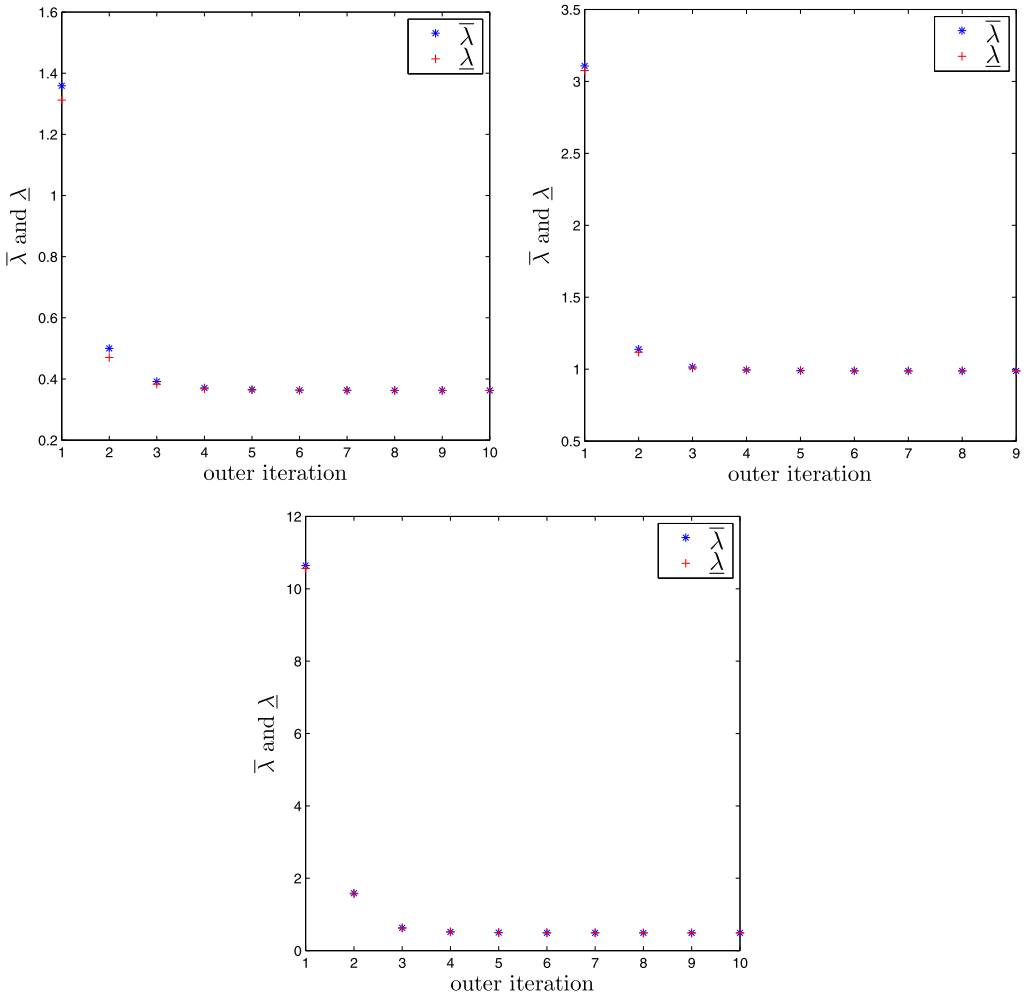

Fig. 2. The courses of $\overline{\lambda}$ and $\underline{\lambda}$ for Examples 6.5–6.7.

$E = \{(1,2,3),(4,5,6),(6,7,8)\}$. The adjacency tensor $\mathcal{A}_{\mathcal{H}}$ of an m-uniform hypergraph $\mathcal{H} = (V, E)$ is an mth order n-dimensional symmetric tensor, with $\mathcal{A}_{\mathcal{H}} = (a_{i_1 i_2 \cdots i_m})$, where $a_{i_1 i_2 \cdots i_m} = \frac{1}{(m-1)!}$ if $(i_1, i_2, \ldots, i_m) \in E$, and 0 otherwise [6].

For this kind of hypergraph, we use Algorithms 3.1 and 3.2 to find the largest H-eigenvalue of its adjacency tensor. The results are listed in Table 2. The parameters $(m, n1, n2)$ of the tested cases are clear from the table. In the table, $cpu(s)$ denotes the time cost when the algorithm stops. If the algorithm could not stop within 3000 steps, we put a '–' in the corresponding cross. '*' means that the problem setting is out of the memory of our PC.

Third, by Theorems 5.7 and 5.9, we use Algorithms 3.1 and 3.2 to find the largest singular values of two rectangular tensors Examples 6.8–6.9 and two general tensors Examples 6.10–6.11. Note that for the rectangular tensors, the CQZ method (Algorithm 3.3) could not stop within 3000 steps. The results are listed in Table 3. In the table, σ means the largest singular value at the final iteration.

Example 6.8. The rectangular tensor $\mathcal{D} \in R^{2\times 2\times 3}$ is given by $d(1,1,1) = 0.5815$, $d(2,2,2) = 0.5394$, $d(1,2,3) = 0.8182$, and zero elsewhere.

Table 2
Numerical results for hypergraph with two connective parts.

m	$n1$	$n2$	Algorithm 3.1			Algorithm 3.2		
			λ	vol	$cpu(s)$	λ	vol	$cpu(s)$
3	3	3	1.0000	0.0000	0.0026	1.0000	6.0000e−09	0.0022
3	3	5	–	–	–	1.2598	2.2631e−05	0.0895
3	3	11	–	–	–	1.4808	2.7000e−06	0.0683
3	3	21	–	–	–	1.5511	4.7873e−07	0.0754
3	11	11	1.4808	1.2934e−08	0.0696	1.4808	3.6441e−07	0.0088
3	11	21	–	–	–	1.5511	7.5766e−07	0.5441
4	4	4	1.0000	6.9389e−18	0.0025	1.0000	2.2627e−07	0.0022
4	4	10	–	–	–	1.2720	3.2410e−07	0.1337
4	4	25	–	–	–	1.3792	1.2095e−07	0.1448
4	10	10	1.2720	2.1550e−09	0.0545	1.2720	9.7333e−08	0.0127
4	10	25	–	–	–	1.3792	1.2649e−07	0.6414
5	5	5	1.0000	1.7347e−18	0.0028	1.0000	1.0000e−07	0.0027
5	5	9	–	–	–	1.1487	9.6305e−08	0.3268
5	5	13	–	–	–	1.2123	1.8314e−08	0.3841
5	5	17	–	–	–	1.2457	3.0977e−08	0.6878
5	5	25	*	*	*	*	*	*

Table 3
Numerical results for Examples 6.8–6.11.

Example	Algorithm 3.1			Algorithm 3.2		
	σ	vol	$cpu(s)$	σ	vol	$cpu(s)$
Example 6.8	–	–	–	0.5815	1.6169e−06	0.3148
Example 6.9	–	–	–	0.9006	7.1596e−11	0.1641
Example 6.10	–	–	–	1.1838	3.1149e−05	0.1178
Example 6.11	–	–	–	0.7961	3.5667e−06	0.1238

Example 6.9. The rectangular tensor $\mathcal{E} \in R^{3\times3\times4\times4}$ is given by $e(1,1,1,2) = 0.8875$, $e(1,1,4,4) = 0.9005$, $e(1,2,1,2) = 0.4480$, $e(3,1,2,3) = 0.2689$, $e(3,3,2,2) = 0.5538$, and zero elsewhere.

Example 6.10. The general tensor $\mathcal{F} \in R^{2\times3\times4}$ is given by $f(2,3,1) = 0.4666$, $f(2,1,2) = 0.1490$, $f(2,3,3) = 0.9786$, $f(1,2,4) = 0.9894$, and zero elsewhere.

Example 6.11. The general tensor $\mathcal{G} \in R^{3\times4\times5\times6}$ is given by $g(1,2,1,3) = 0.1159$, $g(1,2,2,4) = 0.3596$, $g(2,1,2,6) = 0.3061$, $g(2,2,4,2) = 0.3022$, $g(2,3,5,1) = 0.5614$, $g(3,4,3,5) = 0.7961$, and zero elsewhere.

Note that the largest singular values of $\mathcal{D}$ and $\mathcal{E}$ are equal to the largest eigenvalues of $\mathcal{C}_{\mathcal{D}}$ and $\mathcal{C}_{\mathcal{E}}$ in Examples 6.8–6.9, while the largest singular value of $\mathcal{F}$ is half of the largest eigenvalue of $sym(\mathcal{F})$ in Example 6.9 and the largest singular value of $\mathcal{G}$ is one-sixth of the largest eigenvalue of $sym(\mathcal{G})$ in Example 6.10.

Moreover, by reformulation, we test Algorithm 3.2 on some random 3-order nonnegative tensors. All elements are randomly distributed in $(0,1)$. The numerical results are reported in Table 4. The parameters $(d1,d2,d3)$ denote the dimensions of the tested cases. For each case, we simulate 100 times to obtain the average of the largest singular value, the violation and the CPU time.

Finally, two comparisons between our algorithms are presented. One is the comparison between Algorithm 3.2 and Algorithm 4.5 for Example 6.12. The other one is between Algorithm 3.2 and Algorithm 3.4 for Example 6.8. The numerical results are listed in Tables 5 and 6. In Table 5, ite_{total} means the total iterations when the algorithm stops.

Example 6.12. The 3-order 3-dimensional tensor $\mathcal{H}$ is given by $h(1,1,1) = 0.3785$, $h(2,2,2) = 0.9328$, $h(3,3,3) = 0.4293$, $h(1,2,1) = 0.3942$, and zero elsewhere.

Table 4
Numerical results for random 3-order tensors.

$d1$	$d2$	$d3$	Algorithm 3.2		
			σ	vol	$cpu(s)$
5	10	15	41.4081	3.0414e−05	0.0149
5	10	20	50.1316	2.2169e−05	0.0158
5	10	30	65.6525	4.6225e−06	0.0163
5	10	50	92.2702	6.1572e−06	0.0207
5	10	100	146.5652	1.0052e−05	0.0368
10	20	30	165.2803	5.9362e−05	0.0230
10	20	50	232.0583	7.7809e−06	0.0266
10	20	100	368.3402	1.1958e−05	0.0410
20	40	60	660.3639	1.1794e−04	0.0424
20	40	80	800.0576	1.0049e−04	0.0517
20	40	100	928.2820	1.5366e−05	0.0638
40	50	60	1.2163e+03	1.4998e−04	0.0522
40	50	80	1.4736e+03	1.6877e−04	0.0706
40	50	100	1.7099e+03	1.8670e−04	0.0967
50	60	80	1.9309e+03	1.8949e−04	0.0881
50	60	100	2.2405e+03	2.0799e−04	0.1211
60	80	100	3.0652e+03	2.3910e−04	0.1723

Table 5
The comparison of Algorithm 3.2 and Algorithm 4.5 for Example 6.12.

	Algorithm 3.2	Algorithm 4.5
ite_{total}	202	482
$cpu(s)$	0.008569	0.040912
λ	0.9328	0.9328
x	$(0.5796, 0.8149, 0.0000)^T$	$(0.5796, 0.8149, 0.0000)^T$

Table 6
The comparison of Algorithm 3.2 and Algorithm 3.4 for Example 6.8.

	Algorithm 3.2 for $\mathcal{C}_{\mathcal{E}}$	Algorithm 3.4 for $\mathcal{E}$
ite_{out}	11	11
$cpu(s)$	0.043224	0.103781
σ	0.5815	0.5815
x	$(0.7071, 0.0000)^T$	$(0.7071, 0.0000)^T$
y	$(0.7071, 0.0000, 0.0047)^T$	$(0.7071, 0.0000, 0.0047)^T$

From Tables 1–6, we have the following observations:

- From Table 1, we see that the inexact power-type algorithm can solve the case efficiently while the NQZ method doesn't work. From Table 2, we see that for a class of hypergraph, our method can also perform well while the NQZ method works only for the case $n1 = n2$.
- From Table 3, we see that by reformulation, the inexact power-type algorithm works well for the corresponding square tensors while the NQZ method cannot terminate within 3000 steps. And Table 4 shows that our method could also perform well for random nonnegative general tensors. We see that the vol is of magnitude 10^{-4} when the scale of the problem becomes large. That is because the tolerance is 10^{-4}.
- From Table 5, we see that the one-step inner iteration power-type algorithm costs more time than the inexact power-type algorithm for finding the largest eigenvalue. In practice, the one-step inner iteration power-type algorithm does not perform well in some cases.
- From Table 6, we see that for rectangular tensors, it may be more efficient to find the largest singular value by reformulation technique than to solve the problem directly.

Acknowledgements

The authors are very grateful to the editor and the two referees for their valuable suggestions and constructive comments, which have considerably improved the presentation of the paper. And we are grateful to Ms. Wang Wenjuan for her helpful discussion in Section 5.

References

[1] K.C. Chang, K. Pearson, T. Zhang, Perron–Frobenius Theorem for nonnegative tensors, Commun. Math. Sci. 6 (2008) 507–520.
[2] K.C. Chang, K. Pearson, T. Zhang, Primitivity, the convergence of the NQZ method, and the largest eigenvalue for nonnegative tensors, SIAM J. Matrix Anal. Appl. 32 (2011) 806–819.
[3] K.C. Chang, L. Qi, G. Zhou, Singular values of a real rectangular tensor, J. Math. Anal. Appl. 370 (2010) 284–294.
[4] K.C. Chang, T. Zhang, Multiplicity of singular values for tensors, Commun. Math. Sci. 7 (2009) 611–625.
[5] P. Common, Tensor decompositions, state of the art and applications, in: McWhirter, Proudler (Eds.), Mathematics in Signal Processing V, Oxford University Press, Oxford, UK, 2002, pp. 1–24, hal-00347139.
[6] J. Cooper, A. Dutle, Spectra of uniform hypergraphs, Linear Algebra Appl. 436 (2012) 3268–3292.
[7] D. Dahl, J.M. Leinass, J. Myrheim, E. Ovrum, A tensor product matrix approximation problem in quantum physics, Linear Algebra Appl. 420 (2007) 711–725.
[8] L. De Lathauwer, B. De Moor, J. Vandewalle, On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors, SIAM J. Matrix Anal. Appl. 21 (2000) 1324–1342.
[9] P. Drineas, L.-H. Lim, A multilinear spectral theory of hypergraphs and expander hypergraphs, 2005.
[10] S. Friedland, S. Gaubert, L. Han, Perron–Frobenius theorem for nonnegative multilinear forms and extensions, Linear Algebra Appl. 438 (2) (2013) 738–749.
[11] S. Gaubert, J. Gunawardena, The Perron–Frobenius theorem for homogeneous, monotone functions, Trans. Amer. Math. Soc. 356 (2004) 4931–4950.
[12] B. He, Inexact implicit methods for monotone general variational inequalities, Math. Program. 86 (1999) 199–217.
[13] S. Hu, Z.-H. Huang, L. Qi, Finding the spectral radius of a nonnegative tensor, http://arxiv.org/abs/1111.2138v1.
[14] S. Hu, L. Qi, Algebraic connectivity of an even uniform hypergraph, J. Comb. Optim. 24 (2012) 564–579.
[15] L.-H. Lim, Singular values and eigenvalues of tensors: a variational approach, in: Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, vol. 1, 2005, pp. 129–132.
[16] L.-H. Lim, Multilinear pagerank: measuring higher order connectivity in linked objects, in: The Internet: Today and Tomorrow, 2005.
[17] Y. Liu, G. Zhou, N.F. Ibrahim, An always convergent algorithm for the largest eigenvalue of an irreducible nonnegative tensor, J. Comput. Appl. Math. 235 (2010) 286–292.
[18] Q. Ni, L. Qi, F. Wang, An eigenvalue method for testing positive definiteness of a multivariate form, IEEE Trans. Automat. Control 53 (2008) 1096–1107.
[19] M. Ng, L. Qi, G. Zhou, Finding the largest eigenvalue of a non-negative tensor, SIAM J. Matrix Anal. Appl. 31 (2009) 1090–1099.
[20] K. Pearson, Essentially positive tensors, Int. J. Algebra 4 (2010) 421–427.
[21] L. Qi, Eigenvalues of a real supersymmetric tensor, J. Symbolic Comput. 40 (2005) 1302–1324.
[22] L. Qi, W. Sun, Y. Wang, Numerical multilinear algebra and its applications, Front. Math. China 2 (2007) 501–526.
[23] S. Ragnarsson, C.F. Van Loan, Block tensors and symmetric embeddings, Linear Algebra Appl. 438 (2) (2013) 853–874.
[24] R. Varga, Matrix Iterative Analysis, Springer-Verlag, 2000.
[25] Y. Wang, M. Aron, A reformulation of the strong ellipticity conditions for unconstrained hyperelastic media, J. Elasticity 44 (1996) 89–96.
[26] Q. Yang, Y. Yang, Further results for Perron–Frobenius Theorem for nonnegative tensors II, SIAM J. Matrix Anal. Appl. 32 (2011) 1236–1250.
[27] Y. Yang, Q. Yang, Further results for Perron–Frobenius Theorem for nonnegative tensors, SIAM J. Matrix Anal. Appl. 31 (2010) 2517–2530.
[28] Y. Yang, Q. Yang, Singular values of nonnegative rectangular tensors, Front. Math. China 6 (2011) 363–378.
[29] Y. Yang, Q. Yang, Y. Li, An algorithm to find the spectral radius of nonnegative tensors and its convergence analysis, preprint, arXiv:1102.2668, 2011.
[30] Q. Yang, J. Zhao, The projection methods for the split feasibility problem, Numer. Acad. Sin. 2 (2006) 11–22.
[31] G. Zhou, L. Caccetta, L. Qi, Convergence of an algorithm for the largest singular value of a nonnegative rectangular tensor, Linear Algebra Appl. 438 (2) (2013) 959–968.
[32] L. Zhang, Linear convergence of an algorithm for computing the largest singular value of a nonnegative rectangular tensor, Front. Math. China 8 (2013) 141–153.
[33] L. Zhang, L. Qi, Linear convergence of an algorithm for computing the largest eigenvalue of a nonnegative tensor, Numer. Linear Algebra Appl. 19 (2012) 830–841.
[34] G. Zhou, L. Qi, S. Wu, Efficient algorithms for computing the largest eigenvalue of nonnegative tensor, Front. Math. China 8 (2013) 155–168.