CSIAM Trans. Appl. Math.
doi: 10.4208/csiam-am.2021.nla.07

Vol. **2**, No. 2, pp. 376-394
June 2021

# The Projected Newton Iteration Approach for Computing the Nonnegative Z-Eigenpairs of Nonnegative Tensors

Peihuan Bi[1], Wen Li[1,*], Dongdong Liu[2] and Mingqing Xiao[3]

[1] *School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China.*
[2] *School of Applied Mathematics, Guangdong University of Technology, Guangzhou 510006, China.*
[3] *Department of Mathematics, Southern Illinois University Carbondale, IL 62901, USA.*

Received 22 May 2020; Accepted 28 September 2020

**Abstract.** In this paper, we propose a new projected Newton iteration for computing the nonnegative Z-eigenpairs of nonnegative tensors. We show that the required iteration has a local quadratic convergence. More specially, the formulation aims to solve the tensor equation arising from the multilinear PageRank problem. Numerical experiments are provided to illustrate the effectiveness and superiority of the proposed approach.

**AMS subject classifications**: 65F15, 65F50

**Key words**: Nonnegative tensor, nonnegative Z-eigenpair, local quadratic convergence, multilinear PageRank.

## 1 Introduction

The eigenvalue problems of tensors have received much attention in recent years because of their wide applications such as medical imaging [2,5], higher order Markov chains [14], blind source separation [9], etc. Unlike the matrix case, there are several definitions of tensor eigenvalues, e.g., H-eigenvalues [15,18], Z-eigenvalues [15,18], D-eigenvalues [19], based on the practical problems. In this paper, we focus on computing the Z-eigenvalues and the corresponding Z-eigenvectors by targeting on the applications to the PageRank problem.

*Corresponding author. *Email addresses:* bipeihuan@m.scnu.edu.cn (P. Bi), liwen@m.scnu.edu.cn (W. Li), ddliu@gdut.edu.cn (D. Liu), mxiao@siu.edu (M. Xiao)

 

Currently, there are many proposed approaches in terms of iteration formulation in literature to compute one or more Z-eigenpairs of tensors with special structures, such as nonnegative tensors and symmetric tensors. The challenge is, even if a tensor is symmetric and/or nonnegative, that there may be more than one eigenpair. This leads to the impossibility for the global convergence in general. One of the notable methods for computing Z-eigenpairs of symmetric tensors is the so-called shifted power method (SS-HOPM) proposed by Kolda and Mayo [10]. The convergence rate of SS-HOPM depends on the choice of the shift. If the shift is not properly chosen, the method will converge slowly or even diverge in some occasions. To address this issue, Kolda and Mayo in [11] further added an adaptive procedure for choosing the shift. Most recently, Zhao et al. [20] proposed a modified normalized Newton method (MNNM) for computing Z-eigenpairs of symmetric tensors which can be convergent cubically. For the nonnegative tensors, Guo et al. [7] proposed a modified Newton iteration (MNI) to find some positive Z-eigenpairs and showed that their method has a local quadratic convergence under appropriate assumptions. In this paper, by reexamining the existing approaches, we develop a new approach via the projected Newton iteration for computing the nonnegative Z-eigenpairs of nonnegative tensors which improves the convergence rate of MNI.

The rest of the paper is organized as follows. In Section 2, we introduce some necessary notions as well as some preliminary results related to tensors, and provide a review for MNI. In Section 3, we propose a projected Newton iteration (PNI) for computing the nonnegative Z-eigenpairs of nonnegative tensors and show its local quadratic convergence. In Section 4, numerical experiments are provided to demonstrate the effectiveness and convergent behavior of the proposed PNI. In particular, we apply PNI to the multilinear PageRank. Finally, concluding remarks are given in Section 5.

## 2 Preliminaries

In this section, we recall some definitions and properties related to tensors, we also briefly review the MNI method for computing nonnegative Z-eigenpairs of nonnegative tensors in literature.

### 2.1 Notations and definitions

Let $\mathbb{R}$ be the real field, $\langle n\rangle=\{1,\cdots,n\}$ and $\mathcal{A}\in\mathbb{R}^{n\times\cdots\times n}$ be a tensor of order $m$ and dimension $n$, with entries $\mathcal{A}_{i_1\cdots i_m}$, where $i_1,\cdots,i_m\in\langle n\rangle$. We say that $\mathcal{A}$ is nonnegative, if $\mathcal{A}_{i_1\cdots i_m}\geq 0$ for $i_1,\cdots,i_m\in\langle n\rangle$ element wise. Throughout this paper, we denote the set of all real nonnegative tensors of order $m$ and dimension $n$ by $\mathbb{R}_+^{[m,n]}$. $\mathcal{A}$ is called semisymmetric [17] if the value of its entries $\mathcal{A}_{i_1\cdots i_m}$ is invariant under any permutation of their indices $i_2,\cdots,i_m$.

For real vectors $\mathbf{v}=(v_1,\cdots,v_n)$ and $\mathbf{w}=(w_1,\cdots,w_n)$, we define the following element-wise operations:

$$\max\{\mathbf{w},\mathbf{v}\} = (\max(w_1,v_1),\cdots,\max(w_n,v_n))^T,$$
$$\min\{\mathbf{w},\mathbf{v}\} = (\min(w_1,v_1),\cdots,\min(w_n,v_n))^T.$$

When $\mathbf{v}=\mathbf{0}$, we denote $\max\{\mathbf{w},\mathbf{v}\}$ and $\min\{\mathbf{w},\mathbf{v}\}$ by $(\mathbf{w})_+$ and $(\mathbf{w})_-$, respectively. For a pair of nonnegative vectors $\mathbf{w}$ and $\mathbf{v}$, $\mathbf{v}\neq\mathbf{0}$, we define

$$\max\left(\frac{\mathbf{w}}{\mathbf{v}}\right) = \begin{cases} \max\left\{\max\limits_{i\in\mathbb{S}_2}\left(\frac{w_i}{v_i}\right), \max\limits_{i\in\mathbb{S}_1\setminus(\mathbb{S}_1\cap\mathbb{S}_2)}(w_i)\right\} & \text{if } \mathbb{S}_1\setminus(\mathbb{S}_1\cap\mathbb{S}_2)\neq\emptyset, \\ \max\limits_{i\in\mathbb{S}_2}\left(\frac{w_i}{v_i}\right) & \text{if } \mathbb{S}_1\setminus(\mathbb{S}_1\cap\mathbb{S}_2)=\emptyset, \end{cases} \tag{2.1}$$

$$\min\left(\frac{\mathbf{w}}{\mathbf{v}}\right) = \begin{cases} 0 & \text{if } \mathbb{S}_1\setminus(\mathbb{S}_1\cap\mathbb{S}_2)\neq\emptyset, \\ \min\limits_{i\in\mathbb{S}_2}\left(\frac{w_i}{v_i}\right) & \text{if } \mathbb{S}_1\setminus(\mathbb{S}_1\cap\mathbb{S}_2)=\emptyset, \end{cases} \tag{2.2}$$

where $\mathbb{S}_1\subset\langle n\rangle$ and $\mathbb{S}_2\subset\langle n\rangle$ are the index sets of all nonzero elements of $\mathbf{w}$ and $\mathbf{v}$, respectively. For a real-valued, $m$th-order, $n$-dimensional tensor $\mathcal{A}$, a real-valued $n$-vector $\mathbf{x}$ and an integer $0\le r\le m-1$, the $(m-r)$-times product of $\mathcal{A}$ and $\mathbf{x}$ is a $r$th-order $n$-dimensional tensor, denoted by $\mathcal{A}\mathbf{x}^{m-r}$, which is defined by the following construction:

$$(\mathcal{A}\mathbf{x}^{m-r})_{i_1\cdots i_r} = \sum_{i_{r+1},\cdots,i_m=1}^{n} \mathcal{A}_{i_1\cdots i_m} x_{i_{r+1}}\cdots x_{i_m}, \quad 1\le i_1,\cdots,i_r\le n. \tag{2.3}$$

In particular, $\mathcal{A}\mathbf{x}^{m-1}$ is an $n$-vector with entries:

$$(\mathcal{A}\mathbf{x}^{m-1})_i = \sum_{i_2,\cdots,i_m=1}^{n} \mathcal{A}_{ii_2\cdots i_m} x_{i_2}\cdots x_{i_m}, \quad 1\le i\le n, \tag{2.4}$$

and $\mathcal{A}\mathbf{x}^{m-2}$ is an $n\times n$ matrix with entries:

$$(\mathcal{A}\mathbf{x}^{m-2})_{ij} = \sum_{i_3,\cdots,i_m=1}^{n} \mathcal{A}_{iji_3\cdots i_m} x_{i_3}\cdots x_{i_m}, \quad 1\le i,j\le n. \tag{2.5}$$

When $\mathcal{A}$ is a semisymmetric tensor, the gradient of $\mathcal{A}\mathbf{x}^{m-1}$ can be found [17]

$$\nabla_x(\mathcal{A}\mathbf{x}^{m-1}) = (m-1)\mathcal{A}\mathbf{x}^{m-2}. \tag{2.6}$$

Here we consider the one introduced as Z-eigenpairs in [18] and $l^2$-eigenpairs in [15].

**Definition 2.1** ([15,18])**.** *For $\mathcal{A}\in\mathbb{R}^{[m,n]}$, we say that $(\mathbf{x},\lambda)\in(\mathbb{R}^n\setminus\{\mathbf{0}\})\times\mathbb{R}$ is a Z-eigenpair of $\mathcal{A}$ if*

$$\mathcal{A}\mathbf{x}^{m-1} = \lambda\mathbf{x}, \|\mathbf{x}\|=1. \tag{2.7}$$

It is worthy to mention here that unlike the definition of H-eigenpair, the definition of Z-eigenpair requires $\|\mathbf{x}\|=1$ with a given vector norm. This is because $c\mathbf{x}$ is not necessarily a Z-eigenvector for any $c \neq 0$ even if $\mathbf{x}$ is a Z-eigenvector. It is easy to verify that if the order $m$ of $\mathcal{A}$ is even, $(-\mathbf{x},\lambda)$ is a Z-eigenpair if and only if $(\mathbf{x},\lambda)$ is a Z-eigenpair. If the order $m$ of $\mathcal{A}$ is odd, then $(-\mathbf{x},-\lambda)$ is a Z-eigenpair if and only if $(\mathbf{x},\lambda)$ is a Z-eigenpair. In other words, Z-eigenpair exists in pairs. If the 2-norm is used, a Z-eigenpair is called $Z_2$-eigenpair. If the 1-norm is used, a Z-eigenpair is called $Z_1$-eigenpair. As noted in [4], $(\mathbf{x},\lambda)$ is a $Z_1$-eigenpair if and only if

$$\left(\frac{\mathbf{x}}{\|\mathbf{x}\|_2},\frac{\lambda}{\|\mathbf{x}\|_2^{m-2}}\right)$$

is a $Z_2$-eigenpair. Without further illustrations, in this paper a Z-eigenpair means a $Z_1$-eigenpair.

The following result is given in Theorem 2.5 of [3]:

**Theorem 2.1** ([3]). *If $\mathcal{A}\in\mathbb{R}_+^{[m,n]}$, then $\mathcal{A}$ has a nonnegative Z-eigenpair.*

As we know that the multilinear PageRank [6] is an important application of a Z-eigenpair of nonnegative tensors, which can be formulated as:

$$\mathbf{x}=\alpha\mathcal{P}\mathbf{x}^{m-1}+(1-\alpha)\mathbf{v}, \tag{2.8}$$

where $\mathcal{P}$ and $\mathbf{v}$ are a given stochastic tensor [14] and a stochastic vector, respectively, $\alpha\in(0,1)$ is a given parameter and $\mathbf{x}$ is an unknown stochastic vector to be solved. (2.8) can be rewritten as the following multilinear systems:

$$\mathbf{x}=\mathcal{A}_\alpha\mathbf{x}^{m-1}, \quad \|\mathbf{x}\|_1=1, \tag{2.9}$$

where $\mathcal{A}_\alpha=\alpha\mathcal{P}+(1-\alpha)\mathcal{V}$ and $\mathcal{V}=(\mathcal{V}_{i_1i_2\cdots i_m})$ with $\mathcal{V}_{i_1i_2\cdots i_m}=v_{i_1}$, $\forall i_2,\cdots,i_m$. It is easy to verify that $\mathcal{A}_\alpha$ is also a stochastic tensor.

In [6], Gleich et al. presented a simplified uniqueness condition, i.e., if $\alpha<\frac{1}{m-1}$, the multilinear PageRank vector $\mathbf{x}$ in (2.8) is unique. More tighter uniqueness conditions for the multilinear PageRank vector have been obtained recently (see, e.g., [12,13]). For (2.9), Li et al. [14] proposed a uniqueness condition. Let

$$\delta_m=\min_{\mathbb{S}\subset\langle n\rangle}\left\{\min_{i_2,i_3,\cdots,i_m\in\langle n\rangle}\sum_{i\in\mathbb{S}}\mathcal{A}_{ii_2\cdots i_m}+\min_{i_2,i_3,\cdots,i_m\in\langle n\rangle}\sum_{i\in\mathbb{S}'}\mathcal{A}_{ii_2\cdots i_m}\right\}, \tag{2.10}$$

where $\mathbb{S}$ is a proper subset of $\langle n\rangle$ and $\mathbb{S}'$ is its complementary set in $\langle n\rangle$, i.e., $\mathbb{S}'=\langle n\rangle\backslash\mathbb{S}$. It is easy to check that $\delta_m\le 1$ if $\mathcal{A}$ is a stochastic tensor.

**Theorem 2.2** ([14]). *Suppose that $\mathcal{A}$ is an mth-order and n-dimensional stochastic tensor. If $\delta_m>\frac{m-2}{m-1}$, then the model $\mathcal{A}\mathbf{x}^{m-1}=\mathbf{x}$ has a unique solution.*

The following is the main characteristic about the Z-eigenvalue problem for stochastic tensors.

**Theorem 2.3** ([4]). *Let $\mathcal{A}\in\mathbb{R}_+^{[m,n]}$ be a stochastic tensor. Then, 1 is the unique Z-eigenvalue of $\mathcal{A}$ with a corresponding nonnegative eigenvector* $\mathbf{x}$.

In this paper, we focus on the computation of the eigenpair $(\mathbf{x}^*,\lambda^*)$ with $\mathbf{x}^*\geq 0$ and $\|\mathbf{x}^*\|_1=\mathbf{e}^T\mathbf{x}^*=1$, where $\mathbf{e}=[1,\cdots,1]^T$. Define a vector-valued function $\mathbf{f}:\mathbb{R}_+^{n+1}\to\mathbb{R}^{n+1}$ as follows:

$$\mathbf{f}(\mathbf{x},\lambda)=\begin{bmatrix}\mathbf{r}(\mathbf{x},\lambda)\\ \mathbf{e}^T\mathbf{x}-1\end{bmatrix}, \tag{2.11}$$

where $\mathbf{r}:\mathbb{R}_+^{n+1}\to\mathbb{R}^n$ is defined as

$$\mathbf{r}(\mathbf{x},\lambda)=\lambda\mathbf{x}-\mathcal{A}\mathbf{x}^{m-1}. \tag{2.12}$$

Then the Jacobian of $\mathbf{f}(\mathbf{x},\lambda)$ is given by:

$$\mathbf{Jf}(\mathbf{x},\lambda)=\begin{bmatrix}\lambda\mathbf{I}-\mathbf{T}(\mathbf{x}) & \mathbf{x}\\ \mathbf{e}^T & 0\end{bmatrix}, \tag{2.13}$$

where the entries of $\mathbf{T}(\mathbf{x})$ can be expressed as

$$\mathbf{T}(\mathbf{x})_{ij}=\frac{\partial}{\partial x_j}\left(\mathcal{A}\mathbf{x}^{m-1}\right)_i, \quad 1\leq i,j\leq n. \tag{2.14}$$

In general, (2.14) is a complex expression. Without loss of generality, we assume that $\mathcal{A}$ is semisymmetric. Thus, (2.14) reduces to (2.6). Further, if we denote

$$\bar{\mathcal{A}}_{ii_2\cdots i_m}=\frac{1}{(m-1)!}\sum_{(j_2\cdots j_m)\in G}\mathcal{A}_{ij_2\cdots j_m}, \tag{2.15}$$

where $G$ is denoted as a set of all different permutations of $(i_2,i_3,\cdots,i_m)$, then a simple calculation yields

$$\mathcal{A}\mathbf{x}^{m-1}=\bar{\mathcal{A}}\mathbf{x}^{m-1}. \tag{2.16}$$

Therefore

$$\mathcal{A}\mathbf{x}^{m-1}=\lambda\mathbf{x} \Leftrightarrow \bar{\mathcal{A}}\mathbf{x}^{m-1}=\lambda\mathbf{x}. \tag{2.17}$$

By Theorem 2.1, a nonnegative tensor $\mathcal{A}$ has a nonnegative eigenpair $(\mathbf{x}_*,\lambda_*)$, which may be approximated by using Newton's iteration to solve $\mathbf{f}(\mathbf{x},\lambda)=\mathbf{0}$, where $\mathbf{f}$ is defined in (2.11). $\mathbf{Jf}(\mathbf{x},\lambda)$ satisfies a Lipschitz continuity in a neighborhood of $(\mathbf{x}_*,\lambda_*)$ because its Fréchet derivative is continuous. We assume that

$$\mathbf{Jf}(\mathbf{x}_*,\lambda_*)=\begin{bmatrix}\lambda_*\mathbf{I}-\mathbf{T}(\mathbf{x}_*) & \mathbf{x}_*\\ \mathbf{e}^T & 0\end{bmatrix}, \tag{2.18}$$

is nonsingular. If $(\widehat{\mathbf{x}}_0,\widehat{\lambda}_0)$ is close enough to $(\mathbf{x}_*,\lambda_*)$, the sequence generated by Newton's iteration is well defined and converges to $(\mathbf{x}_*,\lambda_*)$. In our approach, we assume that $(\widehat{\mathbf{x}}_0,\widehat{\lambda}_0)$ is close enough to $(\mathbf{x}_*,\lambda_*)$. For a nonnegative pair $(\widehat{\mathbf{x}}_k,\widehat{\lambda}_k)$, the next pair $(\widehat{\mathbf{x}}_{k+1},\widehat{\lambda}_{k+1})$ resulting from Newton's iteration is given as follows:

$$\begin{bmatrix} \widehat{\lambda}_k\mathbf{I}-\mathbf{T}(\widehat{\mathbf{x}}_k) & \widehat{\mathbf{x}}_k \\ \mathbf{e}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}_k \\ \delta_k \end{bmatrix} = \begin{bmatrix} \mathbf{r}(\widehat{\mathbf{x}}_k,\widehat{\lambda}_k) \\ \mathbf{e}^T\widehat{\mathbf{x}}_k-1 \end{bmatrix}, \tag{2.19}$$

$$\widehat{\mathbf{x}}_{k+1}=\widehat{\mathbf{x}}_k-\mathbf{d}_k, \tag{2.20}$$

$$\widehat{\lambda}_{k+1}=\widehat{\lambda}_k-\delta_k. \tag{2.21}$$

Assume that $\widehat{\lambda}_k\mathbf{I}-\mathbf{T}(\widehat{\mathbf{x}}_k)$ is nonsingular and $\mathbf{e}^T\widehat{\mathbf{x}}_k=1$. Using block Gaussian elimination for (2.19) gets

$$\left(\mathbf{e}^T\widehat{\mathbf{w}}_k\right)\delta_k=\mathbf{e}^T\left(\widehat{\lambda}_k\mathbf{I}-\mathbf{T}(\widehat{\mathbf{x}}_k)\right)^{-1}\mathbf{r}(\widehat{\mathbf{x}}_k,\widehat{\lambda}_k), \tag{2.22}$$

where

$$\widehat{\mathbf{w}}_k=\left(\widehat{\lambda}_k\mathbf{I}-\mathbf{T}(\widehat{\mathbf{x}}_k)\right)^{-1}\widehat{\mathbf{x}}_k. \tag{2.23}$$

Since

$$\begin{aligned}\mathbf{r}\left(\widehat{\mathbf{x}}_k,\widehat{\lambda}_k\right)&=\frac{1}{m-1}\left((m-2)\widehat{\lambda}_k\widehat{\mathbf{x}}_k+\widehat{\lambda}_k\widehat{\mathbf{x}}_k-(m-1)\mathcal{A}\widehat{\mathbf{x}}_k^{m-1}\right)\\ &=\frac{1}{m-1}\left((m-2)\widehat{\lambda}_k\widehat{\mathbf{x}}_k+\left(\widehat{\lambda}_k\mathbf{I}-\mathbf{T}(\widehat{\mathbf{x}}_k)\right)\widehat{\mathbf{x}}_k\right),\end{aligned} \tag{2.24}$$

by (2.22), (2.23) and $\mathbf{e}^T\widehat{\mathbf{x}}_k=1$, we have

$$\left(\mathbf{e}^T\widehat{\mathbf{w}}_k\right)\left(\delta_k-\frac{m-2}{m-1}\widehat{\lambda}_k\right)=\frac{1}{m-1}, \tag{2.25}$$

which gives $\mathbf{e}^T\widehat{\mathbf{w}}_k\neq 0$ and

$$\delta_k=\frac{m-2}{m-1}\widehat{\lambda}_k+\frac{1}{(m-1)\mathbf{e}^T\widehat{\mathbf{w}}_k}. \tag{2.26}$$

Combining with (2.19) and (2.23)-(2.26) together gives

$$\mathbf{d}_k=\frac{1}{m-1}\widehat{\mathbf{x}}_k-\frac{1}{(m-1)\mathbf{e}^T\widehat{\mathbf{w}}_k}\widehat{\mathbf{w}}_k. \tag{2.27}$$

Hence the Newton iteration is given by:

$$\widehat{\mathbf{x}}_{k+1}=\widehat{\mathbf{x}}_k-\mathbf{d}_k=\frac{1}{m-1}\left((m-2)\widehat{\mathbf{x}}_k+\frac{1}{\mathbf{e}^T\widehat{\mathbf{w}}_k}\widehat{\mathbf{w}}_k\right), \tag{2.28}$$

$$\widehat{\lambda}_{k+1}=\widehat{\lambda}_k-\delta_k=\frac{1}{m-1}\left(\widehat{\lambda}_k-\frac{1}{\mathbf{e}^T\widehat{\mathbf{w}}_k}\right). \tag{2.29}$$

However it can not guarantee that $(\widehat{\mathbf{x}}_{k+1},\widehat{\lambda}_{k+1})$ generated by the Newton iteration is nonnegative. For overcoming this problem, Guo, et al. in [7] gave a modification of $\widehat{\mathbf{w}}_k$ in order to retain $(\widehat{\mathbf{x}}_{k+1},\widehat{\lambda}_{k+1})$ positive in the following subsection.

## 2.2 A modified Newton iteration

Guo et al. [7] proposed some modifications on Newton's iteration that has a good chance of finding a positive eigenpair starting from $(\widehat{\mathbf{x}}_0,\widehat{\lambda}_0)>0$. Either $\widehat{\mathbf{w}}_k\leq 0$ or $\widehat{\mathbf{w}}_k\geq 0$, $\widehat{\mathbf{x}}_{k+1}>0$. When $\widehat{\mathbf{w}}_k$ has both positive and negative components, they used a post-processing procedure which can avoid drastic changes. Let $s_k=\max(\widehat{\mathbf{w}}_k)\min(\widehat{\mathbf{w}}_k)$. The procedure is listed as follows:

$$\mathbf{w}_k=\begin{cases}(\widehat{\mathbf{w}}_k)_+ & \text{if } s_k<0 \text{ and } |\max(\widehat{\mathbf{w}}_k)|>|\min(\widehat{\mathbf{w}}_k)|,\\ (\widehat{\mathbf{w}}_k)_- & \text{if } s_k<0 \text{ and } |\max(\widehat{\mathbf{w}}_k)|\leq|\min(\widehat{\mathbf{w}}_k)|,\\ \widehat{\mathbf{w}}_k & \text{if } s_k\geq 0.\end{cases} \tag{2.30}$$

The modified $\mathbf{w}_k$ can guarantee that $\widehat{\mathbf{x}}_{k+1}>0$. To get a new approximation $\lambda_{k+1}>0$, they also gave the following modification:

$$\lambda_{k+1}=\begin{cases}\overline{\lambda}_{k+1} & \text{if } \widehat{\lambda}_{k+1}>\overline{\lambda}_{k+1},\\ \underline{\lambda}_{k+1} & \text{if } \widehat{\lambda}_{k+1}<\underline{\lambda}_{k+1},\\ \widehat{\lambda}_{k+1} & \text{if } \widehat{\lambda}_{k+1}\in[\underline{\lambda}_{k+1},\overline{\lambda}_{k+1}],\end{cases} \tag{2.31}$$

where

$$\underline{\lambda}_{k+1}=\min\left(\frac{\mathcal{A}\mathbf{x}_{k+1}^{m-1}}{\mathbf{x}_{k+1}}\right),\quad \overline{\lambda}_{k+1}=\max\left(\frac{\mathcal{A}\mathbf{x}_{k+1}^{m-1}}{\mathbf{x}_{k+1}}\right). \tag{2.32}$$

The following algorithm is the MNI algorithm given by [7].

They also gave the local quadratic convergence of Algorithm 1.

---

**Algorithm 1** [7] Modified Newton's iteration (MNI)

---

**Input:** $\mathbf{x}_0>0$, $tol>0$

**Output:** $\mathbf{x}_{k+1},\lambda_{k+1}$

1. $\mathbf{x}_0=\mathbf{x}_0/\|\mathbf{x}_0\|_1$.
2. Compute $\overline{\lambda}_0=\max\left(\frac{\mathcal{A}\mathbf{x}_0^{m-1}}{\mathbf{x}_0}\right)$ and $\underline{\lambda}_0=\min\left(\frac{\mathcal{A}\mathbf{x}_0^{m-1}}{\mathbf{x}_0}\right)$.
3. Choose $\lambda_0\in[\underline{\lambda}_0,\overline{\lambda}_0]$ such that $\lambda_0\mathbf{I}-\mathbf{T}(\mathbf{x}_0)$ is nonsingular.
4. **for** $k=0,1,2,\cdots$ until $\left\|\mathcal{A}\mathbf{x}_k^{m-1}-\lambda_k\mathbf{x}_k\right\|_1<tol$.
5. Solve the linear system $(\lambda_k\mathbf{I}-\mathbf{T}(\mathbf{x}_k))\widehat{\mathbf{w}}_k=\mathbf{x}_k$.
6. Determine the vector $\mathbf{w}_k$ by (2.30).
7. $\widehat{\mathbf{x}}_{k+1}=(m-2)\mathbf{x}_k+\mathbf{w}_k/\left(\mathbf{e}^T\mathbf{w}_k\right)$.
8. $\mathbf{x}_{k+1}=\widehat{\mathbf{x}}_{k+1}/\|\widehat{\mathbf{x}}_{k+1}\|_1$.
9. Compute $\overline{\lambda}_{k+1}=\max\left(\frac{\mathcal{A}\mathbf{x}_{k+1}^{m-1}}{\mathbf{x}_{k+1}}\right)$ and $\underline{\lambda}_{k+1}=\min\left(\frac{\mathcal{A}\mathbf{x}_{k+1}^{m-1}}{\mathbf{x}_{k+1}}\right)$.
10. Choose $\lambda_{k+1}\in[\underline{\lambda}_{k+1},\overline{\lambda}_{k+1}]$ such that $\lambda_{k+1}\mathbf{I}-\mathbf{T}(\mathbf{x}_{k+1})$ is nonsingular.
11. **end**

---

**Theorem 2.4** ([7]). *Let* $(\mathbf{x}_*,\lambda_*)$ *be a positive eigenpair of a nonnegative tensor* $\mathcal{A}$*, with* $\mathbf{Jf}(\mathbf{x}_*,\lambda_*)$ *in* (2.18) *being nonsingular, and let* $\{(\mathbf{x}_k,\lambda_k)\}$ *be generated by Algorithm* 1*. Suppose that* $(\mathbf{x}_{k_0},\lambda_{k_0})$ *is sufficiently close to* $(\mathbf{x}_*,\lambda_*)$ *for some* $k_0\geq 0$*. Then* $\mathbf{x}_k$ *converges to* $\mathbf{x}_*$ *quadratically and* $\lambda_k$ *converges to* $\lambda_*$ *quadratically.*

It is noted that the premise of Theorem 2.4 is to assume that $\widehat{\mathbf{w}}_k\geq 0$, i.e. $\mathbf{x}_{k+1}$ does not need to be modified.

# 3 The projected Newton iteration

For keeping $(\widehat{\mathbf{x}}_{k+1},\widehat{\lambda}_{k+1})$ nonnegative for Newton's iteration in (2.28) and (2.29), we propose a new Newton iteration with the projected technique, and then give the convergence analysis.

## 3.1 Projected Newton's iteration

For Algorithm 1, $\widehat{\mathbf{w}}_k$ has been modified, and then $\widehat{\mathbf{x}}_{k+1}$ is calculated through (2.28), resulting $\mathbf{x}_{k+1}$ being significantly changed. In the new proposed algorithm, we modify $\widehat{\mathbf{x}}_{k+1}$ directly without modifying $\widehat{\mathbf{w}}_k$ in Algorithm 1. Define a projection:

$$proj(\widehat{\mathbf{x}}_{k+1})=(\widehat{\mathbf{x}}_{k+1})_+/\|(\widehat{\mathbf{x}}_{k+1})_+\|_1. \tag{3.1}$$

The essence of such a projection is to project $\widehat{\mathbf{x}}_{k+1}$ onto the nearest 1-norm nonnegative vector. Since $\mathbf{e}^T\widehat{\mathbf{w}}_k\neq 0$ by (2.25), $\frac{1}{\mathbf{e}^T\widehat{\mathbf{w}}_k}\widehat{\mathbf{w}}_k$ in (2.28) has some positive elements, which makes $\|(\widehat{\mathbf{x}}_{k+1})_+\|_1\neq 0$. Based on the technique (3.1), we propose the following Algorithm 2, which is called the projected Newton iteration (PNI).

In PNI, we give the following modification such that $\lambda_{k+1}\mathbf{I}-\mathbf{T}(\mathbf{x}_{k+1})$ is nonsingular:

$$\lambda_{k+1}=\begin{cases}\widehat{\lambda}_{k+1}+\beta_{k+1}\left(\overline{\lambda}_{k+1}-\widehat{\lambda}_{k+1}\right) & \text{if } \widehat{\lambda}_{k+1}\leq\left(\underline{\lambda}_{k+1}+\overline{\lambda}_{k+1}\right)/2,\\ \widehat{\lambda}_{k+1}+\beta_{k+1}\left(\underline{\lambda}_{k+1}-\widehat{\lambda}_{k+1}\right) & \text{if } \widehat{\lambda}_{k+1}>\left(\underline{\lambda}_{k+1}+\overline{\lambda}_{k+1}\right)/2,\end{cases} \tag{3.2}$$

where $0\leq\beta_{k+1}\leq 1$ is a given constant. To get a better approximation $\lambda_{k+1}$ to $\lambda_*$, we can adjust $\beta_{k+1}$ such that $\lambda_{k+1}\in\left[\underline{\lambda}_{k+1},\overline{\lambda}_{k+1}\right]$.

**Remark 3.1.** By (2.1) and (2.2), there are two special cases to be illustrated for the proposed algorithm.

**(1)** It is seen that $\overline{\lambda}_{k+1}>\underline{\lambda}_{k+1}$ if one of the following conditions is satisfied:

- **(i)** there is an $i\in\langle n\rangle$ such that $(\mathbf{x}_{k+1})_i=0$ and $(\mathcal{A}\mathbf{x}_{k+1}^{m-1})_i\neq 0$;
- **(ii)** there are $i,j\in\langle n\rangle$, $(\mathbf{x}_{k+1})_i>0$ and $(\mathbf{x}_{k+1})_j>0$ such that $\frac{\left(\mathcal{A}\mathbf{x}_{k+1}^{m-1}\right)_i}{(\mathbf{x}_{k+1})_i}\neq\frac{\left(\mathcal{A}\mathbf{x}_{k+1}^{m-1}\right)_j}{(\mathbf{x}_{k+1})_j}$.

**(2)** $\overline{\lambda}_{k+1}=\underline{\lambda}_{k+1}$ if and only if $(\mathcal{A}\mathbf{x}_{k+1}^{m-1})_i=\overline{\lambda}_{k+1}(\mathbf{x}_{k+1})_i$, $\forall i\in\langle n\rangle$, i.e. $\mathcal{A}\mathbf{x}_{k+1}^{m-1}=\overline{\lambda}_{k+1}\mathbf{x}_{k+1}$.

For the case **(1)**, $\beta_{k+1}$ can be always adjusted such that $\lambda_{k+1}\mathbf{I}-\mathbf{T}(\mathbf{x}_{k+1})$ is nonsingular.

**Algorithm 2** The projected Newton iteration (PNI)

**Input:** $\mathbf{x}_0 > 0$, $tol > 0$

**Output:** $\mathbf{x}_{k+1}$, $\lambda_{k+1}$

1. $\mathbf{x}_0 = \mathbf{x}_0 / \|\mathbf{x}_0\|_1$.
2. Compute $\widehat{\lambda}_0 = \overline{\lambda}_0 = \max\left(\frac{\mathcal{A}\mathbf{x}_0^{m-1}}{\mathbf{x}_0}\right)$ and $\underline{\lambda}_0 = \min\left(\frac{\mathcal{A}\mathbf{x}_0^{m-1}}{\mathbf{x}_0}\right)$.
3. Choose $\beta_0$ in (3.2) such that $\lambda_0 \mathbf{I} - \mathbf{T}(\mathbf{x}_0)$ is nonsingular.
4. **for** $k = 0,1,2,\cdots$ until $\left\|\mathcal{A}\mathbf{x}_k^{m-1} - \lambda_k \mathbf{x}_k\right\|_1 < tol$.
5. Solve linear system $(\lambda_k \mathbf{I} - \mathbf{T}(\mathbf{x}_k))\widehat{\mathbf{w}}_k = \mathbf{x}_k$.
6. $\widehat{\mathbf{x}}_{k+1} = (m-2)\mathbf{x}_k + \frac{\widehat{\mathbf{w}}_k}{\mathbf{e}^T \widehat{\mathbf{w}}_k}$.
7. $\mathbf{x}_{k+1} = proj(\widehat{\mathbf{x}}_{k+1})$.
8. $\widehat{\lambda}_{k+1} = \frac{1}{m-1}\left(\lambda_k - \frac{1}{\mathbf{e}^T \widehat{\mathbf{w}}_k}\right)$.
9. Compute $\overline{\lambda}_{k+1} = \max\left(\frac{\mathcal{A}\mathbf{x}_{k+1}^{m-1}}{\mathbf{x}_{k+1}}\right)$ and $\underline{\lambda}_{k+1} = \min\left(\frac{\mathcal{A}\mathbf{x}_{k+1}^{m-1}}{\mathbf{x}_{k+1}}\right)$.
10. Choose $\beta_{k+1}$ in (3.2) such that $\lambda_{k+1}\mathbf{I} - \mathbf{T}(\mathbf{x}_{k+1})$ is nonsingular.
11. **end**

## 3.2 Convergence analysis of Algorithm 2

In this subsection, we will prove that PNI still has local quadratic convergence even $\mathbf{x}_{k+1}$ is modified. The following result is a direct consequence of Newton's iteration; see Theorem 5.1.2 in [8] for example.

**Lemma 3.1.** *Suppose that the matrix in* (2.18) *is nonsingular. Let* $(\widehat{\mathbf{x}}_{k+1}, \widehat{\lambda}_{k+1})$ *be obtained by Newton's iteration as in* (2.28) *and* (2.29)*, from* $(\mathbf{x}_k, \lambda_k)$ *instead of* $(\widehat{\mathbf{x}}_k, \widehat{\lambda}_k)$*. Then, there are* $\eta > 0$ *and* $c > 0$ *such that*

$$\left\|\begin{bmatrix} \widehat{\mathbf{x}}_{k+1} \\ \widehat{\lambda}_{k+1} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_* \\ \lambda_* \end{bmatrix}\right\|_1 \le c \left\|\begin{bmatrix} \mathbf{x}_k \\ \lambda_k \end{bmatrix} - \begin{bmatrix} \mathbf{x}_* \\ \lambda_* \end{bmatrix}\right\|_1^2, \tag{3.3}$$

*whenever*

$$\left\|\begin{bmatrix} \mathbf{x}_k \\ \lambda_k \end{bmatrix} - \begin{bmatrix} \mathbf{x}_* \\ \lambda_* \end{bmatrix}\right\|_1 < \eta.$$

Next, the following lemma will be used.

**Lemma 3.2.** *Let* $\mathbf{x} \in \mathbb{R}^n$ *with* $\|(\mathbf{x})_+\|_1 \neq 0$ *and* $\mathbf{y}$ *be a stochastic vector. If* $\widehat{\mathbf{x}} = proj(\mathbf{x})$*, then*

$$\|\widehat{\mathbf{x}} - \mathbf{y}\|_1 \le 4\|\mathbf{x} - \mathbf{y}\|_1. \tag{3.4}$$

*Proof.* Since $\mathbf{y}$ is a stochastic vector, we get

$$|\mathbf{e}^T \mathbf{x} - 1| = |\mathbf{e}^T(\mathbf{x} - \mathbf{y})| \le \|\mathbf{x} - \mathbf{y}\|_1.$$

Then we have

$$\|(\mathbf{x})_+\|_1+\|\mathbf{x}-\mathbf{y}\|_1\geq \mathbf{e}^T\mathbf{x}+|\mathbf{e}^T\mathbf{x}-1|\geq 1. \tag{3.5}$$

Note that

$$\|(\widehat{\mathbf{x}}-\mathbf{y})_+\|_1-\|(\widehat{\mathbf{x}}-\mathbf{y})_-\|_1=\mathbf{e}^T(\widehat{\mathbf{x}}-\mathbf{y})=\mathbf{e}^T\widehat{\mathbf{x}}-\mathbf{e}^T\mathbf{y}=0, \tag{3.6}$$

and

$$|\|(\mathbf{x}-\mathbf{y})_+\|_1-\|(\mathbf{x}-\mathbf{y})_-\|_1|=|\mathbf{e}^T\mathbf{x}-\mathbf{e}^T\mathbf{y}|\leq\|\mathbf{x}-\mathbf{y}\|_1. \tag{3.7}$$

Then, by (3.6), (3.5) and (3.7)

$$\begin{aligned}
\|\widehat{\mathbf{x}}-\mathbf{y}\|_1 &= 2\big\|(\widehat{\mathbf{x}}-\mathbf{y})_+\big\|_1\\
&\leq 2\|((\|(\mathbf{x})_+\|_1+\|\mathbf{x}-\mathbf{y}\|_1)\widehat{\mathbf{x}}-\mathbf{y})_+\|_1\\
&\leq 2\|(\|(\mathbf{x})_+\|_1\widehat{\mathbf{x}}-\mathbf{y})_+\|_1+2\|\mathbf{x}-\mathbf{y}\|_1\\
&= 2\|((\mathbf{x})_+-\mathbf{y})_+\|_1+2\|\mathbf{x}-\mathbf{y}\|_1\\
&= 2\|((\mathbf{x})_++(\mathbf{x})_--\mathbf{y})_+\|_1+2\|\mathbf{x}-\mathbf{y}\|_1\\
&= 2\|(\mathbf{x}-\mathbf{y})_+\|_1+2\|\mathbf{x}-\mathbf{y}\|_1\\
&\leq 4\|\mathbf{x}-\mathbf{y}\|_1.
\end{aligned}$$

This completes the proof. □

**Remark 3.2.** When $\mathbf{e}^T\mathbf{x}=1$, (3.5) and (3.7) can be rewritten respectively as

$$\|(\mathbf{x})_+\|_1\geq\mathbf{e}^T\mathbf{x}=1,$$

and

$$|\|(\mathbf{x}-\mathbf{y})_+\|_1-\|(\mathbf{x}-\mathbf{y})_-\|_1|=|\mathbf{e}^T\mathbf{x}-\mathbf{e}^T\mathbf{y}|=0.$$

Thus $\|\widehat{\mathbf{x}}-\mathbf{y}\|_1\leq\|\mathbf{x}-\mathbf{y}\|_1$, which is Lemma 4 in [16].

Since the modification of $\lambda_{k+1}$ in (3.2) should be small, it is suitable to assume that $\beta_{k+1}$ can be chosen to satisfy

$$\beta_{k+1}\leq\min\left\{1,\frac{\max\left\{\|\mathbf{x}_{k+1}-\mathbf{x}_*\|_1,|\widehat{\lambda}_{k+1}-\lambda_*|\right\}}{\max\left\{|\overline{\lambda}_{k+1}-\widehat{\lambda}_{k+1}|,|\underline{\lambda}_{k+1}-\widehat{\lambda}_{k+1}|\right\}}\right\}, \tag{3.8}$$

such that $\lambda_{k+1}\mathbf{I}-\mathbf{T}(\mathbf{x}_{k+1})$ is nonsingular. It is noted that

$$\max\left\{\left|\overline{\lambda}_{k+1}-\widehat{\lambda}_{k+1}\right|,\left|\underline{\lambda}_{k+1}-\widehat{\lambda}_{k+1}\right|\right\}\neq 0,$$

unless $(\mathbf{x}_{k+1},\widehat{\lambda}_{k+1})$ is already an eigenpair. With the above assumption, we prove the local quadratic convergence of Algorithm 2.

**Theorem 3.1.** *Let* $(\mathbf{x}_*,\lambda_*)$ *be a nonnegative eigenpair of a nonnegative tensor* $\mathcal{A}$*, with* $\mathbf{Jf}(\mathbf{x}_*,\lambda_*)$ *being nonsingular, and let* $\{(\mathbf{x}_k,\lambda_k)\}$ *be generated by Algorithm* 2*. Suppose that* $(\mathbf{x}_{k_0},\lambda_{k_0})$ *is sufficiently close to* $(\mathbf{x}_*,\lambda_*)$ *for some* $k_0\geq 0$*. Then* $\begin{bmatrix}\mathbf{x}_k\\ \lambda_k\end{bmatrix}$ *converges to* $\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}$ *quadratically.*

*Proof.* For $k=k_0$, by the assumption, $\epsilon$ can be taken to be small enough such that

$$\left\|\begin{bmatrix}\mathbf{x}_k\\ \lambda_k\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1<\epsilon.$$

By Lemma 3.1, there exists a $c>0$ such that (3.3) holds. By (3.2) and (3.8), we have

$$\begin{aligned}|\lambda_{k+1}-\lambda_*|&\leq|\widehat{\lambda}_{k+1}-\lambda_*|+\beta_{k+1}\max\left\{\left|\overline{\lambda}_{k+1}-\widehat{\lambda}_{k+1}\right|,\left|\underline{\lambda}_{k+1}-\widehat{\lambda}_{k+1}\right|\right\}\\ &\leq 2|\widehat{\lambda}_{k+1}-\lambda_*|+\|\mathbf{x}_{k+1}-\mathbf{x}_*\|_1.\end{aligned}$$

Note that $\mathbf{x}_*$ is a stochastic vector and $\|(\widehat{\mathbf{x}}_{k+1})_+\|_1\neq 0$. Then by (3.4), we obtain

$$\left\|\begin{bmatrix}\mathbf{x}_{k+1}\\ \lambda_{k+1}\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1=\|\mathbf{x}_{k+1}-\mathbf{x}_*\|_1+|\lambda_{k+1}-\lambda_*|\leq 8\|\widehat{\mathbf{x}}_{k+1}-\mathbf{x}_*\|_1+2|\widehat{\lambda}_{k+1}-\lambda_*|.$$

Then, it is seen that

$$\left\|\begin{bmatrix}\mathbf{x}_{k+1}\\ \lambda_{k+1}\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1\leq 8\left\|\begin{bmatrix}\widehat{\mathbf{x}}_{k+1}\\ \widehat{\lambda}_{k+1}\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1. \tag{3.9}$$

By (3.9) and (3.3), we get

$$\left\|\begin{bmatrix}\mathbf{x}_{k+1}\\ \lambda_{k+1}\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1\leq 8c\left\|\begin{bmatrix}\mathbf{x}_k\\ \lambda_k\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1^2.$$

We can take

$$\epsilon\leq\frac{1}{8c},$$

such that

$$\left\|\begin{bmatrix}\mathbf{x}_{k+1}\\ \lambda_{k+1}\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1<8c\epsilon^2\leq\epsilon.$$

We can then repeat the above process to obtain

$$\left\|\begin{bmatrix}\mathbf{x}_{k+1}\\ \lambda_{k+1}\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1\leq d\left\|\begin{bmatrix}\mathbf{x}_k\\ \lambda_k\end{bmatrix}-\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}\right\|_1^2,\quad\forall k\geq k_0,$$

where $d=8c$. Thus, $\begin{bmatrix}\mathbf{x}_k\\ \lambda_k\end{bmatrix}$ converges to $\begin{bmatrix}\mathbf{x}_*\\ \lambda_*\end{bmatrix}$ quadratically. □

# 4 Numerical experiments

In this section, we present some numerical results to demonstrate the advantages of the proposed algorithm. All the experiments are performed under Windows 10 running on a desktop (Intel Core i5, @ 2.40GHz, 8.00G RAM). The code is written and executed in MATLAB R2019a. All the nonsingular linear equations are solved by *mldivide* function of MATLAB. Besides, the products (2.3) is computed by the *ttv* function of the package Tensor Toolbox 2.6 [1] for all the experiments.

We will adjust $\lambda_k$ by (3.2) when the 2-norm condition number of $\lambda_k\mathbf{I}-\mathbf{T}(\mathbf{x_k})$ is larger than $10^{13}$. We use $\beta_k=10^{-12}/(\overline{\lambda}_k-\underline{\lambda}_k)$, $\forall k\geq 0$ in our experiments. It turns out that the adjustment is needed only for Example 2. The termination condition of the iteration is

$$\|\mathcal{A}\mathbf{x}_k^{m-1}-\lambda_k\mathbf{x}_k\|_1<10^{-12}, \tag{4.1}$$

or the number of steps exceeds the maximum $k_{max}=1000$. If $\overline{\lambda}_k=\underline{\lambda}_k$, MNI and PNI stop and output $\mathbf{x_k}$ and $\overline{\lambda}_k$.

## 4.1 Numerical results for computing Z-eigenpairs

We use three examples in [7] to demonstrate the effectiveness of PNI and compare it with MNI.

**Example 4.1** **([7]).** Consider $\mathcal{A}\in\mathbb{R}_+^{[4,2]}$ defined by $\mathcal{A}_{1111}=1.1$, $\mathcal{A}_{2222}=1.2$, $\mathcal{A}_{1112}=\mathcal{A}_{1222}=0.25$, and $\mathcal{A}_{ijkl}=0$ elsewhere.

The tensor has three nonnegative Z-eigenpairs:

$$\begin{aligned}
\left(\mathbf{x}^{(1)},\lambda^{(1)}\right)&\approx\left([0.1874,0.8126]^T,0.7923\right),\\
\left(\mathbf{x}^{(2)},\lambda^{(2)}\right)&=\left([1,0]^T,1.1\right),\\
\left(\mathbf{x}^{(3)},\lambda^{(3)}\right)&\approx\left([0.4412,0.5588]^T,0.3746\right).
\end{aligned}$$

For this example, we use 5000 random initial vectors to find different nonnegative eigenpairs. Each time, the sequence generated by the iteration converges to one of the three eigenpairs. In Table 1, "Occurr" denotes the number of occurrences with convergence to a particular eigenpair. For each eigenpair, "A-Sign" denotes the average number of times with $s_k=\max(\widehat{\mathbf{w}}_k)\min(\widehat{\mathbf{w}}_k)<0$ (this tells us the frequency that MNI needs to modify $\widehat{\mathbf{w}}_k$), "A-IT" stands for the average number of iterations to achieve convergence, "A-CPU" represents the average time of iterations to achieve convergence, and "A-Err" refers to the average residual error when the iteration is terminated.

It is noted that when $s_k\geq 0$, both MNI and PNI are actually Newton's iteration, without modifying $\widehat{\mathbf{w}}_k$ or $\widehat{\mathbf{x}}_k$. From Table 1, we can see that MNI needs more iterative steps to approach the second eigenpair because there is a zero component in the second eigenvector. Moreover, there are many times of occurrence of $s_k<0$ when MNI finds the second

Table 1: Numerical results for Example 4.1.

| | MNI | | | | | PNI | | | |
|---|---|---|---|---|---|---|---|---|---|
| | A-Sign | Occurr | A-CPU | A-IT | A-Err | Occurr | A-CPU | A-IT | A-Err |
| $(\mathbf{x}^{(1)},\lambda^{(1)})$ | 0.2840 | 1137 | 0.0059 | 4.8795 | 5.5762e-14 | 1381 | 0.0074 | 5.4106 | 5.4521e-14 |
| $(\mathbf{x}^{(2)},\lambda^{(2)})$ | **64.0757** | 872 | 0.0706 | 64.0757 | 8.1657e-13 | 962 | **0.0024** | **1.0187** | 0 |
| $(\mathbf{x}^{(3)},\lambda^{(3)})$ | 0.3697 | 2991 | 0.0045 | 4.2594 | 4.4568e-14 | 2657 | 0.0056 | 4.6797 | 4.4453e-14 |

eigenvector. It is obvious that PNI performs better than MNI in terms of IT and CPU when $s_k<0$ occurs frequently.

**Example 4.2** ([7])**.** Consider the diagonal tensor $\mathcal{A}\in\mathbb{R}_+^{[3,5]}$ with just three nonzero entries:

$$\mathcal{A}_{111}=1,\quad \mathcal{A}_{333}=2,\quad \mathcal{A}_{555}=3.$$

The tensor has seven nonnegative Z-eigenvectors corresponding to seven positive eigenvalues and infinitely many nonnegative eigenvectors corresponding to the eigenvalue 0:

$$\begin{aligned}
\left(\mathbf{x}^{(1)},\lambda^{(1)}\right)&=\left([6/11,0,3/11,0,2/11]^T,6/11\right),\\
\left(\mathbf{x}^{(2)},\lambda^{(2)}\right)&=\left([2/3,0,1/3,0,0]^T,2/3\right),\\
\left(\mathbf{x}^{(3)},\lambda^{(3)}\right)&=\left([3/4,0,0,0,1/4]^T,3/4\right),\\
\left(\mathbf{x}^{(4)},\lambda^{(4)}\right)&=\left([1,0,0,0,0]^T,1\right),\\
\left(\mathbf{x}^{(5)},\lambda^{(5)}\right)&=\left([0,0,0.6,0,0.4]^T,1.2\right),\\
\left(\mathbf{x}^{(6)},\lambda^{(6)}\right)&=\left([0,0,1,0,0]^T,2\right),\\
\left(\mathbf{x}^{(7)},\lambda^{(7)}\right)&=\left([0,0,0,0,1]^T,3\right),\\
\left(\mathbf{x}^{(8)},\lambda^{(8)}\right)&=\left([0,x_2,0,x_4,0]^T,0\right).
\end{aligned}$$

For this example, we also use 5000 random initial vectors to find different nonnegative eigenpairs. It is noted that every eigenvector has zero components. From Table 2, we can see that MNI modify $\widehat{\mathbf{w}}_k$ frequently to approach every eigenpair. The more frequent $s_k<0$ occurs, the better PNI performs than MNI in terms of IT and CPU. We then take a random initial vector, with MNI and PNI convergence to $(\mathbf{x}^{(5)},\lambda^{(5)})$, and plot the errors $\|\mathbf{x}_k-\mathbf{x}^{(5)}\|_1$ and $|\lambda_k-\lambda^{(5)}|$ in Fig. 1. It can be seen that $\mathbf{x}_k$ and $\lambda_k$ obtained by PNI converge quadratically to $\mathbf{x}^{(5)}$ and $\lambda^{(5)}$ respectively. PNI is faster than MNI.

Table 2: Numerical results for Example 4.2.

| | MNI | | | | | PNI | | | |
|---|---|---|---|---|---|---|---|---|---|
| | A-Sign | Occurr | A-CPU | A-IT | A-Err | Occurr | A-CPU | A-IT | A-Err |
| $(\mathbf{x}^{(1)},\lambda^{(1)})$ | **37.8912** | 708 | 0.0289 | 37.8912 | 7.3743e-13 | 684 | **0.0055** | **5.8085** | 4.8107e-14 |
| $(\mathbf{x}^{(2)},\lambda^{(2)})$ | **38.5103** | 243 | 0.0283 | 38.5103 | 6.9232e-13 | 259 | **0.0052** | **5.2857** | 6.1025e-14 |
| $(\mathbf{x}^{(3)},\lambda^{(3)})$ | **38.7959** | 245 | 0.0295 | 38.7959 | 7.0473e-13 | 247 | **0.0050** | **5.5020** | 5.3170e-14 |
| $(\mathbf{x}^{(4)},\lambda^{(4)})$ | **39.3200** | 25 | 0.0294 | 39.3200 | 7.4668e-13 | 18 | **0.0023** | **1.3333** | 0 |
| $(\mathbf{x}^{(5)},\lambda^{(5)})$ | **39.5409** | 1002 | 0.0296 | 39.5409 | 6.8898e-13 | 1021 | **0.0051** | **5.4770** | 5.5802e-14 |
| $(\mathbf{x}^{(6)},\lambda^{(6)})$ | **40.5339** | 118 | 0.0298 | 40.5339 | 6.9919e-13 | 37 | **0.0023** | **1.5946** | 0 |
| $(\mathbf{x}^{(7)},\lambda^{(7)})$ | **40.9960** | 251 | 0.0324 | 40.9960 | 8.0492e-13 | 39 | **0.0022** | **1.4359** | 0 |
| $(\mathbf{x}^{(8)},\lambda^{(8)})$ | **1.6823** | 2408 | 0.0163 | 20.5631 | 5.0971e-13 | 2695 | **0.0149** | **18.8803** | 5.1064e-13 |

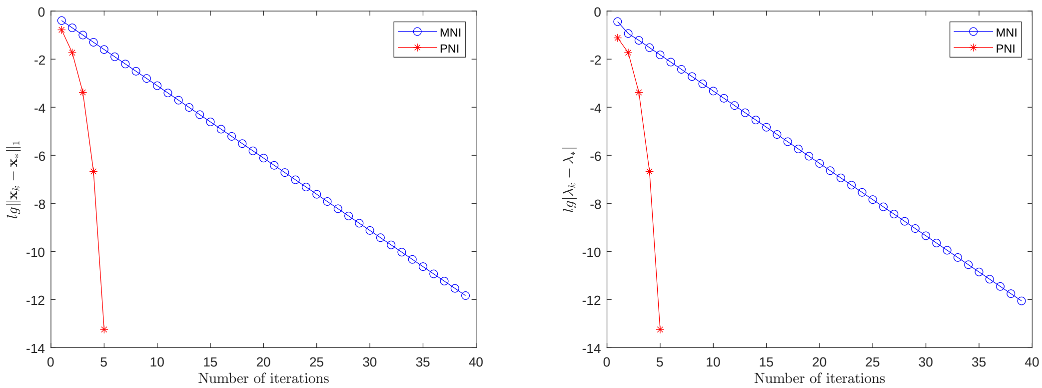


Figure 1: Convergence of $\|\mathbf{x}_k-\mathbf{x}^{(5)}\|_1$ and $|\lambda_k-\lambda^{(5)}|$ for Example 4.2.

**Example 4.3** ([7])**.** Consider a block diagonal tensor $\mathcal{A}\in\mathbb{R}_+^{[3,kn]}$ with

$$\mathcal{A}=diag(\mathcal{A}_1,\cdots,\mathcal{A}_k),$$

where $\mathcal{A}_i=rand(n,n,n)\in\mathbb{R}_+^{[3,n]}$ for $i=1,\cdots,k$.

For this example, we use a random initial vector. In Table 3, "Sign" denotes the number of times with $s_k=\max(\widehat{\mathbf{w}}_k)\min(\widehat{\mathbf{w}}_k)<0$, "IT" stands for the number of iterations to achieve convergence, "CPU" represents the time of iterations to achieve convergence, and "Err" refers to the residual error when the iteration is terminated.

One can see from Table 3 that, when $s_k<0$ occurs frequently, PNI is obviously superior to MNI in terms of IT and CPU. It can be verified that the modification of $\mathbf{x}_k$ works better than that of $\mathbf{w}_k$.

Table 3: Numerical results for Example 4.3.

| $m=3$ | MNI | | | | PNI | | |
|---|---|---|---|---|---|---|---|
| | Sign | CPU | IT | Err | CPU | IT | Err |
| $(k=10,\ n=2)$ | **38** | 0.0337 | 38 | 6.2712e-13 | **0.0264** | **28** | 5.5279e-13 |
| $(k=10,\ n=5)$ | **34** | 0.0490 | 36 | 6.6334e-13 | **0.0275** | **18** | 9.5580e-13 |
| $(k=10,\ n=10)$ | **35** | 0.1172 | 37 | 6.2742e-13 | **0.0484** | **13** | 2.1699e-13 |
| $(k=10,\ n=20)$ | 0 | 0.0980 | 11 | 1.7186e-13 | 0.0981 | 11 | 1.7186e-13 |
| $(k=20,\ n=2)$ | **38** | 0.0410 | 38 | 7.8179e-13 | **0.0249** | **20** | 5.5689e-13 |
| $(k=20,\ n=5)$ | **73** | 0.2078 | 85 | 6.9782e-13 | **0.0577** | **20** | 5.1300e-13 |
| $(k=20,\ n=10)$ | **997** | 9.2718 | 1000 | 1.7277 | **0.1473** | **15** | 6.1119e-13 |
| $(k=20,\ n=20)$ | 1 | 0.7804 | 11 | 5.7686e-14 | 0.7742 | 11 | 5.6574e-14 |

## 4.2 Numerical results for multilinear PageRank

We apply PNI to solve the multilinear PageRank problem, and compare PNI with the existing algorithms of the same type, which are Newton's method (NM) [6], Algorithm 4 (Alg4) in [16] and MNI. There are two ways to apply PNI to solve the multilinear PageRank problem: 1) Take $\lambda_k=1$ for all $k\geq 0$, which is a special case in Algorithm 4 of [16] when $\alpha=m-1$, $\gamma=\frac{1}{m-1}$. 2) Take $\lambda$ as unknown to solve, which can adjust $\lambda_k$ to make $\lambda_k\mathbf{I}-\mathbf{T}(\mathbf{x}_k)$ nonsingular as well as make the convergence of $\lambda_k$ have some influence on the convergence of $\mathbf{x}_k$. In numerical experiments, we use the latter. For MNI, $\lambda$ can be referred as an unknown eigenvalue to solve. For the sake of fairness, we semisymmetric tensors and use the function *ttv* of the package Tensor Toolbox 2.6 in all algorithms. In most cases, Algorithm 4 performs best when $\alpha=m-1$, $\gamma=\frac{1}{m-1}$, which we use in the following experiments. $\mathbf{x}_0=\mathbf{v}=\frac{1}{n}\mathbf{e}$ is used.

**Example 4.4.** This example is given by Test 4 of [12] (or see [16]). Let nodes set $\mathbb{V}=\{1,2,\cdots,n\}$, $\mathbb{P}$ be the subset of $\mathbb{V}$ for which arbitrary two nodes are pairwise-connected, and $\mathbb{D}$ be the subset with all dangling nodes in $\mathbb{V}$. Let $n_p$ $(n_p\geq 2)$ denote the number of nodes in the subset $\mathbb{P}$.

$$\mathcal{P}_{i_1,i_2,\cdots,i_m}=\begin{cases}\gamma_{i_1,i_2,\cdots,i_m}, & i_k\neq i_{k+1},\ i_k\in\mathbb{P},\ k=1,\cdots,m-1,\\ 0, & i_1=i_2 \text{ (or } i_1\in\mathbb{D}),\ i_k\neq i_{k+1},\ i_k\in\mathbb{P},\ k=2,\cdots,m-1,\\ \frac{1}{n}, & \text{else},\end{cases}$$

where $\gamma_{i_1,i_2,\cdots,i_m}\in(0,1)$, and then normalizing the entries with $\widehat{\mathcal{P}}_{i_1,i_2,\cdots,i_m}=\frac{\mathcal{P}_{i_1,i_2,\cdots,i_m}}{\sum_{i_1=1}^{n}\mathcal{P}_{i_1,i_2,\cdots,i_m}}$ generates a stochastic tensor $\widehat{\mathcal{P}}=(\widehat{\mathcal{P}}_{i_1,i_2,\cdots,i_m})$. It is noted that in the following numerical test, $\gamma_{i_1,i_2,\cdots,i_m}$ is taken in $(0,1)$ randomly and independently.

We take $\alpha$ as 0.7, 0.8, 0.9 and 0.99 respectively. For every value of $\alpha$, we let 1) $m=3$, $n=500$, $n_p=400,430,460,490$. 2) $m=4$, $n=100$, $n_p=60,70,80,90$. 3) $m=5$, $n=40$, $n_p=20,25,30,35$. 4) $m=6$, $n=20$, $n_p=10,12,15,18$. 5) $m=7$, $n=10$, $n_p=3,5,7,9$.

Table 4: Numerical results for Example 4.4 ($\alpha=0.7$).

| $n/m$ | $n_p$ | NM | | | Alg4 | | | | MNI | | | PNI | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CPU | IT | Err | CPU | IT | Err | Sign | CPU | IT | Err | CPU | IT | Err |
| 500/3 | 400 | 0.6472 | 4 | 1.7244e-14 | 0.3514 | 4 | 2.7671e-14 | 1 | 0.2947 | 4 | 5.6407e-15 | **0.2683** | **3** | 4.4362e-14 |
| | 430 | 0.2047 | **3** | 5.0355e-13 | 0.1846 | **3** | 4.9523e-13 | 1 | 0.2412 | 4 | 4.5305e-15 | **0.1709** | **3** | 5.0015e-14 |
| | 460 | 0.6186 | 4 | 8.1604e-14 | 0.8355 | 4 | 8.3628e-14 | 1 | 0.5761 | 4 | 5.2776e-14 | **0.3576** | **3** | 5.2195e-15 |
| | 490 | 0.3204 | 3 | 3.5807e-14 | 0.2841 | 3 | 5.6409e-14 | 1 | 0.3001 | 3 | 8.1189e-13 | **0.2803** | 3 | 7.3881e-14 |
| 100/4 | 60 | 0.3957 | 5 | 5.0313e-15 | 0.2736 | 5 | 6.4808e-15 | 0 | 0.2408 | **4** | 1.8541e-14 | **0.2349** | **4** | 1.9266e-14 |
| | 70 | 0.2094 | 5 | 4.4122e-15 | 0.2158 | 5 | 6.4038e-15 | 1 | 0.2463 | **4** | 2.8751e-13 | **0.1956** | **4** | 1.1532e-15 |
| | 80 | 0.4470 | **4** | 2.8773e-13 | 0.2400 | **4** | 3.0928e-13 | 2 | 0.2684 | 5 | 7.3326e-15 | **0.2269** | **4** | 3.6935e-15 |
| | 90 | 0.3385 | 4 | 7.1709e-15 | 0.2427 | 4 | 4.3459e-15 | 2 | 0.3960 | 5 | 7.3651e-16 | **0.2189** | **3** | 6.3417e-13 |
| 40/5 | 20 | 0.8993 | 15 | 2.9154e-13 | 0.9255 | 15 | 3.1712e-13 | 0 | 1.6794 | **13** | 4.1934e-13 | **0.8042** | **13** | 3.9662e-13 |
| | 25 | 0.9122 | 19 | 2.4610e-13 | 0.9061 | 19 | 2.7349e-13 | 0 | 1.0938 | **17** | 3.0868e-13 | **0.8751** | **17** | 3.2153e-13 |
| | 30 | 1.2530 | 18 | 4.5706e-13 | 0.8839 | 18 | 4.3236e-13 | 1 | 1.2644 | 18 | 3.9573e-13 | **0.8001** | **16** | 5.5904e-13 |
| | 35 | 0.6202 | 13 | 1.1201e-13 | 0.6181 | 13 | 1.2767e-13 | 2 | 0.8171 | 13 | 4.1457e-13 | **0.5934** | **11** | 3.7191e-13 |
| 20/6 | 10 | 0.6083 | 9 | 2.7551e-13 | 0.4578 | 9 | 2.8479e-13 | 0 | 0.5809 | **7** | 9.1609e-13 | **0.4003** | **7** | 9.4269e-13 |
| | 12 | 0.4511 | 11 | 1.8017e-13 | 0.4097 | 11 | 2.2064e-13 | 0 | 0.4216 | **9** | 6.6569e-13 | **0.3668** | **9** | 7.2902e-13 |
| | 15 | 0.4592 | 13 | 4.6858e-13 | 0.4599 | 13 | 3.3088e-13 | 1 | 0.8136 | **12** | 6.9427e-13 | **0.4486** | **12** | 2.2389e-13 |
| | 18 | 0.4366 | 11 | 7.6988e-14 | 0.3960 | 11 | 7.1399e-14 | 2 | 0.4593 | 11 | 4.4452e-13 | **0.3618** | **9** | 3.6460e-13 |
| 10/7 | 3 | 0.1545 | 4 | 3.0851e-13 | 0.1028 | 4 | 3.5120e-13 | 0 | 0.0629 | **3** | 2.2492e-13 | **0.0541** | **3** | 2.3753e-13 |
| | 5 | 0.2267 | 6 | 6.5897e-13 | 0.6946 | 6 | 6.9952e-13 | 0 | 0.0844 | **5** | 2.0384e-13 | **0.0702** | **5** | 2.2657e-13 |
| | 7 | 0.0995 | 9 | 1.3022e-13 | 0.1002 | 9 | 1.2061e-13 | 0 | 0.0887 | **8** | 1.6337e-13 | **0.8175** | **8** | 1.7156e-13 |
| | 9 | 0.1224 | 11 | 6.5355e-13 | 0.1155 | 11 | 6.0746e-13 | 1 | 0.1183 | 11 | 3.1534e-13 | **0.1031** | **10** | 8.6837e-13 |

Table 5: Numerical results for Example 4.4 ($\alpha=0.8$).

| $n/m$ | $n_p$ | NM | | | Alg4 | | | | MNI | | | PNI | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CPU | IT | Err | CPU | IT | Err | Sign | CPU | IT | Err | CPU | IT | Err |
| 500/3 | 400 | 0.2357 | 4 | 4.3338e-14 | 0.2243 | 4 | 2.8420e-14 | 2 | 0.2633 | 4 | 2.9480e-13 | **0.2164** | **3** | 4.7517e-14 |
| | 430 | 0.2377 | 4 | 3.0272e-14 | 0.2253 | 4 | 3.7441e-14 | 2 | 0.2509 | 4 | 5.3368e-13 | **0.2081** | **3** | 3.8014e-15 |
| | 460 | 0.2130 | 4 | 2.1669e-13 | 0.2431 | 4 | 2.6367e-13 | 2 | 0.2586 | 4 | 4.1432e-13 | **0.2050** | **3** | 4.8201e-14 |
| | 490 | 0.2257 | 4 | 5.1785e-14 | 0.2401 | 4 | 3.4346e-14 | 2 | 0.3252 | 4 | 7.1794e-15 | **0.2048** | **3** | 7.1517e-15 |
| 100/4 | 60 | 0.2316 | 5 | 4.8800e-15 | 0.2235 | 5 | 8.4995e-15 | 1 | 0.2247 | **4** | 5.5680e-13 | **0.2103** | **4** | 2.0876e-15 |
| | 70 | 0.2058 | 5 | 5.4440e-15 | 0.2063 | 5 | 1.7046e-14 | 2 | 0.1479 | 5 | 6.9818e-14 | **0.1978** | **4** | 2.3581e-14 |
| | 80 | 0.2196 | 5 | 4.1153e-15 | 0.2140 | 5 | 6.2601e-15 | 2 | 0.2649 | 5 | 4.9953e-13 | **0.2026** | **4** | 4.8375e-15 |
| | 90 | 0.2096 | 4 | 5.7501e-15 | 0.1953 | 4 | 4.7646e-15 | 3 | 0.3011 | 6 | 4.9757e-15 | **0.1737** | **3** | 7.8836e-16 |
| 40/5 | 20 | 0.9264 | 17 | 6.6409e-13 | 0.9247 | 17 | 6.9046e-13 | 0 | 1.0567 | **15** | 8.8107e-13 | **0.9026** | **15** | 8.6060e-13 |
| | 25 | 1.4476 | 25 | 5.8574e-13 | 1.4200 | 25 | 5.1569e-13 | 1 | 1.4423 | **23** | 6.3234e-13 | **1.3856** | **23** | 6.5065e-13 |
| | 30 | 1.3066 | 23 | 7.3933e-13 | 1.1491 | 23 | 7.8885e-13 | 2 | 1.2759 | 23 | 2.2807e-13 | **1.1023** | **21** | 4.3059e-13 |
| | 35 | 0.7241 | 14 | 2.8271e-13 | 0.7003 | 14 | 3.1831e-13 | 999 | 51.5446 | 1000 | 0.9900 | **0.6834** | **12** | 4.8511e-13 |
| 20/6 | 10 | 0.5127 | 10 | 6.3710e-14 | 0.4185 | 10 | 6.3852e-14 | 0 | 0.4163 | **8** | 2.8565e-13 | **0.3772** | **8** | 2.0312e-13 |
| | 12 | 0.4978 | 12 | 1.9225e-13 | 0.4624 | 12 | 1.2568e-13 | 0 | 0.4145 | **10** | 6.5783e-13 | **0.4128** | **10** | 6.3273e-13 |
| | 15 | 0.5807 | 15 | 2.4746e-13 | 0.5583 | 15 | 2.7431e-13 | 1 | 0.6003 | 15 | 2.6708e-13 | **0.5118** | **13** | 8.0639e-13 |
| | 18 | 0.4285 | 11 | 6.9018e-13 | 0.4966 | 11 | 6.7209e-13 | 999 | 38.1724 | 1000 | 0.9690 | **0.4107** | **10** | 6.4433e-13 |
| 10/7 | 3 | 0.0417 | 4 | 5.4159e-13 | 0.0326 | 4 | 5.1555e-13 | 0 | 0.0331 | **3** | 3.5465e-13 | **0.0312** | **3** | 3.5643e-13 |
| | 5 | 0.2621 | 7 | 2.3867e-14 | 0.0729 | 7 | 2.5050e-14 | 0 | 0.0496 | **5** | 5.0651e-13 | **0.0487** | **5** | 5.8686e-13 |
| | 7 | 0.0834 | 9 | 5.7485e-13 | 0.1697 | 9 | 5.7457e-13 | 0 | 0.1102 | **8** | 4.8479e-13 | **0.0718** | **8** | 4.2371e-13 |
| | 9 | 0.1086 | 12 | 4.3375e-13 | 0.1055 | 12 | 4.9641e-13 | 2 | 0.1007 | 12 | 4.1435e-13 | **0.9791** | **11** | 4.1035e-13 |

Tables 4-7 show that, with the increase of $m$, PNI performs better. We can see that PNI needs less iterations and time to achieve convergence in most cases.

Table 6: Numerical results for Example 4.4 ($\alpha=0.9$).

| $n/m$ | $n_p$ | NM | | | Alg4 | | | | MNI | | | PNI | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CPU | IT | Err | CPU | IT | Err | Sign | CPU | IT | Err | CPU | IT | Err |
| 500/3 | 400 | 0.2481 | 4 | 4.9086e-14 | 0.2166 | 4 | 6.1152e-14 | 3 | 0.3447 | 5 | 7.7444e-15 | **0.2108** | **3** | 3.7247e-14 |
| | 430 | 0.2369 | 4 | 7.1222e-14 | 0.2198 | 4 | 1.2081e-14 | 3 | 0.3407 | 5 | 2.2497e-14 | **0.2011** | **3** | 6.9849e-15 |
| | 460 | 0.2477 | 4 | 4.4862e-13 | 0.2485 | 4 | 4.2927e-13 | 3 | 0.5739 | 5 | 4.7165e-14 | **0.2346** | **3** | 6.5742e-15 |
| | 490 | 0.2315 | 4 | 6.7110e-14 | 0.2243 | 4 | 9.1008e-14 | 3 | 0.4117 | 5 | 5.3139e-15 | **0.2183** | **3** | 3.9283e-14 |
| 100/4 | 60 | 0.2819 | 6 | 6.8929e-15 | 0.2602 | 6 | 4.5987e-15 | 2 | 0.3075 | 6 | 1.4966e-15 | **0.2565** | **5** | 5.8257e-16 |
| | 70 | 0.2087 | 5 | 6.8158e-15 | 0.2161 | 5 | 9.1412e-15 | 3 | 0.2785 | 6 | 5.6047e-13 | **0.2012** | **4** | 8.7906e-16 |
| | 80 | 0.3407 | 5 | 5.1934e-15 | 0.2497 | 5 | 1.3848e-14 | 4 | 0.3661 | 7 | 4.7486e-15 | **0.2413** | **4** | 4.3848e-15 |
| | 90 | 0.2074 | 4 | 7.4388e-15 | 0.2883 | 4 | 4.8923e-15 | 999 | 51.4835 | 1000 | 0.9579 | **0.1878** | **3** | 1.6024e-15 |
| 40/5 | 20 | 1.2653 | 21 | 5.7290e-13 | 1.0943 | 21 | 5.8944e-13 | 0 | 1.0472 | **19** | 6.3618e-13 | **1.0021** | **19** | 6.2037e-13 |
| | 25 | 2.2759 | 45 | 8.5303e-13 | 2.2488 | 45 | 7.9272e-13 | 1 | 2.2813 | 45 | 3.2099e-13 | **2.2154** | **42** | 7.9820e-13 |
| | 30 | 1.9445 | 37 | 7.4164e-13 | 1.8233 | 37 | 7.2687e-13 | 3 | 1.7877 | **35** | 6.7307e-13 | **1.7006** | **35** | 3.5341e-13 |
| | 35 | 0.8126 | 16 | 1.8451e-13 | 0.8110 | 16 | 1.2172e-13 | 999 | 49.3824 | 1000 | 0.9613 | **0.7992** | **14** | 5.3751e-13 |
| 20/6 | 10 | 0.4487 | 10 | 3.0717e-13 | 0.3911 | 10 | 3.3625e-13 | 0 | 0.4382 | **9** | 7.1096e-14 | **0.3785** | **9** | 6.9136e-14 |
| | 12 | 0.4826 | 13 | 2.8910e-13 | 0.4801 | 13 | 2.3512e-13 | 0 | 0.4593 | **11** | 8.2279e-13 | **0.4543** | **11** | 8.0405e-13 |
| | 15 | 0.8022 | 17 | 5.8993e-13 | 0.6634 | 17 | 6.3182e-13 | 2 | 0.7093 | 17 | 7.2094e-13 | **0.6521** | **16** | 5.4840e-13 |
| | 18 | 0.4627 | 12 | 5.1756e-13 | 0.4588 | 12 | 5.7677e-13 | 999 | 36.7346 | 1000 | 0.9352 | **0.4502** | **11** | 2.1739e-13 |
| 10/7 | 3 | 0.0514 | 4 | 7.5073e-13 | 0.04285 | 4 | 7.2938e-13 | 0 | 0.0362 | **3** | 5.8733e-13 | **0.0358** | **3** | 5.8235e-13 |
| | 5 | 0.0624 | 7 | 4.9988e-14 | 0.0832 | 7 | 4.3745e-14 | 0 | 0.0593 | **6** | 2.5081e-14 | **0.0547** | **6** | 2.7485e-14 |
| | 7 | 0.0835 | 10 | 1.3999e-13 | 0.0822 | 10 | 1.9317e-13 | 0 | 0.0947 | **9** | 1.4277e-13 | **0.0781** | **9** | 1.0422e-13 |
| | 9 | 0.1234 | 13 | 3.1576e-13 | 0.1238 | 13 | 3.3614e-13 | 2 | 0.1259 | 13 | 3.0589e-13 | **0.1210** | **12** | 4.8434e-13 |

Table 7: Numerical results for Example 4.4 ($\alpha=0.99$).

| $n/m$ | $n_p$ | NM | | | Alg4 | | | | MNI | | | PNI | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CPU | IT | Err | CPU | IT | Err | Sign | CPU | IT | Err | CPU | IT | Err |
| 500/3 | 400 | 0.2604 | 4 | 6.8900e-14 | 0.2264 | 4 | 3.0589e-14 | 6 | 0.5824 | 8 | 4.2025e-14 | **0.2217** | **3** | 7.0874e-14 |
| | 430 | 0.2349 | 4 | 5.5731e-14 | 0.2364 | 4 | 4.4370e-14 | 6 | 0.4825 | 8 | 3.0888e-14 | **0.2177** | **3** | 1.7757e-14 |
| | 460 | 0.2489 | 4 | 8.7339e-13 | 0.3457 | 4 | 9.1452e-13 | 6 | 0.5753 | 8 | 6.6158e-14 | **0.2377** | **3** | 3.7982e-14 |
| | 490 | 0.2271 | 4 | 5.8202e-14 | 0.2200 | 4 | 8.3686e-14 | 6 | 0.5328 | 8 | 2.8099e-14 | **0.2068** | **3** | 2.0607e-14 |
| 100/4 | 60 | 0.2632 | 6 | 6.0908e-15 | 0.2596 | 6 | 6.8046e-15 | 4 | 0.3815 | 8 | 2.5470e-15 | **0.2531** | **5** | 3.8951e-13 |
| | 70 | 0.3272 | 5 | 1.8325e-13 | 0.2628 | 5 | 2.0626e-13 | 6 | 0.3991 | 8 | 3.1869e-15 | **0.2336** | **4** | 1.6000e-15 |
| | 80 | 0.2239 | 5 | 8.0601e-15 | 0.2356 | 5 | 2.2335e-15 | 7 | 0.4621 | 9 | 7.2197e-13 | **0.2201** | **4** | 1.2593e-15 |
| | 90 | 0.2113 | **4** | 9.6949e-15 | 0.2106 | **4** | 1.9221e-14 | 999 | 47.5993 | 1000 | 0.9890 | **0.2025** | **4** | 2.6556e-15 |
| 40/5 | 20 | 1.7443 | 28 | 4.6336e-13 | 1.7230 | 28 | 4.4018e-13 | 0 | 1.3626 | **25** | 9.3578e-13 | **1.3258** | **25** | 9.4540e-13 |
| | 25 | 49.8389 | 1001 | 0.1570 | 50.8897 | 1001 | 0.1602 | 320 | 49.9114 | 1000 | 0.0714 | 49.9223 | 1000 | 0.0373 |
| | 30 | 14.5388 | 225 | 9.3387e-13 | 13.1873 | 225 | 9.0633e-13 | 999 | 51.8649 | 1000 | 0.9580 | **12.4918** | **215** | 7.2669e-13 |
| | 35 | 0.8443 | 17 | 8.0926e-13 | 0.8952 | 17 | 7.6713e-13 | 999 | 48.1732 | 1000 | 0.9870 | **0.8165** | **16** | 4.9449e-13 |
| 20/6 | 10 | 0.5598 | 11 | 9.6563e-14 | 0.5720 | 11 | 9.4164e-14 | 0 | 0.4126 | **9** | 3.2416e-13 | **0.3741** | **9** | 3.3510e-13 |
| | 12 | 0.6003 | 14 | 3.6997e-13 | 0.6271 | 14 | 3.4887e-13 | 0 | 0.5973 | **13** | 1.8204e-13 | **0.5214** | **13** | 2.3300e-13 |
| | 15 | 0.9354 | 21 | 5.0941e-13 | 0.9317 | 21 | 5.8318e-13 | 2 | 1.0572 | 21 | 7.4267e-13 | **0.9081** | **20** | 7.2569e-13 |
| | 18 | 0.4812 | 13 | 5.5771e-13 | 0.4725 | 13 | 4.9614e-13 | 999 | 41.6059 | 1000 | 0.9410 | **0.4659** | **12** | 3.0634e-13 |
| 10/7 | 3 | 0.0510 | 5 | 3.7194e-15 | 0.0405 | 5 | 3.9020e-15 | 0 | 0.0356 | **3** | 8.1952e-13 | **0.0352** | **3** | 7.8977e-13 |
| | 5 | 0.0936 | 7 | 1.8216e-13 | 0.0743 | 7 | 1.6659e-13 | 0 | 0.0741 | **6** | 5.6567e-14 | **0.0710** | **6** | 5.5355e-14 |
| | 7 | 0.0780 | 10 | 4.1480e-13 | 0.0813 | 10 | 4.6154e-13 | 0 | 0.0804 | **9** | 3.3739e-13 | **0.0782** | **9** | 3.4387e-13 |
| | 9 | 0.1147 | 14 | 2.5992e-13 | 0.1102 | 14 | 2.8780e-13 | 999 | 8.9857 | 1000 | 0.9350 | **0.1078** | **13** | 2.1159e-13 |

## 5 Concluding remarks

In order to compute the nonnegative Z-eigenpairs of nonnegative tensors efficiently, we propose a projected Newton iteration, which maintains a local quadratic convergence. The numerical experiment results demonstrate the effectiveness and superiority of the proposed approach. Furthermore, the proposed algorithm can be employed to solve the multilinear PageRank, which appears to be more effective comparing with the existing algorithms in handling the multilinear PageRank.

## Acknowledgments

This research was supported by the National Natural Science Foundations of China (12071159, 11671158, U1811464), the Guangdong Basic and Applied Basic Research Foundations (2020B1515310013, 2020A1515110967) and the NSF-DMS 1854638 of the United States. The first author would like to thank the hospitality from the Department of Mathematics, Southern Illinois University Carbondale during her visit from October 2019 to June 2020. The authors would like to thank anonymous referees for their very helpful comments and suggestions.

**References**

[1] B. W. Bader and T. G. Kolda, Algorithm 862: Matlab tensor classes for fast algorithm prototyping, ACM Trans. Math. Softw., 32(2006), 635-653.
[2] L. Bloy and R. Verma, On computing the underlying fiber directions from the diffusionn orientation distribution function, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Berlin, 2008, pp. 1-8.
[3] K. C. Chang, K. J. Pearson and T. Zhang, Some variational principles for Z-eigenvalues of nonnegative tensors, Linear Algebra Appl., 438(2013), 4166-4182.
[4] K. C. Chang and T. Zhang, On the uniqueness and non-uniqueness of the positive Z-eigenvector for transition probability tensors, J. Math. Anal. Appl., 408(2013), 525-540.
[5] Y. N. Chen, Y. H. Dai, D. R. Han and W. Y. Sun, Positive semidefinite generalized diffusion tensor imaging via quadratic semidefinite programming, SIAM J. Imaging Sci., 6(2013), 1531-1552.
[6] D. F. Gleich, L. H. Lim and Y. Y. Yu, Multilinear pagerank, SIAM J. Matrix Anal. Appl., 36(2015), 1507-1541.
[7] C. H. Guo, W. W. Lin and C. S. Liu, A modified Newton iteration for finding nonnegative Z-eigenpairs of a nonnegative tensor, Numer. Algorithms, 80(2019), 595-616.
[8] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM Press, Philadelphia, 1995.
[9] E. Kofidis and P. A. Regalia, On the best rank-1 approximation of higher-order supersymmetric tensors, SIAM J. Matrix Anal. Appl., 23(2002), 863-884.
[10] T. G. Kolda and J. R. Mayo, Shifted power method for computing tensor eigenpairs, SIAM J. Matrix Anal. Appl., 32(2011), 1095-1124.

[11] T. G. Kolda and J. R. Mayo, An adaptive shifted power method for computing generalized tensor eigenpairs, SIAM J. Matrix Anal. Appl., 35(2014), 1563-1581.

[12] W. Li, D. D. Liu, M. K. Ng and S. W. Vong, The uniqueness of multilinear pagerank vectors, Numer. Linear Algebra Appl., 24(2017), e2107.

[13] W. Li, D. D. Liu, S. W. Vong and M. Q. Xiao, Multilinear pagerank: Uniqueness, error bound and perturbation analysis, Appl. Numer. Math., 156(2020), 584-607.

[14] W. Li and M. K. Ng, On the limiting probability distribution of a transition probability tensor, Linear Multilinear Algebra, 62(2014), 362-385.

[15] L. H. Lim, Singular values and eigenvalues of tensors: A variational approach, Proc. IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 1(2005), 129-132.

[16] D. D. Liu, W. Li and S. W. Vong, Relaxation methods for solving the tensor equation arising from the higher-order markov chains, Numer. Linear Algebra Appl., 26(2019), e2260.

[17] Q. Ni and L. Q. Qi, A quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map, J. Glob. Optim., 61(2015), 627-641.

[18] L. Q. Qi, Eigenvalues of a real supersymmetric tensor, J. Symb. Comput., 40(2005), 1302-1324.

[19] L. Q. Qi, Y. J. Wang and E. X. Wu, D-eigenvalues of diffusion kurtosis tensors, J. Comput. Appl. Math., 221(2008), 150-157.

[20] R. J. Zhao, B. Zheng, M. L. Liang and Y. Y. Xu, A locally and cubically convergent algorithm for computing Z-eigenpairs of symmetric tensors, Numer. Linear Algebra Appl., 27(2020), e2284.