**Journal of Mathematical Modeling**
Vol. -, No. -, 2021, pp. -

JMM

# A fast and efficient Newton-Shultz-type iterative method for computing inverse and Moore-Penrose inverse of tensors

**Eisa Khosravi Dehdezi*** **, Saeed Karimi**

*Department of Mathematics, Persian Gulf University, Bushehr, Iran*

*Email(s): esakhosravidehdezi@gmail.com, karimi@pgu.ac.ir*

**Abstract.** A fast and efficient Newton-Shultz-type iterative method is presented to compute the inverse of an invertible tensor. Analysis of the convergence error shows that the proposed method has the sixth order convergence. It is shown that the proposed algorithm can be used for finding the Moore-Penrose inverse of tensors. Computational complexities of the algorithm is presented to support the theoretical aspects of the paper. Using the new method, we obtain a new preconditioner to solve the multilinear system $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$. The effectiveness and accuracy of this method are re-verified by several numerical examples. Finally, some conclusions are given.

*Keywords*: Tensor, iterative methods, Moore-Penrose inverse, outer inverse, Einstein product.
*AMS Subject Classification 2010*: 15A10, 15A69, 15A72, 15A99, 65F10.

## 1 Introduction

In recent years, tensors or hypermatrices have been applied in many types of research and application areas such as data analysis, psychometrics, chemometrics, image processing, graph theory, Markov chains, hypergraphs, etc. [35]. Tensor equations (or multilinear systems), with the Einstein product, have been discussed in [9,14], which have many applications in continuum physics, engineering, isotropic and anisotropic elastic models [26]. Wang and Xu presented some iterative methods for solving several kinds of tensor equations in [42]. Huang and Ma, in [20], proposed the Krylov subspace methods to solve a class of tensor equations. In [18], Huang and Ma presented an iterative algorithm to solve the generalized Sylvester tensor equations. They

*Corresponding author.
Received: 23 February 2021/ Revised: 12 March 2021/ Accepted: 15 March 2021
DOI: 10.22124/jmm.2021.19005.1627

 

also, in [19], proposed the global least-squares methods based on tensor form to solve a class of generalized Sylvester tensor equations. In [7], F. P. A. Beik and S. Ahmadi-Asl applied the steepest descent based iterative method for solving tensor equations. In [24], Khosravi Dehdezi and Karimi proposed the extended conjugate gradient squared and conjugate residual squared methods for solving the generalized coupled Sylvester tensor equations

$$\sum_{j=1}^{n} \mathcal{X}_j \times_1 A_{ij1} \times_2 A_{ij2} \times \cdots \times_d A_{ijd} = \mathcal{C}_i, \quad i = 1, 2, \ldots, n, \tag{1}$$

where the matrices $A_{ijl} \in \mathbb{C}^{n_{ijl} \times n_{ijl}}$ , $l = 1, 2, \ldots, d$, the tensors $\mathcal{C}_i \in \mathbb{C}^{n_{i1} \times \cdots \times n_{id}}$, $i = 1, 2, \ldots, n$ and $\mathcal{X}_j \in \mathbb{C}^{n_{j1} \times \cdots \times n_{jd}}$, $j = 1, 2, \ldots, n$ are known and unknown, respectively and also $\times_j$, $j = 1, 2, \ldots, n$ denote the $j$-mode product (see [25] for more details about $j$-mode product). Stanimirovi et al., in [38], proposed some basic properties of the range and null space of tensors with respect to Einstein tensor product. Also, an efficient definition of the tensor rank is introduced in [38]. This definition is called reshaping rank and it is related to the matrix rank. Using these properties, conditions for the existence and representations of outer inverses of tensors are considered. These representations apply to complex tensors and are based on finding solutions to certain matrix equations and simple Einstein product with appropriate tensors. Algorithms arising from the introduced representations are developed. In addition, results related to the (b,c)-inverses on semigroups, originated in [16], in a specific semigroup of tensors with a binary associative operation defined as the Einstein tensor product, was considered in [38]. One may refer to [3–6, 10, 12, 15, 21–23, 28–34, 39, 40, 43] to see the other topics in tensor.

The importance of tensor inversion for solving multilinear systems (see [9]), computing approximation of Moore-Penrose and outer inverses of tensors, motivated the authors to present a fast and efficient iterative algorithm based on the Newton-Shultz method and divided differences for computing the approximation of the inverse of tensors. In the following, some definitions and propositions which will be used later are collected.

Throughout the paper, the following notations are used. Tensors are written in calligraphic letters, e.g., $\mathcal{A}$. Let $N, I_j, 1 \leq j \leq N$ be the positive integers. An order $N$ tensor $\mathcal{A} = (a_{i_1 \cdots i_N})$, $(1 \leq i_j \leq I_j,\ j = 1, 2, \ldots, N)$ is a multidimensional array with $I = I_1 \times \cdots \times I_N$ entries [25], where $\times$ denotes the "multiplication" operation. $\mathcal{O}$ with all zero entries denotes the zero tensor. With this definition of tensors, matrices are tensors of order two where signified by capital letters, e.g., $A$. As usual, $\mathbb{R}$ and $\mathbb{C}$ denote the real and complex numbers field, respectively. Finally $\mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\mathbb{C}^{I_1 \times \cdots \times I_N}$ are the set of order $N$, dimension $I_1 \times I_2 \times \cdots \times I_N$ tensors over $\mathbb{R}$ and $\mathbb{C}$, respectively.

**Definition 1.** *[42] The linear transformation $\Phi_{IJ} : \mathbb{R}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_N} \to \mathbb{R}^{I \times J}$ with $\Phi_{IJ}(\mathcal{A}) = A$ $(I = I_1 \times \cdots \times I_N, J = J_1 \times \cdots \times J_N)$ is defined component-wise as*

$$(\mathcal{A})_{i_1 \cdots i_N j_1 \cdots j_N} \to (A)_{st},$$

*where $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_N}, A \in \mathbb{R}^{I \times J}, s = i_N + \sum_{p=1}^{N-1} \left((i_p - 1) \prod_{q=p+1}^{N} I_q\right)$ and $t = j_N + \sum_{p=1}^{N-1} \left((j_p - 1) \prod_{q=p+1}^{N} J_q\right)$.*

**Note.** When $I = J$, for simplicity, we write $\Phi(\mathcal{A})$ instead of $\Phi_{II}(\mathcal{A})$.

**Definition 2.** *[39] Let $N, M, I_i, J_i, K_j, 1 \leq i \leq N, 1 \leq j \leq M$ be the positive integers, $\mathcal{A} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_N}$ and $\mathcal{B} \in \mathbb{C}^{J_1 \times \cdots \times J_N \times K_1 \times \cdots \times K_M}$. The Einstein product of $\mathcal{A}$ and $\mathcal{B}$ is defined by operation $*_N$ via*

$$(\mathcal{A} *_N \mathcal{B})_{i_1 \cdots i_N k_1 \cdots k_M} = \sum_{j_N=1}^{J_N} \cdots \sum_{j_1=1}^{J_1} a_{i_1 \cdots i_N j_1 \cdots j_N} b_{j_1 \cdots j_N k_1 \cdots k_M}.$$

*Thus $\mathcal{A} *_N \mathcal{B} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times K_1 \times \cdots \times K_M}$ and the associative law of this tensor product holds.*

For $\mathcal{A} = (a_{i_1 \cdots i_N j_1 \cdots j_M}) \in \mathbb{C}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$, let $\mathcal{B} = (b_{j_1 \cdots j_M i_1 \cdots i_N}) \in \mathbb{C}^{J_1 \times \cdots \times J_M \times I_1 \times \cdots \times I_N}$ be the conjugate transpose of $A$, where $b_{j_1 \cdots j_M i_1 \cdots i_N} = \bar{a}_{j_1 \cdots j_M i_1 \cdots i_N}$. The tensor $\mathcal{B}$ is denoted by $\mathcal{A}^*$. When $b_{i_1 \cdots i_M j_1 \cdots j_N} = a_{j_1 \cdots j_M i_1 \cdots i_N}$, $\mathcal{B}$ is called the transpose of $\mathcal{A}$, denoted by $\mathcal{A}^T$. Let $\mathcal{A} = (a_{i_1 \cdots i_N j_1 \cdots j_N}) \in \mathbb{C}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$. The trace of $\mathcal{A}$ is defined by

$$tr(\mathcal{A}) = \sum_{i_n=1}^{I_N} \cdots \sum_{i_1=1}^{I_1} a_{i_1 \cdots i_N i_1 \cdots i_N}. \tag{2}$$

Inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$ is defined by

$$< \mathcal{X}, \mathcal{Y} >= tr(\mathcal{Y}^* *_N \mathcal{X}) = \sum_{j_M=1}^{J_M} \cdots \sum_{j_1=1}^{J_1} \sum_{i_N=1}^{I_N} \cdots \sum_{i_1=1}^{I_1} x_{i_1 \cdots i_N j_1 \cdots j_M} \bar{y}_{j_1 \cdots j_M i_1 \cdots i_N},$$

so the tensor norm induced by this inner product is

$$\|\mathcal{X}\| = \sqrt{< \mathcal{X}, \mathcal{X} >} = \sqrt{\sum_{j_M=1}^{J_M} \cdots \sum_{j_1=1}^{J_1} \sum_{i_N=1}^{I_N} \cdots \sum_{i_1=1}^{I_1} |x_{i_1 \cdots i_N j_1 \cdots j_M}|^2},$$

which is the tensor Frobenius norm. Let $\mathcal{A} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$, then $\mathcal{A}^{i+1} = \mathcal{A} *_N \mathcal{A}^i$, $i = 1, 2, \ldots$.

**Definition 3.** *[9] $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$ is said diagonal tensor if $a_{i_1 \cdots i_N j_1 \cdots j_N} = 0$ for $i_l \neq j_l$, $l = 1, \cdots, N$. A diagonal tensor $\mathcal{I} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$ is identity if $i_{i_1 \cdots i_N j_1 \cdots j_N} = \prod_{k=1}^{N} \delta_{i_k j_k}$, where*

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

$\mathcal{A}$ is orthogonal if $\mathcal{A}^T *_N \mathcal{A} = \mathcal{I}$. The singular vallue decomposition(SVD) of tensor $\mathcal{A}$ is the form

$$\mathcal{A} = \mathcal{U} *_N \mathcal{D} *_N \mathcal{V}^T,$$

where $\mathcal{U}, \mathcal{V} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$ are orthogonal tensors and $\mathcal{D} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$ is diagonal tensor with entries $\sigma_{i_1 \cdots i_n i_1 \cdots i_n}$, called the singular values of $\mathcal{A}$. $\mathcal{B}, \mathcal{C} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$ are said left and right inverse of $\mathcal{A}$ with respect to $*_N$, respectively, if $\mathcal{B} *_N \mathcal{A} = \mathcal{I}$ and $\mathcal{A} *_N \mathcal{C} = \mathcal{I}$. If $\mathcal{B} = \mathcal{C}$, we say that $\mathcal{A}$ is invertible and $\mathcal{A}^{-1} = \mathcal{B}$.

**Definition 4.** *[13] Let $\mathcal{A} \in \mathbb{C}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$. $\lambda \in \mathbb{C}$ is said an eigenvalue of $\mathcal{A}$ if there exists a nonzero tensor $\mathcal{X} \in \mathbb{C}^{I_1\times\cdots\times I_N}$, called an eigentensor of $\mathcal{A}$ corresponding to $\lambda$, such that*

$$\mathcal{A} *_N \mathcal{X} = \lambda\mathcal{X}.$$

When $N = 1$, Definition 4 reduces to the matrix eigenvalue.

**Definition 5.** *[13] $\mathcal{A} = (a_{i_1\cdots i_N j_1\cdots j_N}) \in \mathbb{C}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$ is called a Toeplitz tensor if*

$$a_{i_1\cdots i_N j_1\cdots j_N} = g_{i_1-j_1,\cdots,i_N-j_N}, \qquad 1\le i_s,j_s\le I_s, \qquad 1\le s\le N,$$

*where $\mathcal{G} = (g_{k_1\cdots k_N}) \in \mathbb{C}^{(2I_1-1)\times\cdots\times(2I_N-1)}$ for $1 - I_s \le k_s \le I_s - 1$ and $1 \le s \le N$. We denote $\mathcal{A} := toep(\mathcal{G})$.*

**Definition 6.** *An iterative method is said to be convergent of $i$-th order, if there exist $\beta \in \mathbb{R}^+$ such that*

$$\|\mathcal{E}_{j+1}\| \le \beta \|\mathcal{E}_j\|^i, \tag{3}$$

*where $\mathcal{E}_j$ is the error obtained in the $j$-th step of iterative method.*

In 1933, Shultz and Hotelling provided the following iterative method, called the Shultz or Newton-Shultz and denoted by NS, for computing the approxiamte inverse of an invertible complex matrix $A$

$$X_{j+1} = X_j(I + E_j),\ E_j = I - AX_j,\ j = 0, 1, 2, \ldots, \tag{4}$$

where $I$ is the identity matrix with the same order as $A$.

By using (4), the NS iterative method for computing the approximate inverse of invertible tensor $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$ can be derived as follows:

$$\mathcal{X}_{j+1} = \mathcal{X}_j *_N (\mathcal{I} + \mathcal{E}_j),\ \mathcal{E}_j = \mathcal{I} - \mathcal{A} *_N \mathcal{X}_j,\ j = 0, 1, 2, \ldots, \tag{5}$$

where $\mathcal{I}$ is the identity tensor with the same order and dimension as $\mathcal{A}$. The higher-order iterative methods can be considered as follows:

$$\mathcal{X}_{j+1} = \mathcal{X}_j + \mathcal{X}_j *_N \mathcal{E}_j + \cdots + \mathcal{X}_j *_N \mathcal{E}_j^{p-1}. \tag{6}$$

The hyperpower iteration of order $p$ (6) for computing approxamate inverse of invertible tensor, requires $p$ tensor-tensor multiplications

**Proposition 1.** *Let $\mathcal{A}$ be an invertible tensor. Then the iterative method (5) is convergent to the inverse of $\mathcal{A}$ if $\|\mathcal{E}_0\| = \|\mathcal{I} - \mathcal{A} *_N \mathcal{X}_0\| \le q < 1$, and*

$$\left\|\mathcal{X}_j - \mathcal{A}^{-1}\right\| \le \frac{\|\mathcal{X}_0\|}{1-q}\|\mathcal{I} - \mathcal{A} *_N \mathcal{X}_j\| \le \frac{\|\mathcal{X}_0\|}{1-q}q^{2^j}.$$

*Proof.* By considering the definition $\mathcal{E}_j$ and the associative property of Einstein product, we have

$$\begin{aligned}\mathcal{E}_j &= \mathcal{I} - \mathcal{A} *_N \mathcal{X}_j = \mathcal{I} - \mathcal{A} *_N \mathcal{X}_{j-1} *_N (\mathcal{I} + \mathcal{E}_{j-1}) = \mathcal{I} - \mathcal{A} *_N \mathcal{X}_{j-1} - \mathcal{A} *_N \mathcal{X}_{j-1} *_N \mathcal{E}_{j-1} \\ &= \mathcal{E}_{j-1} - \mathcal{A} *_N \mathcal{X}_{j-1} *_N \mathcal{E}_{j-1} = (\mathcal{I} - \mathcal{A} *_N \mathcal{X}_{j-1}) *_N \mathcal{E}_{j-1} = \mathcal{E}_{j-1}^2.\end{aligned}$$

If we continue this process, we get $\mathcal{E}_j = \mathcal{E}_0^{2^j}$, $j = 1, 2, \ldots$. Therefore, we have $\|\mathcal{E}_j\| \to 0$ as $j \to \infty$ with the assumption $\|\mathcal{E}_0\| < 1$. So we conclude that $\mathcal{X}_j \to \mathcal{A}^{-1}$ as $j \to \infty$. On the other hand, $\|\mathcal{E}_0\| \leq q < 1$ results in $\left\|\mathcal{A}^{-1}\right\| \leq \dfrac{\|\mathcal{X}_0\|}{1-q}$ and we get

$$\left\|\mathcal{X}_j - \mathcal{A}^{-1}\right\| \leq \left\|\mathcal{A}^{-1}\right\| \|\mathcal{I} - \mathcal{A} *_N \mathcal{X}_j\| = \left\|\mathcal{A}^{-1}\right\| \|\mathcal{E}_j\| \leq \frac{\|\mathcal{X}_0\|}{1-q} \|\mathcal{E}_0\|^{2^j} \leq \frac{\|\mathcal{X}_0\|}{1-q} q^{2^j},$$

which completes the proof. □

Several iterative methods have been proposed based on Eq. (4) to compute the approximate inverse of the invertible matrix $A$ that are known as the Newton-Shultz-type methods. For example Frontini and Sormani, from now on denoted by FS, [17] and Li and Li, from now on denoted by LL, [27], proposed the following methods respectively:

$$X_{j+1} = \tfrac{1}{4} X_j(13I - AX_j(15I - AX_j(7I - AX_j))), \quad \text{(FS)}$$

$$X_{j+1} = X_j(4I - 6AX_j + 4(AX_j)^2 - (AX_j)^3), \quad \text{(LL)}$$

which have three and four convergence order. In the numerical examples, we compare the new method with the three methods $NS, FS$ and LL in the tensor form.

In this paper, we propose a fast, stable, and efficient iterative method based on the Newton-Shultz method and divided differences which is called the FNS algorithm for computing the approximate inverse of the invertible tensor $\mathcal{A}$.

The outline of this paper is as follows. In the next section, we propose the FNS method. The FNS algorithm is the six*th*-order convergent and uses only five tensor multiplications per iteration. Some propositions on the convergence analysis are also expressed and proved in this section. In addition, we obtain an error bound for a perturbed solution at every iteration. To verify the theoretical aspects of the paper, the computational complexities of the new algorithm is presented in Section 3. Using the FNS algorithm, the Moore-Penrose inverse of tensors and its convergence analysis are expressed in Section 4. In Section 5, we use the proposed algorithm to obtain a preconditioner for solving the multilinear system $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$. In Section 6, some numerical examples are given to illustrate the efficiency and superiority of FNS. Finally, some conclusions are shown in Section 7.

## 2 The Newton-Shultz-type iterative method for tensors

Assume that the sufficiently differentiable function $f$ has a simple root $\alpha$ in the open interval $D \subseteq \mathbb{R}$, and D contains $x_0$ as an initial approximation of $\alpha$. We propose a fast Newton-type

method with the sixth order convergence for computing $\alpha$. For an initial approximation $x_0$ close enough to $\alpha$, a fast Newton-type method for computing the root of $f(x)$ is as follows.

**Algorithm 1.** *A fast Newton-type algorithm*
***Input*** $x_0, f(x)$
***Begin***

1. *For* $j = 0, 1, 2, \ldots$ *until convergence Do:*
2. $u_j = x_j - f(x_j)/f'(x_j)$.
3. $v_j = u_j - f(u_j)/f'(u_j)$.
4. $x_{j+1} = v_j - \big(f[v_j, u_j]\big)^{-1} f(v_j)$.
5. *EndDo.*

***End***

In the above algorithm

$$f[x_j, x_i] = \frac{f(x_j) - f(x_i)}{x_j - x_i},$$

is the two-point divided difference. By taking $f(x) = \frac{1}{x} - a$, $q_j = ax_j$ and $p_j = q_j(2 - q_j)$, some steps of Algorithm 1 can be summarized as follows:

$$f'(x_j) = -\frac{1}{x_j^2},\ f[x_j, x_i] = -\frac{1}{x_j x_i},\quad x_{j+1} = x_j(2 - q_j)(3 - p_j(3 - p_j)).$$

The above summary has an important role in the following FNS algorithm.

**Algorithm 2.** *FNS algorithm*
***Input*** $\mathcal{A}, \mathcal{X}_0$
***Begin***

1. *For* $j = 0, 1, 2, \ldots$ *until convergence Do:*
2. $\mathcal{Q}_j = \mathcal{A} *_N \mathcal{X}_j,\ \mathcal{P}_j = \mathcal{Q}_j *_N (2\mathcal{I} - \mathcal{Q}_j)$.
3. $\mathcal{X}_{j+1} = \mathcal{X}_j *_N (2\mathcal{I} - \mathcal{Q}_j) *_N (3\mathcal{I} - \mathcal{P}_j *_N (3\mathcal{I} - \mathcal{P}_j))$.
4. *EndDo.*

***End***

For the sake of the simplicity, from now on, if no other special illustration, we refrain from writing $*_N$ in Einstein product of tensors.

**Proposition 2.** *Let* $\mathcal{X}_{j+1}$ *be the approximate inverse of* $\mathcal{A}$ *obtained by FNS algorithm and let* $\mathcal{E}_{j+1} = I - \mathcal{A}\mathcal{X}_{j+1}$. *If* $\|\mathcal{E}_0\| < 1$ *then* $\|\mathcal{E}_{j+1}\| \leq \|\mathcal{E}_j\|^6$.

*Proof.* According to definitions of $\mathcal{Q}_j$, $\mathcal{P}_j$ and $\mathcal{E}_j$, we have $\mathcal{Q}_j = \mathcal{I} - \mathcal{E}_j$ and $\mathcal{P}_j = (\mathcal{I} - \mathcal{E}_j)(\mathcal{I} + \mathcal{E}_j)$. By using FNS algorithm, we have

$$\begin{aligned}
\mathcal{X}_{j+1} &= \mathcal{X}_j(\mathcal{I} + \mathcal{E}_j)(3\mathcal{I} - (\mathcal{I} - \mathcal{E}_j)(\mathcal{I} + \mathcal{E}_j)(3\mathcal{I} - (\mathcal{I} - \mathcal{E}_j)(\mathcal{I} + \mathcal{E}_j))) \\
&= \mathcal{X}_j(\mathcal{I} + \mathcal{E}_j)(3\mathcal{I} - (\mathcal{I} - \mathcal{E}_j^2)(3\mathcal{I} - (\mathcal{I} - \mathcal{E}_j^2))) \\
&= \mathcal{X}_j(\mathcal{I} + \mathcal{E}_j)(3\mathcal{I} - (\mathcal{I} - \mathcal{E}_j^2)(2\mathcal{I} + \mathcal{E}_j^2)) \\
&= \mathcal{X}_j(\mathcal{I} + \mathcal{E}_j)(3\mathcal{I} - (2\mathcal{I} + \mathcal{E}_j^2 - 2\mathcal{E}_j^2 - \mathcal{E}_j^4)) \\
&= \mathcal{X}_j(\mathcal{I} + \mathcal{E}_j)(\mathcal{I} + \mathcal{E}_j^2 + \mathcal{E}_j^4) \\
&= \mathcal{X}_j(\mathcal{I} + \mathcal{E}_j + \mathcal{E}_j^2 + \mathcal{E}_j^3 + \mathcal{E}_j^4 + \mathcal{E}_j^5).
\end{aligned}$$

Thus

$$\begin{aligned}
\mathcal{E}_{j+1} &= \mathcal{I} - \mathcal{A}\mathcal{X}_{j+1} = \mathcal{I} - \mathcal{A}\mathcal{X}_j(\mathcal{I} + \mathcal{E}_j + \mathcal{E}_j^2 + \mathcal{E}_j^3 + \mathcal{E}_j^4 + \mathcal{E}_j^5) \\
&= \mathcal{I} - (\mathcal{I} - \mathcal{E}_j)(\mathcal{I} + \mathcal{E}_j + \mathcal{E}_j^2 + \mathcal{E}_j^3 + \mathcal{E}_j^4 + \mathcal{E}_j^5) = \mathcal{E}_j^6.
\end{aligned} \tag{7}$$

Hence, by taking tensor norm from both sides of the Eq. (7), we obtain

$$\|\mathcal{E}_{j+1}\| \leq \|\mathcal{E}_j\|^6. \tag{8}$$

Using Eq. (8) and mathematical induction, it is not difficult to verify that

$$\|\mathcal{E}_{j+1}\| \leq \|\mathcal{E}_j\|^6 \leq \|\mathcal{E}_{j-1}\|^{6^2} \leq \cdots \leq \|\mathcal{E}_0\|^{6^{j+1}} < 1, \quad j \geq 0.$$

Hence, the sequence $\{\|\mathcal{E}_j\|\}$ is monotonic decreasing. This implies that $\|\mathcal{E}_j\| \to 0$ when $j \to \infty$ and thus $\mathcal{X}_j \to \mathcal{A}^{-1}$ as $j \to \infty$. To complete the proof, it should be shown that the sixth order of convergence is obtained for the sequence $\{\mathcal{X}_j\}$. For this purpose, we define $\epsilon_j = \mathcal{A}^{-1} - \mathcal{X}_j$ as the error tensor in the FNS algorithm. Thus we have $\mathcal{A}\epsilon_j = \mathcal{I} - \mathcal{A}\mathcal{X}_j = \mathcal{E}_j$. This implies that $\mathcal{A}\epsilon_{j+1} = (\mathcal{A}\epsilon_j)^6$, which gives

$$\epsilon_{j+1} = \epsilon_j(\mathcal{A}\epsilon_j)^5. \tag{9}$$

By taking tensor norm from both sides of Eq. (9), we obtain

$$\|\epsilon_{j+1}\| \leq \|\mathcal{A}\|^5 \|\epsilon_j\|^6. \tag{10}$$

Eq. (10) shows that the FNS algorithm is convergent of order six. □

**Proposition 3.** *Let $\{\mathcal{X}_j\}$ be the sequence generated by the FNS algorithm and the same assumptions of Proposition 2 hold. For $j = 0, 1, \ldots$, let*

$$\Delta\mathcal{X}_j = \tilde{\mathcal{X}}_j - \mathcal{X}_j, \tag{11}$$

*be a numerical perturbation that occurs at the $j$-th exact iterate $\mathcal{X}_j$ of FNS algorithm and is so small that we ignore the terms contain $(\Delta\mathcal{X}_j)^l$ for $l \geq 2$. Then $\|\Delta\mathcal{X}_{j+1}\| \leq \Gamma \|\Delta\mathcal{X}_0\|$ ,where $\Gamma = 6^{j+1} \prod_{i=0}^{j} \max\{1, \|\mathcal{E}_i\|^5\}(1 + 5\|\mathcal{A}\| \|\mathcal{X}_i\|)$.*

*Proof.* Using Step 3 of Algorithm 2, we have

$$\begin{aligned}\Delta\mathcal{X}_{j+1} &= \tilde{\mathcal{X}}_{j+1} - \mathcal{X}_{j+1} \\ &= \tilde{\mathcal{X}}_j(\mathcal{I} + \tilde{\mathcal{E}}_j + \tilde{\mathcal{E}}_j^2 + \tilde{\mathcal{E}}_j^3 + \tilde{\mathcal{E}}_j^4 + \tilde{\mathcal{E}}_j^5) - \mathcal{X}_j(\mathcal{I} + \mathcal{E}_j + \mathcal{E}_j^2 + \mathcal{E}_j^3 + \mathcal{E}_j^4 + \mathcal{E}_j^5) \\ &= (\mathcal{X}_j + \Delta\mathcal{X}_j)\sum_{k=0}^{5}\tilde{\mathcal{E}}_j^k - \mathcal{X}_j\sum_{k=0}^{5}\mathcal{E}_j^k \\ &= \Delta\mathcal{X}_j\sum_{k=0}^{5}\tilde{\mathcal{E}}_j^k + \mathcal{X}_j\sum_{k=0}^{5}(\tilde{\mathcal{E}}_j^k - \mathcal{E}_j^k).\end{aligned} \tag{12}$$

Taking tensor norm from both sides of Eq. (12), we have

$$\|\Delta\mathcal{X}_{j+1}\| \le \|\Delta\mathcal{X}_j\|\sum_{k=0}^{5}\left\|\tilde{\mathcal{E}}_j\right\|^k + \|\mathcal{X}_j\|\sum_{k=0}^{5}\left\|\tilde{\mathcal{E}}_j^k - \mathcal{E}_j^k\right\|. \tag{13}$$

Now using definition $\tilde{\mathcal{E}}_j = \mathcal{I} - \mathcal{A}\tilde{\mathcal{X}}_j = \mathcal{E}_j - \mathcal{A}\Delta\mathcal{X}_j$, we have

$$\left\|\tilde{\mathcal{E}}_j\right\|^k = \left\|(\mathcal{E}_j - \mathcal{A}\Delta\mathcal{X}_j)^k\right\| \le (\|\mathcal{E}_j\| + \|\mathcal{A}\Delta\mathcal{X}_j\|)^k \le (\|\mathcal{E}_j\| + \|\mathcal{A}\|\,\|\Delta\mathcal{X}_j\|)^k = \gamma_0^k, \tag{14}$$

where $\gamma_0 = \|\mathcal{E}_j\| + \mathcal{O}(\|\Delta\mathcal{X}_j\|)$ and $k = 0, 1, \ldots, 5$. In addition, we obtain

$$\begin{aligned}\left\|\tilde{\mathcal{E}}_j^k - \mathcal{E}_j^k\right\| = \left\|(\mathcal{E}_j - \mathcal{A}\Delta\mathcal{X}_j)^k - \mathcal{E}_j^k\right\| &= \left\|\sum_{i=0}^{k}(-1)^i\binom{k}{i}\mathcal{E}_j^i(\mathcal{A}\Delta\mathcal{X}_j)^{k-i} - \mathcal{E}_j^k\right\| \\ &\le \|\mathcal{A}\Delta\mathcal{X}_j\|\sum_{i=0}^{k-1}\binom{k}{k-1-i}\|\mathcal{E}_j\|^{k-1-i}\,\|\mathcal{A}\Delta\mathcal{X}_j\|^i \le \lambda_k\,\|\mathcal{A}\|\,\|\Delta\mathcal{X}_j\|,\end{aligned} \tag{15}$$

where

$$\lambda_k = \sum_{i=0}^{k-1}\binom{k}{k-1-i}\|\mathcal{E}_j\|^{k-1-i}\,\|\mathcal{A}\Delta\mathcal{X}_j\|^i = k\,\|\mathcal{E}_j\|^{k-1} + \mathcal{O}(\|\Delta\mathcal{X}_j\|).$$

By substituting Eqs. (14) and (15) in Eq. (13), we get

$$\|\Delta\mathcal{X}_{j+1}\| \le \|\Delta\mathcal{X}_j\|\sum_{k=0}^{5}(\gamma_0^k + \lambda_k\,\|\mathcal{A}\|\,\|\mathcal{X}_j\|) \le (6\max\{1, \|\mathcal{E}_j\|^5\}(1 + 5\,\|\mathcal{A}\|\,\|\mathcal{X}_j\|)) + \mathcal{O}(\|\Delta\mathcal{X}_j\|),$$

and subsequently

$$\|\Delta\mathcal{X}_{j+1}\| \le 6^{j+1}\prod_{i=0}^{j}\max\{1, \|\mathcal{E}_i\|^5\}(1 + 5\,\|\mathcal{A}\|\,\|\mathcal{X}_i\|) + \mathcal{O}(\|\Delta\mathcal{X}_j\|),$$

which completes the proof. □

As with all iterative methods, for convergence, choosing of initial value $\mathcal{X}_0$ is very important which is true for the FNS method as well. By choosing the following initial value of $\mathcal{X}_0$, which is the logical extension of choosing the initial value for matrix case (see [8]), it holds that $\|\mathcal{I} - \mathcal{A}\mathcal{X}_0\| < 1$, when

$$\mathcal{X}_0 = \alpha\mathcal{A}^*,\ 0 < \alpha < \frac{2}{\lambda_1(\mathcal{A}^*\mathcal{A})},$$

where $\lambda_1(\mathcal{A}^*\mathcal{A}) = \|\mathcal{A}\|^2$ is the largest eigenvalue of $\mathcal{A}^*\mathcal{A}$ [13].

## 3 Computational aspects

Now, we will analyze the computational complexity of the FNS method. Since the "addition" and "subtraction" operations are commonly ignored in computational analysis, we only consider the "multiplication" operation in the algorithm. For this purpose, let us consider the following computational efficiency index as given by Traub in Appendix C of [41] for matrix

$$C.E.I = p^{\frac{1}{c}},$$

where $c$ stands for the total computational cost of an algorithm and $p$ is the local convergence order. Let us assume that the cost of tensor-tensor multiplication be unity (as Traub made in [41] for matrix). Then the computational efficiency index with number of tensor-tensor multiplication per step becomes

$$C.E.I = p^{\frac{1}{\eta s}},$$

where $\eta$ is the number of tensor-tensor multiplications and $s$ is the number of iterations (steps) that an iterative algorithm requires to converge.

Soderstrom and Stewart in [36] illustrated that the approximate number of iterations, to converge the Newton-Schultz scheme (4) in a machines precision, is given by

$$s \approx 2\log_2 \kappa_2(A), \tag{16}$$

where $\kappa_2$ denotes the condition number of the matrix $A$ in 2-norm. Hence, similar to (16) under the same conditions, the required approximate number of iterations, for a $p$-th order iterative method to converge [36], is given by

$$s \approx 2\log_p \kappa_2(A).$$

Therefore, the computational efficiency index of a $p$-th order matrix iterative method with $\eta$ number of matrix-matrix multiplication per cycle becomes

$$C.E.I = p^{\frac{1}{\eta(2\log_p \kappa_2(A))}}.$$

For using this index for comparing the FNS method with the others, we define the new index as follows

$$C.E.I = p^{\frac{1}{\eta(2\log_p \kappa_2(\Phi(\mathcal{A})))}}, \tag{17}$$

where $\Phi(\mathcal{A})$ is defined as in Definition 1.

**Example 1.** Using the new index (17), we consider 10 real random tensors of the form $\mathcal{A} = 17 * tenrand([25\ 24\ 25\ 24]) \in \mathbb{R}^{25\times24\times25\times24}$. Choosing $\mathcal{X}_0 = \alpha\mathcal{A}^T$ with $\alpha = 1/\|\mathcal{A}\|^2$ and the stopping criterion $||\mathcal{E}_j|| < 10^{-10}||\mathcal{E}_0||$, where $\mathcal{E}_j = \mathcal{I} - \mathcal{A} *_2 \mathcal{X}_j$, we obtain the approximate inverse of $\mathcal{A}$. A comparison has been made in Figure 1 of the iterative methods NS, FS, LL and FNS. From Figure 1, we see that the FNS method converges to the exact solution in less iterations and CPU time, also proves that the new definition of the condition numbers of a tensor is applicable and can be regarded as a good criterion for comparing various methods. Here the proposed algorithm shows its dominance in terms of computational efficiency.

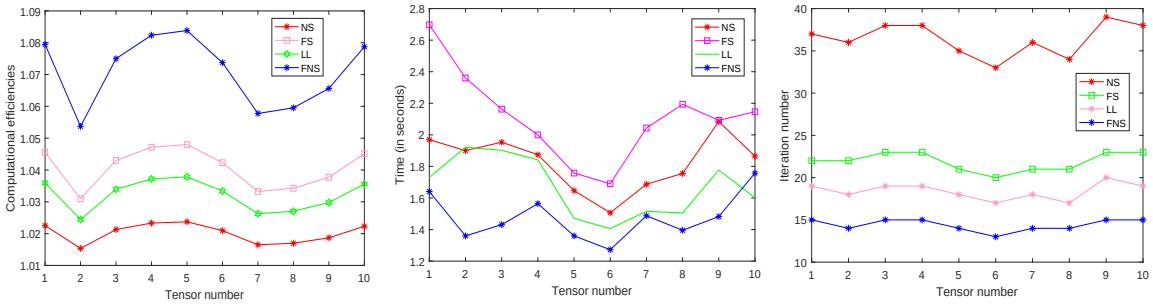


Figure 1: Comparison result of computational efficiency index.

The FNS method posses the sixth order of convergence using only five tensor-tensor multiplication, while the schemes NS, FS and LL reach 2nd, 3rd and 4th orders, respectively, by consuming 2, 4 and 4 tensor-tensor multiplication.

One can apply the definition of the inverse-finder informational efficiency index as

$$I.I.E.I = \frac{p}{\eta}, \tag{18}$$

in which $p$ stands for the local order of convergence and $\eta$ is the number of tensor-tensor multiplication per computing step.

Table 1: Comparison of the computational complexity for different methods.

| Methods | NS | FS | LL | FNS |
|---|---|---|---|---|
| Rate of convergence | 2 | 3 | 4 | 6 |
| Number of tensor-tensor multiplications | 2 | 4 | 4 | 5 |
| I.I.E.I | $\frac{2}{2}=1$ | $\frac{3}{4}=0.75$ | $\frac{4}{4}=1$ | $\frac{6}{5}=1.2$ |

In Table 1, we illustrate a comparison of rate of convergence, the number of tensor-tensor multiplications and the index (18) for different methods. The results show that the new established method is better than the others. In fact, by comparing these results, one can see that the proposed method reduces the computational complexity by using less basic operations and leads to a better equilibrium between high speed and operational cost.

Another index which can be used for comparing Newton-Shultz-type inverse finder methods is the Numerical Local Convergence Order [37] as follows:

$$NLCO = \frac{\ln\left(\frac{\|\mathcal{E}_{j+1}\|}{\|\mathcal{E}_j\|}\right)}{\ln\left(\frac{\|\mathcal{E}_j\|}{\|\mathcal{E}_{j-1}\|}\right)},$$

wherein three last approximations $\mathcal{X}_{j-1}$, $\mathcal{X}_j$ and $\mathcal{X}_{j+1}$ are used.

This definition is useful when applying an iterative method in the high precision computing environment and to observe numerically the maximum local convergence order of different methods for solving academic tests.

Table 2: Results of comparisons for different methods in terms of NLCO.

| Methods | NS | FS | LL | FNS |
|---|---|---|---|---|
| Average of NLCO | 1.9998 | 3.1266 | 3.9996 | 5.9292 |
| Average of iteration | 39.3333 | 23.1667 | 19.8167 | 15.3333 |

**Example 2.** Let us consider six different random tensors of the form $\mathcal{A} = 13 * tenrand([12\ 10\ 8\ 12\ 10\ 8]) \in \mathbb{R}^{12\times10\times8\times12\times10\times8}$. Choosing $\mathcal{X}_0 = \alpha\mathcal{A}^T$, $\alpha = 1/\|\mathcal{A}\|^2$ and the stopping criterion $||\mathcal{E}_j|| < 10^{-10}||\mathcal{E}_0||$, we obtained the approximation inverse of $\mathcal{A}$ and compared the average of NLCO of the methods. The numerical results are shown in Table 2. The numerical results for local convergence orders uphold the theoretical aspects of the paper.

## 4 The Moore-Penrose inverse using FNS

**Definition 7.** *The Moore-Penrose inverse of an arbitrary matrix $A \in \mathbb{C}^{m\times n}$ denoted by $A^\dagger$ is the matrix $X$ satisfying the following four Penrose conditions*

$$AXA = A, \quad XAX = X, \quad (AX)^* = AX, \quad (XA)^* = XA.$$

It is well known that the Moore-Penrose inverse of any arbitrary matrix $A$ is exists. Sun et al. in [39] extended the Moore-Penrose inverse of tensors via Einstein tensor product as follows.

**Definition 8.** *[39] Let $\mathcal{A} \in \mathbb{C}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_N}$. The tensor $\mathcal{X} \in \mathbb{C}^{J_1\times\cdots\times J_N\times I_1\times\cdots\times I_N}$ satisfying*

$$(1)\ \mathcal{A} *_N \mathcal{X} *_N \mathcal{A} = \mathcal{A},$$
$$(2)\ \mathcal{X} *_N \mathcal{A} *_N \mathcal{X} = \mathcal{X},$$
$$(3)\ (\mathcal{A} *_N \mathcal{X})^* = \mathcal{A} *_N \mathcal{X},$$
$$(4)\ (\mathcal{X} *_N \mathcal{A})^* = \mathcal{X} *_N \mathcal{A},$$

*is called the Moore-Penrose inverse of $\mathcal{A}$ denoted by $\mathcal{A}^\dagger$. If $(i)$, $i = 1, 2, 3, 4$ of the above equations holds, then $\mathcal{X}$ is called an $\{i\}$-inverse of $\mathcal{A}$, denoted by $\mathcal{A}^{(i)}$. For a tensor $\mathcal{A} \in \mathbb{C}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$, if $\mathcal{A}$ is invertible, then $\mathcal{A}^\dagger = \mathcal{A}^{-1}$.*

In this section by using the FNS algorithm and taking $\mathcal{X}_0 = \alpha\mathcal{A}^*$ where $0 < \alpha < 2/\|\mathcal{A}\|^2$, we approximate the Moore-Penrose inverse of $\mathcal{A}$ with sixth order convergence.

**Proposition 4.** *For the sequence $\{\mathcal{X}_j\}$ generated by the FNS algorithm and $\mathcal{X}_0 = \alpha\mathcal{A}^*$, it holds that*

$$\left(\mathcal{X}_j\mathcal{A}\right)^* = \mathcal{X}_j\mathcal{A}, \qquad \left(\mathcal{A}\mathcal{X}_j\right)^* = \mathcal{A}\mathcal{X}_j, \qquad \mathcal{A}^\dagger\mathcal{A}\mathcal{X}_j = \mathcal{X}_j, \qquad \mathcal{X}_j\mathcal{A}\mathcal{A}^\dagger = \mathcal{X}_j. \tag{19}$$

*Proof.* By using this fact that $\mathcal{X}_{j+1} = \mathcal{X}_j(6\mathcal{I} - 15\mathcal{Q}_j + 20\mathcal{Q}_j^2 - 15\mathcal{Q}_j^3 + 6\mathcal{Q}_j^4 - \mathcal{Q}_j^5)$. Consider the first equation in (19), for $j = 0$, with $\mathcal{X}_0 = \alpha\mathcal{A}^*$, we have

$$\left(\mathcal{X}_0\mathcal{A}\right)^* = \left(\alpha\mathcal{A}^*\mathcal{A}\right)^* = \alpha\mathcal{A}^*\mathcal{A} = \mathcal{X}_0\mathcal{A}.$$

Let the first equation in (19) holds for $j = k$. For $j = k+1$, we have

$$\begin{aligned}
(\mathcal{X}_{j+1}\mathcal{A})^* &= \Big(\mathcal{X}_j(6\mathcal{I} - 15\mathcal{Q}_j + 20\mathcal{Q}_j^2 - 15\mathcal{Q}_j^3 + 6\mathcal{Q}_j^4 - \mathcal{Q}_j^5)\mathcal{A}\Big)^*, \quad \mathcal{Q}_j = \mathcal{A}\mathcal{X}_j \\
&= \Big(\mathcal{X}_j\big(6\mathcal{I} - 15\mathcal{A}\mathcal{X}_j + 20(\mathcal{A}\mathcal{X}_j)^2 - 15(\mathcal{A}\mathcal{X}_j)^3 + 6(\mathcal{A}\mathcal{X}_j)^4 - (\mathcal{A}\mathcal{X}_j)^5\big)\mathcal{A}\Big)^* \\
&= 6(\mathcal{X}_j\mathcal{A})^* - 15\Big((\mathcal{X}_j\mathcal{A})^*\Big)^2 + 20\Big((\mathcal{X}_j\mathcal{A})^*\Big)^3 - 15\Big((\mathcal{X}_j\mathcal{A})^*\Big)^4 \\
&\quad + 6\Big((\mathcal{X}_j\mathcal{A})^*\Big)^5 - \Big((\mathcal{X}_j\mathcal{A})^*\Big)^6 \\
&= 6(\mathcal{X}_j\mathcal{A}) - 15(\mathcal{X}_j\mathcal{A})^2 + 20(\mathcal{X}_j\mathcal{A})^3 - 15(\mathcal{X}_j\mathcal{A})^4 + 6(\mathcal{X}_j\mathcal{A})^5 - (\mathcal{X}_j\mathcal{A})^6 \\
&= \mathcal{X}_j\big(6\mathcal{I} - 15\mathcal{A}\mathcal{X}_j + 20(\mathcal{A}\mathcal{X}_j)^2 - 15(\mathcal{A}\mathcal{X}_j)^3 + 6(\mathcal{A}\mathcal{X}_j)^4 - (\mathcal{A}\mathcal{X}_j)^5\big)\mathcal{A} \\
&= \mathcal{X}_j(6\mathcal{I} - 15\mathcal{Q}_j + 20\mathcal{Q}_j^2 - 15\mathcal{Q}_j^3 + 6\mathcal{Q}_j^4 - \mathcal{Q}_j^5)\mathcal{A} = \mathcal{X}_{j+1}\mathcal{A},
\end{aligned}$$

where the fourth equality uses this fact that $(\mathcal{X}_j\mathcal{A})^* = \mathcal{X}_j\mathcal{A}$. A similar discussion can be used to the proof of other relations in (19). □

**Proposition 5.** *Let $\mathcal{A} \in \mathbb{C}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_N}$ with the singular values $\sigma_1 > \sigma_2 > \cdots > \sigma_r > 0$ and the initial approximation $\mathcal{X}_0 = \alpha\mathcal{A}^*$ where $0 < \alpha < 1/\sigma_1^2$. Then it holds $\big\|\mathcal{A}(\mathcal{X}_0 - \mathcal{A}^\dagger)\big\| < 1$.*

*Proof.* Let $\mathcal{P}, \mathcal{S} \in \mathbb{C}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$ such that $\mathcal{P}^2 = \mathcal{P}$ and $\mathcal{P}\mathcal{S} = \mathcal{S}\mathcal{P}$ and $\rho(\mathcal{A}) = \max_{\lambda\in\sigma(\mathcal{A})}|\lambda|$ then $\rho(\mathcal{P}\mathcal{S}) \le \rho(\mathcal{S})$, because if $(\lambda, \mathcal{X})$ be the eigenpair of $\mathcal{P}\mathcal{S}$ then $\mathcal{P}\mathcal{S}\mathcal{X} = \lambda\mathcal{X}$ results in $\lambda\mathcal{P}\mathcal{X} = \lambda\mathcal{X}$ which implies that $\lambda = 0$ or $\mathcal{P}\mathcal{X} = \mathcal{X}$. If $\mathcal{P}\mathcal{X} = \mathcal{X}$ then $\mathcal{S}\mathcal{X} = \mathcal{S}\mathcal{P}\mathcal{X} = \mathcal{P}\mathcal{S}\mathcal{X} = \lambda\mathcal{X}$, i.e. $\lambda$ is an eigenvalue of $\mathcal{S}$, which implies that $\rho(\mathcal{P}\mathcal{S}) \le \rho(\mathcal{S})$. Set $\mathcal{P} = \mathcal{A}\mathcal{A}^\dagger$ and $\mathcal{S} = \mathcal{A}\mathcal{X}_0 - \mathcal{I}$. It is not difficult to see that $\mathcal{P}^2 = \mathcal{P}$ and $\mathcal{P}\mathcal{S} = \mathcal{S}\mathcal{P}$. Thus according to Proposition 4, we have

$$\rho(\mathcal{A}(\mathcal{X}_0 - \mathcal{A}^\dagger)) = \rho(\mathcal{A}(\alpha\mathcal{A}^* - \mathcal{A}^\dagger)) \le \rho(\alpha\mathcal{A}\mathcal{A}^* - \mathcal{I}) = \max_{1\le i\le r}|1 - \alpha\lambda_i(\mathcal{A}\mathcal{A}^*)| < 1.$$

It is well known that there exists a positive constant $\epsilon$ for some tensor norm, such that

$$\Big\|\mathcal{A}(\mathcal{X}_0 - \mathcal{A}^\dagger)\Big\| < \rho(\mathcal{A}(\mathcal{X}_0 - \mathcal{A}^\dagger)) + \epsilon < 1,$$

which completes the proof. □

**Proposition 6.** *For the tensor $\mathcal{A}$ with singular values $\sigma_1 > \sigma_2 > \cdots > \sigma_r > 0$ and the initial approximation $\mathcal{X}_0 = \alpha\mathcal{A}^*$ where $0 < \alpha < 1/\sigma_1^2$, the tensor sequence $\{\mathcal{X}_j\}$ generated by the FNS algorithm is convergent to the Moore-Penrose inverse of $\mathcal{A}$ with sixth order convergence. More precisely $\Big\|\mathcal{E}'_{j+1}\Big\| \le \big\|\mathcal{A}^\dagger\big\|\,\|\mathcal{A}\|^6\Big\|\mathcal{E}'_j\Big\|^6$, where $\mathcal{E}'_{j+1} = \mathcal{X}_{j+1} - \mathcal{A}^\dagger$.*

*Proof.* Using the properties of $\mathcal{A}^\dagger$, we have

$$(\mathcal{I} - \mathcal{A}\mathcal{A}^\dagger)^j = \mathcal{I} - \mathcal{A}\mathcal{A}^\dagger,\ (\mathcal{I} - \mathcal{A}\mathcal{A}^\dagger)\mathcal{A}\mathcal{E}'_j = 0,\ j = 1, 2, 3, \ldots. \tag{20}$$

Thus we obtain

$$\begin{aligned}
\mathcal{A}\mathcal{E}'_{j+1} &= \mathcal{A}\mathcal{X}_{j+1} - \mathcal{A}\mathcal{A}^\dagger = \mathcal{A}\mathcal{X}_{j+1} - \mathcal{I} + \mathcal{I} - \mathcal{A}\mathcal{A}^\dagger \\
&= -\mathcal{E}_{j+1} + \mathcal{I} - \mathcal{A}\mathcal{A}^\dagger = -\big(\mathcal{E}_j\big)^6 + \mathcal{I} - \mathcal{A}\mathcal{A}^\dagger,
\end{aligned} \tag{21}$$

and by using Eq. (20), we have

$$\begin{aligned}
-(\mathcal{E}_j)^6+\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger &= -(\mathcal{I}-\mathcal{A}\mathcal{X}_j)^6+\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger \\
&= -(\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger+\mathcal{A}\mathcal{A}^\dagger-\mathcal{A}\mathcal{X}_j)^6+\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger \\
&= -(\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger-\mathcal{A}\mathcal{E}'_j)^6+\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger \\
&= -\Big[\sum_{i=0}^{6}(-1)^i\binom{6}{i}\Big(\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger\Big)^i\Big(\mathcal{A}\mathcal{E}'_j\Big)^{6-i}\Big]+\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger \\
&= -\Big[\Big(\mathcal{A}\mathcal{E}'_j\Big)^6+\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger\Big]+\mathcal{I}-\mathcal{A}\mathcal{A}^\dagger \\
&= -\Big(\mathcal{A}\mathcal{E}'_j\Big)^6.
\end{aligned} \tag{22}$$

By substituting Eq. (22) in Eq. (21), we deduce that

$$\mathcal{A}\mathcal{E}'_{j+1} = -\Big(\mathcal{A}\mathcal{E}'_j\Big)^6.$$

By using Proposition 5 which implies $\|\mathcal{A}\mathcal{E}'_0\|<1$ and what was obtained from the proof process of the Proposition 2, we have

$$\big\|\mathcal{A}\mathcal{E}'_{j+1}\big\| \le \big\|\mathcal{A}\mathcal{E}'_j\big\|^6 \le \|\mathcal{A}\|^6\big\|\mathcal{E}'_j\big\|^6, \quad j=0,1,2,\ldots,$$

then by the properties of the Moore-Penrose inverse and Proposition 4, we obtain

$$\begin{aligned}
\big\|\mathcal{X}_{j+1}-\mathcal{A}^\dagger\big\| &= \big\|\mathcal{A}^\dagger\mathcal{A}\mathcal{X}_{j+1}-\mathcal{A}^\dagger\mathcal{A}\mathcal{A}^\dagger\big\| \le \big\|\mathcal{A}^\dagger\big\|\,\big\|\mathcal{A}\mathcal{X}_{j+1}-\mathcal{A}\mathcal{A}^\dagger\big\| \\
&= \big\|\mathcal{A}^\dagger\big\|\,\Big\|\mathcal{A}\mathcal{E}'_{j+1}\Big\| \le \big\|A^\dagger\big\|\,\|\mathcal{A}\|^6\Big\|\mathcal{E}'_j\Big\|^6.
\end{aligned}$$

This inequality results $\big\|\mathcal{X}_j-\mathcal{A}^\dagger\big\|\to 0$ as $j\to\infty$. In other words, the FNS algorithm is convergent to the Moore-Penrose inverse of $\mathcal{A}$ of order six. □

**Proposition 7.** *Let $\mathcal{A}\in\mathbb{C}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_N}$. If $\mathcal{A}\mathcal{X}_0=\mathcal{X}_0\mathcal{A}$, and $\{\mathcal{X}_j\}$ be the sequence obtained by FNS, then for every $j=1,2,\ldots$, we have*

$$\mathcal{A}\mathcal{X}_j=\mathcal{X}_j\mathcal{A}. \tag{23}$$

*Proof.* From FNS, we have

$$\mathcal{X}_{j+1}=\mathcal{X}_j(2\mathcal{I}-\mathcal{A}\mathcal{X}_j)(3\mathcal{I}-\mathcal{A}\mathcal{X}_j(2\mathcal{I}-\mathcal{A}\mathcal{X}_j)(3\mathcal{I}-\mathcal{A}\mathcal{X}_j(2\mathcal{I}-\mathcal{A}\mathcal{X}_j))).$$

Since $\mathcal{A}\mathcal{X}_0=\mathcal{X}_0\mathcal{A}$, using the last equation, we have

$$\begin{aligned}
\mathcal{A}\mathcal{X}_1 &= \mathcal{A}\Big(\mathcal{X}_0(2\mathcal{I}-\mathcal{A}\mathcal{X}_0)(3\mathcal{I}-\mathcal{A}\mathcal{X}_0(2\mathcal{I}-\mathcal{A}\mathcal{X}_0)(3\mathcal{I}-\mathcal{A}\mathcal{X}_0(2\mathcal{I}-\mathcal{A}\mathcal{X}_0)))\Big) \\
&= \mathcal{A}\mathcal{X}_0\Big((2\mathcal{I}-\mathcal{A}\mathcal{X}_0)(3\mathcal{I}-\mathcal{A}\mathcal{X}_0(2\mathcal{I}-\mathcal{A}\mathcal{X}_0)(3\mathcal{I}-\mathcal{A}\mathcal{X}_0(2\mathcal{I}-\mathcal{A}\mathcal{X}_0)))\Big) \\
&= \Big(\mathcal{X}_0(2\mathcal{I}-\mathcal{X}_0\mathcal{A})(3\mathcal{I}-\mathcal{X}_0\mathcal{A}(2\mathcal{I}-\mathcal{X}_0\mathcal{A})(3\mathcal{I}-\mathcal{X}_0\mathcal{A}(2\mathcal{I}-\mathcal{X}_0\mathcal{A})))\Big)\mathcal{A} \\
&= \mathcal{X}_1\mathcal{A}.
\end{aligned}$$

Thus Eq. (23) holds for $j = 1$. To complete the proof, we use mathematical induction. Assume that $\mathcal{A}\mathcal{X}_j = \mathcal{X}_j\mathcal{A}$, then for all $j > 1$:

$$\begin{aligned}
\mathcal{A}\mathcal{X}_{j+1} &= \mathcal{A}\Big(\mathcal{X}_j(2\mathcal{I} - \mathcal{A}\mathcal{X}_j)(3\mathcal{I} - \mathcal{A}\mathcal{X}_j(2\mathcal{I} - \mathcal{A}\mathcal{X}_j)(3\mathcal{I} - \mathcal{A}\mathcal{X}_j(2\mathcal{I} - \mathcal{A}\mathcal{X}_j)))\Big) \\
&= \mathcal{A}\mathcal{X}_j\Big((2\mathcal{I} - \mathcal{A}\mathcal{X}_j)(3\mathcal{I} - \mathcal{A}\mathcal{X}_j(2\mathcal{I} - \mathcal{A}\mathcal{X}_j)(3\mathcal{I} - \mathcal{A}\mathcal{X}_j(2\mathcal{I} - \mathcal{A}\mathcal{X}_j)))\Big) \\
&= \Big(\mathcal{X}_j(2\mathcal{I} - \mathcal{X}_j\mathcal{A})(3\mathcal{I} - \mathcal{X}_j\mathcal{A}(2\mathcal{I} - \mathcal{X}_j\mathcal{A})(3\mathcal{I} - \mathcal{X}_j\mathcal{A}(2\mathcal{I} - \mathcal{X}_j\mathcal{A})))\Big)\mathcal{A} \\
&= \mathcal{X}_{j+1}\mathcal{A}.
\end{aligned}$$

This completes the proof. □

Let $\mathcal{A} \in \mathbb{C}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_N}$, $\mathcal{X} \in \mathbb{C}^{J_1\times\cdots\times J_N}$ and $\mathcal{B} \in \mathbb{C}^{I_1\times\cdots\times I_N}$. If there exists a tensor $\hat{\mathcal{X}}$ satisfying $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$, then we call $\hat{\mathcal{X}}$ is a solution of $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$ and say tensor equation $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$ is consistent. Else the system $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$ is inconsistent and the tensor $\hat{\mathcal{X}}$ which minimizes $\|\mathcal{A} *_N \mathcal{X} - \mathcal{B}\|$ is called a minimum-norm solution of inconsistent multilinear system $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$.

**Proposition 8.** *If the multilinear system $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$ is inconsistent, then the minimum-norm solution is $\mathcal{X} = \mathcal{A}^\dagger *_N \mathcal{B}$.*

## 5 The new preconditioner for solving multilinear system

Consider the multilinear system

$$\mathcal{A} *_N \mathcal{X} = \mathcal{B}, \tag{24}$$

where $\mathcal{A} \in \mathbb{C}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$, $\mathcal{X}, \mathcal{B} \in \mathbb{C}^{I_1\times\cdots\times I_N}$. The direct methods for solving the consistent multilinear system are expensive because a lot of work and storage are required. To overcome this problem, it is efficient to apply the iterative methods for solving multilinear systems. Iterative methods may have a poor convergence or even fail to converge. To remedy this drawback, the iterative methods usually involve a second tensor that transforms the coefficient tensor $\mathcal{A}$ into one favourable tensor. The involved tensor is called a preconditioner. Let $\mathcal{M}$ be a tensor that approximates the inverse of $\mathcal{A}$ ($\mathcal{M} \approx \mathcal{A}^{-1}$), then the transformed multilinear system $\mathcal{A} *_N \mathcal{M} *_N \mathcal{Y} = \mathcal{B}$, $\mathcal{X} = \mathcal{M} *_N \mathcal{Y}$ will have the same solution as the system (24), and the convergence rate of iterative methods applied to the preconditioned system might be higher. Similar to this system that is preconditioned from the right, the left preconditioning is also possible, i.e., $\mathcal{M} *_N \mathcal{A} *_N \mathcal{X} = \mathcal{M} *_N \mathcal{B}$. Generally, the preconditioner $\mathcal{M}$ should be chosen such that $\mathcal{M} *_N \mathcal{A}$ or $\mathcal{A} *_N \mathcal{M}$ be a good approximation of the identity tensor. Note that it is not easy to find this kind of preconditioner for explain.

The FNS method can be considered for finding preconditioners of tensors as well. The convergence of order higher than six, allows us to find an efficient approximate inverse. If the stopping criterion is satisfactory, one may stop, else the obtained approximate inverse can be taken into account as a preconditioner to cluster the eigenvalues and then allows the iteration processes. So the storage requirement will be low and the method could successfully be convergent.

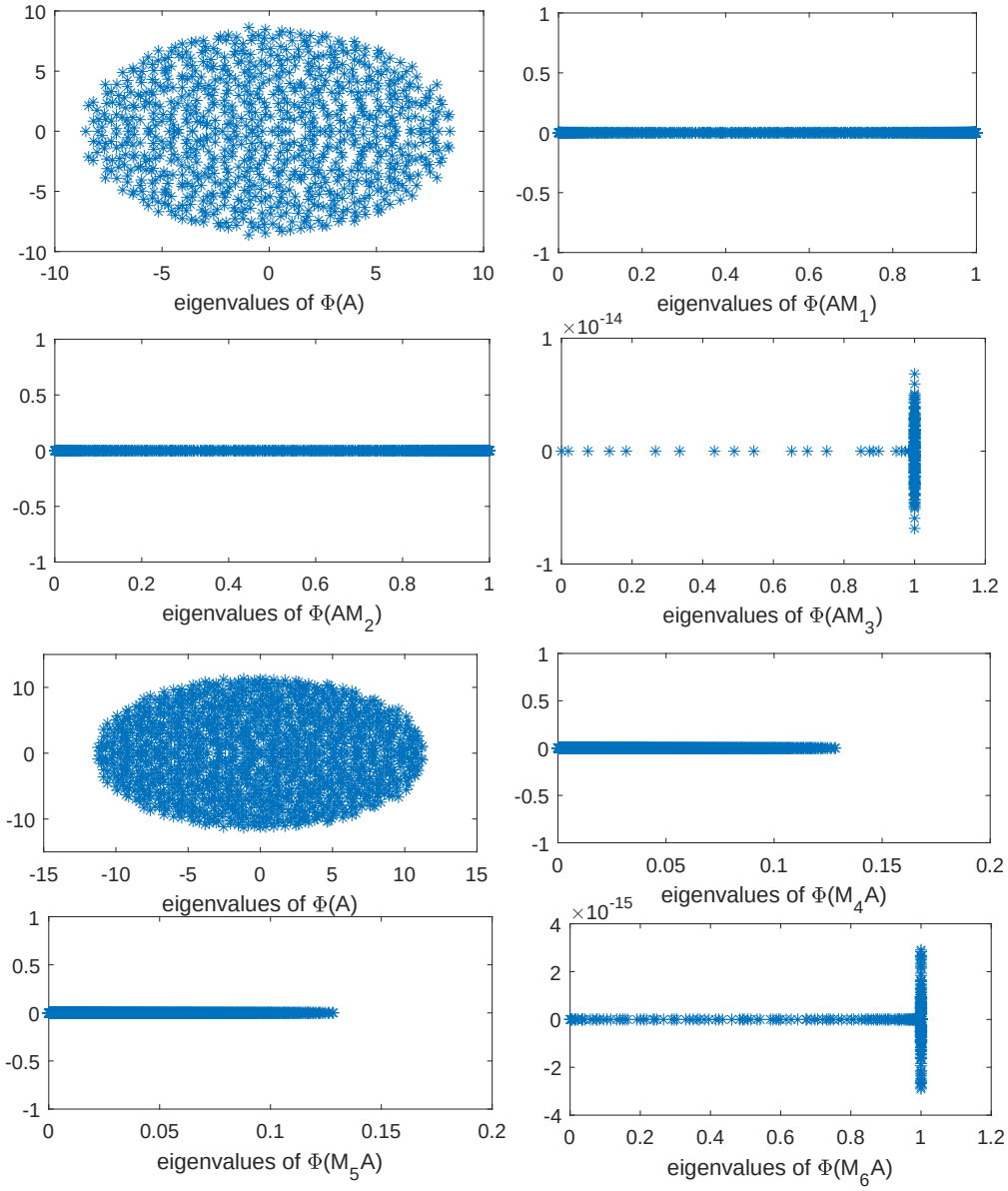


Figure 2: Comparison of eigenvalues of $\Phi(\mathcal{A})$, $\Phi(\mathcal{A}\mathcal{M}_i)$ and $\Phi(\mathcal{M}_j\mathcal{A})$.

**Example 3.** We consider the academical tests as follows. Let $\mathcal{A} = tenrand([n\ n-1\ n\ n-1]) \in \mathbb{R}^{n\times(n-1)\times n\times(n-1)}$. For $n = 30$, take $\mathcal{M}_1 = \mathcal{X}_5$, $\mathcal{M}_2 = \mathcal{X}_7$, $\mathcal{M}_3 = \mathcal{X}_9$ as the right preconditioners, and for $n = 40$, let $\mathcal{M}_4 = \mathcal{X}_3$, $\mathcal{M}_5 = \mathcal{X}_5$, $\mathcal{M}_6 = \mathcal{X}_8$ be the left preconditioners of system $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$, where $\mathcal{X}_j$ is the approximate inverse of $\mathcal{A}$ obtained by FNS. The spectrum of $\Phi(\mathcal{A}\mathcal{M}_i), i = 1, 2, 3$, $\Phi(\mathcal{M}_j\mathcal{A}), j = 4, 5, 6$ and $\Phi(\mathcal{A})$ are depicted in Figure 2. From this figure, we see that the eigenvalues of the corresponding preconditioned tensor are clustered to 1.

## 6 Numerical examples

In this section, we give some numerical examples to compare the FNS algorithm with the other mentioned iterative methods. All tests were carried out in double precision with a MATLAB code, while the computer specifications are Microsoft Windows 10 Intel(R), Core(TM)i7-7500U, CPU 2.70 GHz, with 8 GB of RAM. All used codes came from the MATLAB tensor toolbox developed by Bader and Kolda [1, 2].

**Example 4.** Consider 10 real random tensors of the form $\mathcal{A} = 33 * tenrand([10\ 9\ 8\ 10\ 9\ 8]) \in \mathbb{R}^{10\times9\times8\times10\times9\times8}$ with $\mathcal{X}_0 = \alpha\mathcal{A}^T$, $\alpha = 1/\|\mathcal{A}\|^2$ and the stopping criterion is $||\mathcal{E}_j||/||\mathcal{E}_0|| < 10^{-10}$, where $\mathcal{E}_j = \mathcal{I} - \mathcal{A} *_N \mathcal{X}_j$. The comparison results of the algorithms are shown in Figure 3. We find from Figure 3 that the FNS method converges to the exact solution in fewer iteration

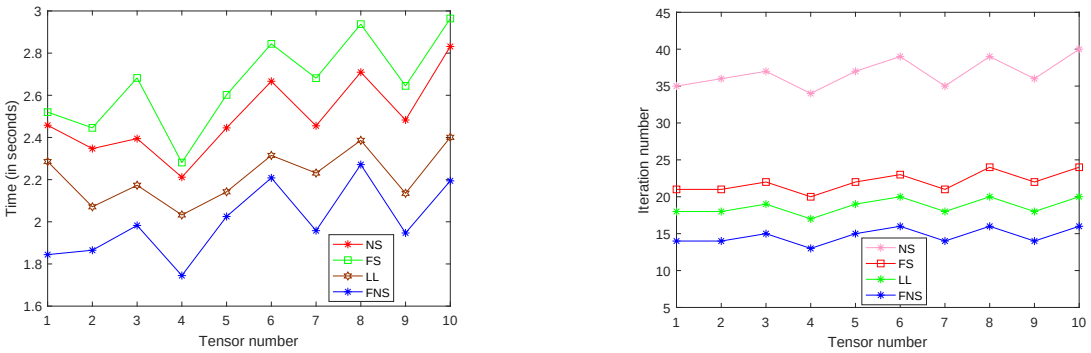


Figure 3: Numerical results for Example 4.

numbers and elapsed CPU time than the other methods. By multiple tests, we also find that this result holds, where re-verified that the FNS method is efficient.

**Example 5.** In this test, we compare the iterative methods for finding the generalized inverse that discussed in Section 4 for 15 real random tensors of the form $\mathcal{A} = 25\times tenrand([10\ 9\ 8\ 7\ 6\ 5]) \in \mathbb{R}^{10\times9\times8\times7\times6\times5}$. We assume that $\mathcal{X}_0 = \alpha\mathcal{A}^T$ with $\alpha = 1/\|\mathcal{A}\|^2$ is the initial tensor and the stopping criterion is

$$\max\{\|\mathcal{X}\mathcal{A} - (\mathcal{X}\mathcal{A})^*\|, \|\mathcal{A}\mathcal{X} - (\mathcal{A}\mathcal{X})^*\|, \|\mathcal{A}\mathcal{X}\mathcal{A} - \mathcal{A}\|, \|\mathcal{X}\mathcal{A}\mathcal{X} - \mathcal{X}\|\} < 10^{-10}.$$

The results of the mentioned algorithms are depicted in Figure 4. As seen in Figure 4, the FNS method converges to the exact solution faster and in a fewer number of iterations than the others. This example further confirms that the FNS method is quite efficient in finding the Moore-Penrose inverse of tensors.

**Example 6.** Finally, let $X, B \in \mathbb{R}^{I_1\times I_2}$. We use the left preconditioned conjugate gradient method for solving the multilinear system $\mathcal{A} *_2 X = B$, where $B = rand(n, n)$, "rand" is the MATLAB function, and $\mathcal{A}$ is real symmetric Toeplitz tensor with $\mathcal{A} = toep(\mathcal{G}) \in \mathbb{R}^{I_1\times I_2\times I_1\times I_2}$ and $\mathcal{G} = g_{k_1,k_2} \in \mathbb{R}^{(2I_1-1)\times(2I_2-1)}$, given by generating sequence

$$g_{k_1,k_2} = \frac{1}{(|k_1|+1)(|k_2|+1)},$$

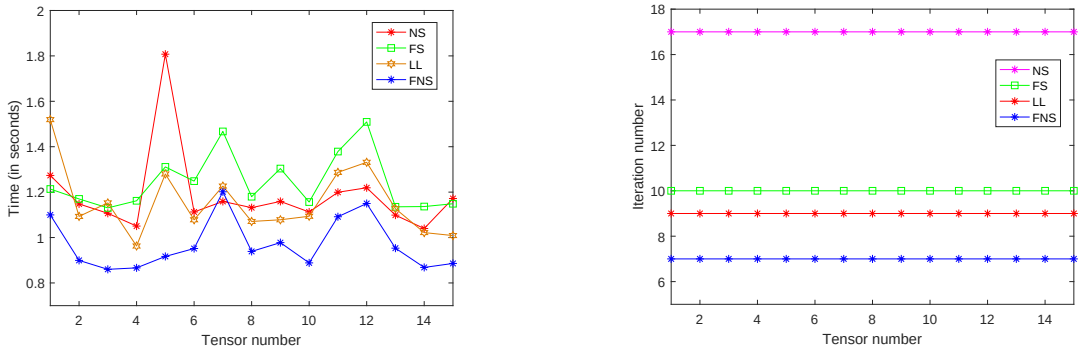


Figure 4: Numerical results for Example 5.

Table 3: The iteration number (Iter) and CPU Time (Time) for the Example 6.

| $I_1 = I_2 = n$ | $n = 10$ | | $n = 15$ | | $n = 20$ | | $n = 25$ | | $n = 30$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | Iter | Time | Iter | Time | Iter | Time | Iter | Time | Iter | Time |
| CG | 222 | 1.3072 | 867 | 1.7465 | 2890 | 9.6315 | 4397 | 18.1790 | 22414 | 140.9580 |
| $\mathcal{M}_1$CG | 8 | 0.1631 | 24 | 0.1771 | 71 | 0.7485 | 72 | 2.1764 | 243 | 6.2495 |
| $\mathcal{M}_2$CG | 2 | 0.1021 | 5 | 0.1541 | 17 | 0.5025 | 13 | 1.4922 | 52 | 3.2779 |
| $\mathcal{M}_3$CG | 2 | 0.0931 | 2 | 0.1542 | 5 | 0.4561 | 3 | 1.2772 | 12 | 3.0101 |

for $1 - I_s \leq k_s \leq I_s - 1, s = 1, 2$. The generating sequence is multivariate generalization of the ones in [11]. In this example, we focus our attention on comparison between conjugate gradient (CG) and preconditioned CG with the preconditioners $\mathcal{X}_8, \mathcal{X}_{10}, \mathcal{X}_{12}$, obtained by the FNS algorithm. In this case, we take $\mathcal{X}_0 = \mathcal{A}/\|\mathcal{A}\|^2$. The stopping criterion is $||E_j|| < 10^{-16}$, where $E_j = I - \mathcal{A} *_2 X_j$. The numerical results are presented in Table 3, where the CPU time (in seconds) and iteration numbers are denoted by Time and Iter, respectively. $\mathcal{M}_1$CG, $\mathcal{M}_2$CG and $\mathcal{M}_3$CG, denote the preconditioned CG algorithm with the preconditioners $X_8, X_{10}$ and $X_{12}$, respectively.

From Table 3, we find that when the grid size is increasing, it needs more computation time and iterative steps to find the approximate solution of the tensor equation $\mathcal{A} *_2 X = B$. Also we observe that the iteration number and elapsed time for obtaining the approximate solution by the preconditioned CG method with the three preconditioners $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ are less than the CG method. Therefore, each iteration obtained by the FNS algorithm can be used as an appropriate preconditioner for solving multilinear system $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$.

## 7 Conclusion

We have presented and analyzed a fast and highly efficient iterative method FNS for computing the approximate inverse of an invertible tensor. It was shown that the proposed method is convergent with the local sixth order of convergence with only five tensor-tensor multiplication

in each step. This makes the method to be interesting for practical problems. Also it was shown that under certain conditions the proposed method is guaranteed to converge to the Moore-Penrose and outer inverse of tensors. We used the each iterate, obtained by the FNS method, as a preconditioner to solve the multilinear system $\mathcal{A} *_N \mathcal{X} = \mathcal{B}$. The numerical results show that the new method is highly efficient.

## References

[1] B.W. Bader, T.G. Kolda, *Efficient Matlab computations with sparse and factored tensors,* SIAM. J. Sci. Comput. **30** (2007) 205–231.

[2] B.W. Bader, T.G. Kolda, *Matlab tensor toolbox,* Version 2.6, Available online at https://www.tensortoolbox.org, 2010.

[3] R. Behera, D. Mishra, *Further results on generalized inverses of tensors via the Einstein product,* Linear Multilinear Algebra **65** (2017) 1662–1682.

[4] F.P.A. Beik, E.l. Ichi, K. Jbilou, R. Sadaka, *Tensor extrapolation methods with applications,* Numer. Algorithms, https://doi.org/10.1007/s11075-020-01013-5, 2020.

[5] F.P.A. Beik, K. Jbilou, M. Najafi-Kalyani, L. Reichel, *Golub-Kahan bidiagonalization for ill-conditioned tensor equations with applications,* Numer. Algorithms **84** (2020) 1535–1563.

[6] F.P.A. Beik, M. Najafi-Kalyani, *A preconditioning technique in conjunction with Krylov subspace methods for solving multilinear systems,* Appl. Math. Lett. **116** (2020) 107051.

[7] F.P.A. Beik, S. Ahmadi-Asl, *Residual norm steepest descent based iterative algorithms for Sylvester tensor equations,* J. Math. Model. **2** (2015) 115–131.

[8] A. Ben-Israel, T.N.E. Greville, *Generalized Inverses,* second ed., Springer, NY, 2003.

[9] M. Brazell, N. Li, C. Navasca, C. Tamon, *Solving multilinear systems via tensor inversion,* SIAM J. Matrix Anal. Appl. **34** (2013) 542–570.

[10] C. Bu, X. Zhang, J. Zhou, et al., *The inverse, rank and product of tensors,* Linear Algebra Appl **446** (2014) 269–280.

[11] R. Chan X. Jin *An Introduction to Iterative Toeplitz Solvers,* Philadelphia: SIAM. 2007.

[12] H. Chen, Y. Wang, *A family of higher-order convergent iterative methods for computing the Moore-Penrose inverse,* Appl. Math. Comput. **218** (2011) 4012–4016.

[13] L. Cui, C. Chen, W. Li, M.K. Ng, *An eigenvalue problem for even order tensors with its applications,* Linear Multilinear Algebra **64** (2016) 602–621.

[14] W. Ding, Y. Wei, *Solving multi-linear systems with $M$-tensors,* J. Sci. Comput. **68** (2016) 689–715.

[15] F. Ding, T. Chen, *Iterative least squares solutions of coupled Sylvester matrix equations,* Syst. Control Lett. **54** (2005) 95–107.

[16] M.P. Drazin, *A class of outer generalized inverses,* Linear Algebra Appl. **436** (2014) 1909–1923.

[17] M. Frontini, F. Sormani, *Some variant of Newtons method with third-order convergence,* Appl. Math. Comput. **140** (2003) 419–426.

[18] B. Huang, C. Ma, *An iterative algorithm to solve the generalized Sylvester tensor equations,* Linear Multilinear Algebra **68** (2020) 1175-1200.

[19] B. Huang, C. Ma, *Global least squares methods based on tensor form to solve a class of generalized Sylvester tensor equations,* Appl. Math. Comput. **369** (2020) 124892.

[20] B. Huang, Y. Xie, C. Ma, *Krylov subspace methods to solve a class of tensor equations via the Einstein product,* Numer. Linear Algebra Appl. **26** (2019) e2254.

[21] J. Ji, Y. Wei, *Weighted Moore-Penrose inverses and fundamental theorem of even-order tensors with Einstein product,* Front Math. China. **12** (2017) 1319–1337.

[22] J. Ji, Y. Wei, *The Drazin inverse of an even-order tensor and its application to singular tensor equations,* Comput. Math. Appl. **75** (2018) 3402–3413.

[23] H. Jin, M. Bai, J. Bentez, et al., *The generalized inverses of tensors and an application to linear models,* Comput. Math. Appl. **74** (2017) 385–397.

[24] E. Khosravi Dehdezi, S. Karimi, *Extended conjugate gradient squared and conjugate residual squared methods for solving the generalized coupled Sylvester tensor equations,* Trans. Inst. Meas. Control. **43** (2021) 519–527.

[25] T.G. Kolda, B.W. Bader, *Tensor decompositions and applications,* SIAM Rev. **51** (2009) 455–500.

[26] W. Lai, D. Rubin, E. Krempl, *Introduction to Continuum Mechanics,* Oxford: Butterworth-Heinemann 2009.

[27] W. Li, Z. Li, *A family of iterative methods for computing the approximate inverse of a square matrix and inner inverse of a non-square matrix,* Appl. Math. Comput. **215** (2010) 3433–3442.

[28] M. Liang, B. Zheng, R. Zhao, *Tensor inversion and its application to the tensor equations with Einstein product,* Linear Multilinear Algebra **67(4)** (2019) 843–870.

[29] H. Ma, N. Li, P.S. Stanimirovic, V.N. Katsikis, *Perturbation theory for Moore-Penrose inverse of tensor via Einstein product,* Comput. Appl. Math. **38** (2019) 111.

[30] M. Najafi-Kalyani, F.P.A. Beik, K. Jbilou, *On global iterative schemes based on Hessenberg process for (ill-posed) Sylvester tensor equations,* J. Comput. Appl. Math. **373** (2020) 112216.

[31] Y. Miao, L. Qi L, Y. Wei, *Generalized tensor function via the tensor singular value decomposition based on the T-product,* Linear Algebra Appl **590** (2020) 258–303.

[32] K. Panigrahy, R. Behera, D. Mishra, *Reverse-order law for the Moore-Penrose inverses of tensors,* Linear Multilinear Algebra **68** (2020) 246–264.

[33] K. Panigrahy, D. Mishra, *Extension the Moore-Penrose inverse of a tensor via the Einstein product,* Linear Multilinear Algebra https://doi.org/10.1080/03081087.2020.1748848, 2020.

[34] K. Panigrahy, D. Mishra, *Reverse order law for weighted Moore-Penrose inverses of tensors,* Adv. Oper. Theory **5** (2020) 39–63.

[35] L. Qi, Z. Luo, *Tensor Analysis: Spectral Theory and Special Tensors,* SIAM, Philadelphia, 2017.

[36] T. Soderstorm, G.W. Stewart, *On the numerical properties of an iterative method for computing the Moore-Penrose generalized inverse,* SIAM J. Numer. Anal. **11** (1974) 61–74.

[37] F. Soleymani, *On finding robust approximate inverses for large sparse matrices,* Linear Multilinear Algebra **62** (2014) 1314-1334.

[38] P.S. Stanimirovic, M. Ciric, V.N. Katsikis, C. Li, H. Ma, *Outer and (b,c) inverses of tensors,* Linear Multilinear Algebra **68** (2020) 940–971.

[39] L. Sun, B. Zheng, C. Bu, Y. Wei, *Moore-Penrose inverse of tensors via Einstein product,* Linear Multilinear Algebra **64** (2016) 686-698.

[40] L. Sun, B. Zheng, Y. Wei, C. Bu, *Generalized inverses of tensors via a general product of tensors,* Front. Math. China. **13** (2018) 893–911.

[41] J.F. Traub, *Iterative Methods for Solution of Equation,* Prentice-Hall, Englewood Cliffs, NJ, 1964.

[42] Q.-W. Wang, X. Xu, *Iterative algorithms for solving some tensor equations,* Linear Multilinear Algebra **67** (2019) 1325–1349.

[43] Z.J. Xie, X.Q. Jin, V.K. Sin, *An optimal preconditioner for tensor equations involving Einstein product,* Linear Multilinear Algebra **68** (2020) 886–902.