Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 20, No. 4, pp. 645–665, 2024

SPARSE LEAST SQUARES SOLUTIONS OF TENSOR EQUATIONS*

Zixin Deng and Zheng-Hai Huang†

Dedicated to Professor Masao Fukushima on the occasion of his 75th birthday

Abstract: In this paper, we investigate the problem of finding a sparse least squares solution to tensor equations, where the tensor involved need not to be a square tensor. When the tensor involved is reduced to a square tensor, the proposed model is reduced to a model recently studied in the literature. By splitting the tensor involved, we propose two linearized methods for solving the problem under consideration, which are extensions of two natural thresholding algorithms proposed recently by Zhao and Luo for solving sparse linear least squares problem. Under proper assumptions, we show that the proposed algorithms are globally linearly convergent. Preliminary numerical results show that the proposed algorithms are effective.

Key words: *tensor equations, sparse tensor least squares problem, linearized method, hard thresholding method, natural thresholding algorithm*

Mathematics Subject Classification: *65K10,15A06,15A69*

1 Introduction

The problem of spare signal recovery has been studied extensively in the past 20 years because of its many practical applications [9, 10, 12, 33]. A fundamental mathematical model for the problem can be described as follows:

$$\min_{x\in\mathbb{R}^n}\{\|Ax-b\|^2 : \|x\|_0 \le k\}, \tag{1.1}$$

where $\|x\|$ means the 2-norm of vector x, $\|x\|_0$ means the number of nonzero entries of the vector x, $A \in \mathbb{R}^{l\times n}$ is a sensing matrix with $l \ll n$, b is the measurement for the signal $x \in \mathbb{R}^n$, and k is a given positive integer. A large number of methods have been designed to solve this problem [9, 10, 12, 33], where the thresholding methods with low computational complexity are particularly suitable for solving it, including the soft thresholding methods [6, 7, 8] and the hard thresholding methods [1, 2, 11, 13, 15, 23, 25]. In the above hard thresholding methods, however, performing hard thresholding does not ensure the reduction of the objective value of (1.1) at every iteration. Recently, Zhao [34] proposed an optimal k-thresholding method and Zhao and Luo [35] proposed a natural thresholding algorithm,

*This work was supported by the National Natural Science Foundation of China under Grant [No. 12371309].

†Corresponding author

DOI: https://doi.org/10.61208/pjo-2023-054

which allow the vector thresholding and objective reduction to be performed simultaneously. Further developments can be found in [26].

In the last ten years, a class of tensor equations (also called multilinear equations) has been studied extensively due to its wide range of applications in engineering and scientific computing such as data mining, numerical partial differential equations, tensor complementarity problems (TCPs) and high-dimensional statistics. Its model can be described as follows:

$$\mathscr{A} x^{m-1} = b, \tag{1.2}$$

where, for given positive integers m and n, $\mathscr{A}$ is an m-th order n-dimensional tensor, i.e., $\mathscr{A} = (a_{i_1 i_2 \cdots i_m})$ with $a_{i_1 i_2 \cdots i_m} \in \mathbb{R}$ for any $i_j \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m\}$, $b \in \mathbb{R}^m$ is a given vector, and $\mathscr{A} x^{m-1} \in \mathbb{R}^m$ with

$$(\mathscr{A} x^{m-1})_i = \sum_{i_2, \ldots, i_m \in \{1,2,\ldots,n\}} a_{i i_2 \cdots i_m} x_{i_2} \cdots x_{i_m}, \quad \forall i \in \{1, 2, \ldots, n\}. \tag{1.3}$$

It is obvious that when $m = 2$, (1.2) reduces to a system of linear equations with the coefficient matrix being a square matrix. A large number of methods have been proposed to solve problem (1.2) [5, 14, 20, 22, 29, 31]. When the tensor $\mathscr{A}$ involved is a Z-type tensor, Luo, Qi and Xiu [24] proved that finding a nonnegative solution of (1.2) is equivalent to finding a solution to the corresponding TCP (see survey papers [18, 19, 27] for TCPs). In this case, several methods for finding the least element solution of the TCP (see, for example, [16, 24, 30, 28]) can be used to find a sparest solution of (1.2), since the least element solution of the TCP is one of sparest solutions of (1.2). More recently, motivated by deeply studies in [32, 38] for sparse nonlinear programming, Li, Luo and Chen [21] proposed a Newton hard-threshold pursuit algorithm for finding a sparse least squares solution of (1.2) and establish its locally quadratic convergence under some regularity conditions.

Let $\mathscr{A}$ be an m-th order $l \times n \times \cdots \times n$-dimensional tensor, i.e., $\mathscr{A} = (a_{i_1 i_2 \cdots i_m})$ with $a_{i_1 i_2 \cdots i_m} \in \mathbb{R}$ for any $i_1 \in \{1, 2, \ldots, l\}$ and $i_j \in \{1, 2, \ldots, n\}$ with $j \in \{2, 3, \ldots, m\}$, and $b \in \mathbb{R}^l$ be a given vector, we consider the following model:

$$\begin{array}{ll} \min & f(x) = \frac{1}{2}\|\mathscr{A} x^{m-1} - b\|^2 \\ \text{s.t.} & \|x\|_0 \leq k, \end{array} \tag{1.4}$$

which is a sparse least squares optimization model for solving tensor equations:

$$\mathscr{A} x^{m-1} = b, \tag{1.5}$$

where $\mathscr{A} x^{m-1} \in \mathbb{R}^l$ with $(\mathscr{A} x^{m-1})_i$ being defined by (1.3) for all $i \in \{1, 2, \ldots, l\}$. When $l = n$, the system of tensor equations (1.5) reduces to the one in (1.2) and the sparse least squares problem (1.4) reduces to the one studied in [21].

In this paper, by splitting the tensor involved, we propose two linearized methods to solve sparse tensor least squares optimization problem (1.4), which are extensions of Zhao-Luo's natural thresholding algorithms [35] to solve the sparse linear least squares problem (1.1). In particular, we show the proposed algorithms are globally linearly convergent under appropriate assumptions. We also report preliminary numerical experiments, which verify the effectiveness of the algorithm numerically.

The rest of this paper is organized as follows. In Section 2, we introduce some symbols, concepts and results which will be used in subsequent analyses. In Section 3, we describe the specific algorithms for solving sparse tensor least squares optimization problem (1.4). In Section 4, we show the theoretical convergence of the proposed methods under proper assumptions. Preliminary numerical results are reported in Section 5, and the concluding remarks are given in Section 6.

2 Preliminaries

Throughout this paper, m, n, l, k are given positive integers with $l \leq n$ and $k \leq n$, and we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. The set of all m-th order $l \times n \times \cdots \times n$-dimensional tensors is denoted by $\mathbb{R}^{l\times[m-1,n]}$ and the set of all m-th order n-dimensional tensors is denoted by $\mathbb{R}^{[m,n]}$. For any $x \in \mathbb{R}^n$, we say that x is k-sparse if $\|x\|_0 \leq k$, and we use $\text{supp}(x)$ to denote the support set of x, i.e., $\text{supp}(x) := \{i \in [n] : x_i \neq 0\}$. For any positive integer $k \leq n$, let $L_k(x)$ denote the index set of k largest entries of x, and $S_k(x)$ denote the index set of k smallest entries of x. When $L_k(x)$ is not uniquely determined, we select the smallest entry indices to avoid the ambiguity in index selection. Similar selection is used for $S_k(x)$ to ensure that it is also well-defined. Given $S \subseteq [n]$, $|S|$ means the cardinality of S. Given $S \subseteq [n]$ and $x \in \mathbb{R}^n$, the vector $x_S \in \mathbb{R}^n$ is obtained from x by retaining the entries supported on S and setting other entries to zeros. For any $x \in \mathbb{R}^n$, we denote $x^{[m-1]} := (x_1^{m-1}, x_2^{m-1}, \ldots, x_n^{m-1})^\top$, and use $\mathcal{H}_k(x)$ to denote the hard thresholding operator defined by

$$\mathcal{H}_k(x)_i = \begin{cases} x_i, & \text{if } i \in L_k(|x|), \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

where $|x| = (|x_1|, |x_2|, \ldots, |x_n|)^\top$. We use '$\otimes$' to represent Hadamard product. For example, for any $u, v \in \mathbb{R}^n$, we have $u \otimes v \in \mathbb{R}^n$ with $(u \otimes v)_i = u_i v_i$ for all $i \in [n]$. We denote $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x \geq 0\}$, and for any $x \in \mathbb{R}^n$, $x_+ := (\max\{0, x_1\}, \ldots, \max\{0, x_n\}) \in \mathbb{R}^n_+$. For any matrix $A \in \mathbb{R}^{l\times n}$, we use $\|A\|$ to denote the spectrum norm of A defined as $\|A\| = \max_{x\in\mathbb{R}^n} \frac{\|Ax\|}{\|x\|}$.

The concepts of the majorization matrix and majorization tensor of an arbitrary square tensor have been extensively used in the literature, see, for example, [37]. The following is a simple extension for a tensor that doesn't have to be square.

Definition 2.1. Let $\mathscr{A} = (a_{i_1 i_2 \cdots i_m}) \in \mathbb{R}^{l\times[m-1,n]}$ with $l \leq n$. A tensor $\mathscr{M} = (\bar{m}_{i_1 i_2 \cdots i_m})$ is called the majorization tensor of $\mathscr{A}$, if for any $i_1 \in [l]$ and $i_2, \ldots, i_m \in [n]$,

$$\bar{m}_{i_1 i_2 \cdots i_m} = \begin{cases} a_{i_1 i_2 \cdots i_m}, & \text{if } i_2 = i_3 = \cdots = i_m, \\ 0, & \text{otherwise}, \end{cases}$$

and the matrix

$$M := (m_{ij}) \in R^{l\times n}, \text{ where } m_{ij} = \bar{m}_{ij\cdots j}, \forall i \in [l], \forall j \in [n] \tag{2.2}$$

is called the majorization matrix of $\mathscr{A}$.

The concept of the binary regularization function can be found in [36, Definition 2.1], which is an effective tool for dealing with binary vectors.

Definition 2.2. A real value function ϕ is called a binary regularization if it satisfies

(i) ϕ is a positive and continuously differentiable function over an open neighborhood of the box $D := \{w \in \mathbb{R}^n : 0 \leq w_i \leq 1, i \in [n]\}$, and

(ii) $\phi(\omega)$ reaches its minimum over D at and only at any binary vectors $w \in \{0, 1\}^n$.

The following condition on the restricted isometry property (RIP condition) can be found in [3, Definition 1.1], which has been extensively used in the theoretical analysis of algorithms for the sparsity-based signal recovery.

Definition 2.3. Given a matrix $A \in \mathbb{R}^{l\times n}$ with $l \leq n$, the k-th restricted isometry constant, denoted by δ_k, is the smallest number $\delta \geq 0$ such that

$$(1-\delta)\|x\|^2 \leq \|Ax\|^2 \leq (1+\delta)\|x\|^2 \tag{2.3}$$

holds for all k-sparse vector $x \in \mathbb{R}^n$.

This following lemma is useful for our theoretical analysis.

Lemma 2.4 ([3, 12])**.** *Let $u \in \mathbb{R}^n$ and $\Lambda \subseteq [n]$, if $|\Lambda \cup \mathrm{supp}(u)| \leq t$, then*

$$\|[(I - A^\top A)u]_\Lambda\| \leq \delta_t \|u\|.$$

3 Algorithm Design

For any given $\mathscr{A} = (a_{i_1 i_2 \cdots i_m}) \in \mathbb{R}^{l\times[m-1,n]}$ with $l \leq n$, let $\mathscr{M}$ and M be the majorization tensor and the majorization matrix of $\mathscr{A}$, respectively (see Definition 2.1). For any $x \in \mathbb{R}^n$, denote $y := x^{[m-1]}$, and let

$$h(x) = \mathscr{A}x^{m-1} - My. \tag{3.1}$$

Before formally presenting the algorithm, we first describe the main idea of the proposed algorithm in the following.

Suppose that we have already reached the point of the p-th step iteration, $x^{(p)}$, and let $y^{(p)} := (x^{(p)})^{[m-1]}$ and

$$\bar{b}_p := -h(x^{(p)}) + b. \tag{3.2}$$

When m is an even number, we consider

$$\begin{array}{ll} \min & f^{(p)}(y) = \frac{1}{2}\|My - \bar{b}_p\|^2 \\ \text{s.t.} & \|y\|_0 \leq k, \end{array} \tag{3.3}$$

and when m is an odd number, we consider

$$\begin{array}{ll} \min & f^{(p)}(y) = \frac{1}{2}\|My - \bar{b}_p\|^2 \\ \text{s.t.} & y \in \mathbb{R}^n_+, \\ & \|y\|_0 \leq k. \end{array} \tag{3.4}$$

We choose a stepsize $\lambda > 0$, and set

$$u^{(p)} := y^{(p)} - \lambda \nabla f^{(p)}(y^{(p)}) = y^{(p)} - \lambda M^\top (My^{(p)} - \bar{b}_p) \tag{3.5}$$

when m is an even number, and set

$$u^{(p)} := \left(y^{(p)} - \lambda \nabla f^{(p)}(y^{(p)})\right)_+ = \left(y^{(p)} - \lambda M^\top (My^{(p)} - \bar{b}_p)\right)_+ \tag{3.6}$$

when m is an odd number. Here, $u^{(p)}$ is a middle vector generated by gradient descent method at the p-th step iteration. In the following, unless otherwise specified, $u^{(p)}$ means the one given by (3.5) when m is an even number, and $u^{(p)}$ means the one given by (3.6) when m is an odd number.

Due to the sparsity of the desired solution, one has to cut off some $n-k$ entries of $u^{(p)}$. In order to find the optimal k entries to reserve, we consider this following 0-1 model which was raised by Zhao in [34]:

$$\begin{array}{ll} \min & \frac{1}{2}\|M(u^{(p)}\otimes w)-\bar{b}_p\|^2 \\ \text{s.t.} & e^\top w = k,\ w\in\{0,1\}^n, \end{array} \tag{3.7}$$

where e is a vector of all ones. However, solving an integer programming problem is expensive. A further approximation of (3.7) is desired. Define $U^p := \mathrm{diag}(u^{(p)})$ (i.e., $(U^p)_{ii} = u_i^{(p)}$ for all $i\in[n]$, and $(U^p)_{ij}=0$ for all $i\in[n]$ with $i\neq j$), and

$$\phi(w) := \left(w+\frac{1}{2}e\right)^\top U^pU^p\left(\frac{3}{2}e-w\right), \quad \forall w\in D:=[0,1]^n. \tag{3.8}$$

Then, $\phi(\cdot)$ is a binary regularization function and it is concave [36].

Let α be a scalar and

$$g_\alpha^p(w) := \frac{1}{2}\|M(u^{(p)}\otimes w)-\bar{b}_p\|_2^2+\alpha\phi(w). \tag{3.9}$$

We consider the following problem:

$$\begin{array}{ll} \min & g_\alpha^p(w) \\ \text{s.t.} & e^\top w = k,\ w\in[0,1]^n. \end{array} \tag{3.10}$$

Lemma 3.1 ([36, Lamma 2.3])**.** *Suppose $w(\alpha_s)$ is the optimal solution of (3.10) where $\alpha=\alpha_s$, then any accumulation point of $\{w(\alpha_s)\}$ is the optimal solution of (3.7) when $\alpha_s\to+\infty$.*

Lemma 3.1 means that if α is large enough, one can get an optimal solution of (3.7) with desired accuracy level by solving (3.10).

Since $u^{(p)}\otimes w = U^pw$, $g_\alpha^p(w)$ can be written as $\frac{1}{2}\|MU^pw-\bar{b}_p\|^2+\alpha\phi(w)$. Then, we have that

$$\nabla g_\alpha^p(w) = (U^pM)^\top\left(MU^pw-\bar{b}_p\right)+2\alpha U^pU^p(e-w)$$

and

$$\nabla^2 g_\alpha^p(w) = (MU^p)^\top MU^p - 2\alpha U^pU^p = U^p\left(M^TM-2\alpha I\right)U^p.$$

Noticed that $\nabla^2 g_\alpha^p(w)$ can be negative semi-definite if one chooses α properly. In this case, $g_\alpha^p(w)$ is concave. Zhao and Luo have further discussed how to choose α in [36]. From now on, suppose that we have already picked an α appropriately such that the function g_α^p defined by (3.9) is concave.

By using first order approximation of $g_\alpha^p(w)$ at some fixed point w^- by the concavity of the function g_α^p, instead of (3.10), we consider

$$\begin{array}{ll} \min & g_\alpha^p(w^-)+[\nabla g_\alpha^p(w^-)]^T(w-w^-) \\ \text{s.t.} & e^\top w = k,\ w\in[0,1]^n. \end{array} \tag{3.11}$$

After removing constant terms, it can be simplified as

$$\begin{array}{ll} \min & [\nabla g_\alpha^p(w^-)]^\top w \\ \text{s.t.} & e^\top w = k,\ w\in[0,1]^n. \end{array} \tag{3.12}$$

An explicit solution of (3.12) can be given by

$$w^+ \in \mathbb{R}^n \quad \text{with} \;\; w_i^+ = \begin{cases} 1, & \text{if } i \in S_k\left(\nabla g_\alpha^p(w^-)\right), \\ 0, & \text{otherwise,} \end{cases} \qquad \forall i \in [n].$$

Now, the algorithm can be formally described as follows:

Algorithm 3.2. Choose $x^{(0)} \in \mathbb{R}^n$ and let $y^{(0)} = (x^{(0)})^{[m-1]}$. Choose a real number $\varepsilon > 0$ and an integer $N_{max} > 0$. Set $p := 0$.
Step 1: If $\|\mathscr{A}(x^{(p)})^{m-1} - b\| < \varepsilon$ or $p > N_{max}$, stop. Otherwise, at $x^{(p)}$, choose $\lambda_p > 0$ and generate $u^{(p)}$ as (3.5) if m is an even number, and as (3.6) if m is an odd number. Let $w^- \in \mathbb{R}^n$ be the k-sparse vector given by

$$w_i^- = \begin{cases} 1, & \text{if } i \in L_k(|u^{(p)}|), \\ 0, & \text{otherwise,} \end{cases} \qquad \forall i \in [n]. \tag{3.13}$$

Compute the gradient $\nabla g_\alpha^p(w^-)$. Then set $w^+ \in \mathbb{R}^n$ as

$$w_i^+ = \begin{cases} 1, & \text{if } i \in S_k\left(\nabla g_\alpha^p(w^-)\right), \\ 0, & \text{otherwise,} \end{cases} \qquad \forall i \in [n]. \tag{3.14}$$

Step 2: Generate y^{p+1} according to the following procedure:
NT: $y^{(p+1)} = u^{(p)} \otimes w^+$.
Step 3: Set

$$x^{(p+1)} = (y^{(p+1)})^{[\frac{1}{m-1}]}. \tag{3.15}$$

Set $p = p + 1$ and go back to **Step 1**.

In the following, we call **Algorithm 3.2** as Algorithm NT. If **Step 2** in Algorithm NT is replaced by
Step 2 $'$: *Generate y^{p+1} according to the following procedure:*

NTP: Set $S^{p+1} = \text{supp}(u^{(p)} \otimes w^+)$, and

$$y^{(p+1)} = \text{argmin}_z \left\{\|\bar{b}_p - Mz\| : \text{supp}(z) \subseteq \mathrm{S}^{p+1}\right\} \tag{3.16}$$

if m is an even number, and set

$$y^{(p+1)} = \text{argmin}_{z \in \mathbb{R}^n_+} \left\{\|\bar{b}_p - Mz\| : \text{supp}(z) \subseteq \mathrm{S}^{p+1}\right\}, \tag{3.17}$$

if m is an odd number.

Then, the corresponding algorithm is called Algorithm NTP.

For convenience, we will denote the set of iterative indices by

$$\mathcal{N} := \{1, 2, \ldots\}.$$

4 Convergence Analysis

In this section, we will investigate the convergence of both Algorithm NT and Algorithm NTP. We will use the following assumption.

Assumption 4.1. Given $\mathscr{A} \in \mathbb{R}^{l\times[m-1,n]}$ and $b \in \mathbb{R}^l$. Let x^* be an optimal solution of (1.4). The function h defined by (3.1) is the sparse $(m-1)$-power Lipschitz continuous with respect to x^*, i.e., there exists a constant $c \geq 0$ such that

$$\|h(x) - h(x^*)\| \leq c\|x^{[m-1]} - (x^*)^{[m-1]}\| \tag{4.1}$$

for any k-sparse vector $x \in \mathbb{R}^n$.

Two similar assumptions were introduced in [4] and [17], respectively, where the concept of the $(m-1)$-*power Lipschitz continuity* introduced in [4] has been extensively used in the literature. Recall that the function h is called the $(m-1)$-power Lipschitz continuous if there exists a constant $\tilde{c} \geq 0$ such that

$$\|h(x) - h(y)\| \leq \tilde{c}\|x^{[m-1]} - y^{[m-1]}\| \tag{4.2}$$

holds for all $x, y \in \mathbb{R}^n$. It is easy to see that if the optimal solution of (1.4) x^* exists, then condition (4.1) is weaker than the requirement that h is the $(m-1)$-power Lipschitz continuous. In fact, on one hand, the $(m-1)$-power Lipschitz continuity of h requires that (4.2) holds for all $x, y \in \mathbb{R}^n$, while one of block variables x and y is fixed as x^* in (4.1); and on the other hand, the sparsity restrict in (4.1) can also reduce the requirement of the corresponding condition. The former is obvious, while the latter can be seen from a simple example in the following remark.

Remark 4.1. For any 1-sparse vector $x \in \mathbb{R}^n$, by the definition of $h(\cdot)$ in (3.1), we always have $h(x) = 0$. Thus, when $k = 1$, for any $\mathscr{A} = (a_{i_1 i_2\cdots i_m}) \in \mathbb{R}^{l\times[m-1,n]}$ and $b \in \mathbb{R}^n$, as long as the optimal solution of (1.4) exists, denoted by x^*, we always have that (4.1) holds for any 1-sparse vector $x \in \mathbb{R}^n$. However, for most tensors $\mathscr{A} = (a_{i_1 i_2\cdots i_m}) \in \mathbb{R}^{l\times[m-1,n]}$, it is difficult to ensure that (4.2) holds for any vectors $x, y \in \mathbb{R}^n$.

In the following remark, we give several examples to illustrate that Assumption 4.1 is satisfied in some cases.

Remark 4.2. (i) Let $l \leq n$. $\mathscr{A} = (a_{i_1 i_2\cdots i_m}) \in \mathbb{R}^{l\times[m-1,n]}$ is called a row diagonal tensor, if its majorization tensor is itself. This is a simple generalization of the related concept for square tensor. For any row diagonal tensor $\mathscr{A}$ and any vector $b \in \mathbb{R}^n$, as long as the optimal solution of (1.4) exists, denoted by x^*, we always have that (4.1) holds for any $x \in \mathbb{R}^n$.

(ii) Suppose $\mathscr{A} \in \mathbb{R}^{6\times[3,10]}$, where $a_{1111} = a_{2333} = a_{3555} = a_{4666} = a_{5777} = a_{6999} = 1$, $a_{1233} = a_{2344} = a_{3455} = a_{4788} = a_{5899} = a_{6,9,10,10} = 5$ and other entries are zero. Let $b = (-8, 0, 0, 1, 0, 0)^\top$, $k = 2$ and $c = 5$. Then $x^* = (-2, 0, 0, 0, 0, 1, 0, 0, 0, 0)^\top$ is a k-sparse solution of $\mathscr{A}x^3 = b$. Since

$$h(x) = (5x_2x_3^2, 5x_3x_4^2, 5x_4x_5^2, 5x_7x_8^2, 5x_8x_9^2, 5x_9x_{10}^2)^\top,$$

one has $h(x^*) = (0, 0, 0, 0, 0, 0)^\top$. For any other k-sparse vector $x \in \mathbb{R}^{10}$, the value of $h(x)$ has seven cases:

(a) $(0, 0, 0, 0, 0, 0)^\top$; (b) $(5x_2x_3^2, 0, 0, 0, 0, 0)^\top$; (c) $(0, 5x_3x_4^2, 0, 0, 0, 0)^\top$;

(d) $(0, 0, 5x_4x_5^2, 0, 0, 0)^\top$; (e) $(0, 0, 0, 5x_7x_8^2, 0, 0)^\top$; (f) $(0, 0, 0, 0, 5x_8x_9^2, 0)^\top$;

(g) $(0, 0, 0, 0, 0, 5x_9x_{10}^2)^\top$.

In the case (a), we have $\|h(x)-h(x^*)\|=0$, and hence, (4.1) holds trivially. For other cases, without loss of generality, we consider the case (b), i.e., $h(x)=[5x_2x_3^2,0,0,0,0,0]^\top$. In this case, it is obvious that $\|x^{[m-1]}-x^{*[m-1]}\|\neq 0$, and

$$\frac{\|h(x)-h(x^*)\|}{\|x^{[m-1]}-x^{*[m-1]}\|}=\frac{5x_2x_3^2}{\sqrt{x_2^6+x_3^6+x^{*6}_1+x^{*6}_6}}<\frac{5x_2x_3^2}{\sqrt{x_2^6+x_3^6}}.$$

If $x_2=x_3$, then one has

$$\frac{5x_2x_3^2}{\sqrt{x_2^6+x_3^6}}=\frac{5}{\sqrt{2}}<c.$$

Otherwise, without loss of generality, we assume $x_2>x_3$, and then

$$\frac{5x_2x_3^2}{\sqrt{x_2^6+x_3^6}}<\frac{5x_2^3}{x_2^3}=c.$$

Thus, $\frac{\|h(x)-h(x^*)\|}{\|x^{[m-1]}-x^{*[m-1]}\|}<c$ for any 2-sparse $x\in\mathbb{R}^{10}$. That is, Assumption 4.1 holds.

(iii) Suppose $\mathscr{A}\in\mathbb{R}^{6\times[5,10]}$, where $a_{111111}=a_{233333}=a_{322222}=a_{466666}=a_{577777}=a_{699999}=1$, $a_{111333}=a_{666999}=3$, $a_{233444}=a_{344555}=a_{588999}=a_{6,9,9,10,10,10}=2$ and other entries are zero. Let $b=(0,0,-32,0,1,0)^\top$, $k=2$ and $c=3$. Then $x^*=(0,-2,0,0,0,1,0,0,0,0)^\top$ is a k-sparse solution of $\mathscr{A}x^5=b$. Similar discussion as done in (ii), we can obtain that Assumption 4.1 holds.

(iv) Suppose $\mathscr{A}\in\mathbb{R}^{6\times[2,10]}$, where $a_{111}=a_{222}=a_{333}=a_{444}=a_{555}=a_{666}=1$, $a_{123}=a_{234}=a_{345}=a_{478}=a_{589}=a_{6,9,10}=5$ and other entries are zero. Let $b=(4,0,0,0,0,1)^\top$, $k=2$ and $c=5$. Then $x^*=(0,\pm 2,0,0,0,0,\pm 1,0,0,0)^\top$ is a k-sparse solution of $\mathscr{A}x^2=b$. Similar discussion as done in (ii), we can obtain that Assumption 4.1 holds.

First, we discuss the convergence of Algorithm NT. For this purpose, we give two useful lemmas.

Lemma 4.3. *For any a, $b\in\mathbb{R}$, one has $|a_+-b_+|\leq|a-b|$.*

Proof. If $a,b\geq 0$, then $a_+=a$ and $b_+=b$, and if $a,b\leq 0$, then $a_+=b_+=0$. Thus, the desired result holds obviously for these two cases. Otherwise, if one of a and b is nonnegative and the other one is negative, without loss of generality, we assume $a\geq 0$ and $b<0$. In this case, one has $|a_+-b_+|=|a|\leq|a|+|b|=|a-b|$. So, the desired result holds. □

By Lemma 4.3, the following result holds obviously.

Corollary 4.4. *For any $u,v\in\mathbb{R}^n$, one has $\|u_+-v_+\|\leq\|u-v\|$.*

Lemma 4.5. *For any $z\in\mathbb{R}^n$ and any k-sparse $x\in\mathbb{R}^n$, let $S:=\mathrm{supp}(x)$ and $S^*:=\mathrm{supp}(\mathcal{H}_k(z))$, then one has*

$$\|x-\mathcal{H}_k(z)\|\leq 2\|(z-x)_{S^*\cup S}\|. \tag{4.3}$$

Proof. From Lemma 4.1 in [35], one has

$$\|x-\mathcal{H}_k(z)\|\leq\|(z-x)_{S^*\cup S}\|+\|(z-x)_{S^*\setminus S}\|.$$

Since $S^*\setminus S\subset S^*\cup S$, it follows that $\|(z-x)_{S^*\setminus S}\|\leq\|(z-x)_{S^*\cup S}\|$. Thus, we have

$$\|x-\mathcal{H}_k(z)\|\leq 2\|(z-x)_{S^*\cup S}\|.$$

□

Now, we discuss the convergence of Algorithm NT.

Theorem 4.6. *Assume that x^* is an optimal solution of (1.4), and $b = \mathscr{A}(x^*)^{m-1} + r$, where r is measurement error. Let M and $h(\cdot)$ be defined by (2.2) and (3.1), respectively, and let M satisfy the RIP condition given in Definition 2.3. Denote $S := \operatorname{supp}(x^*)$ and $y^* := (x^*)^{[m-1]}$. Let the sequence $\{y^{(p)}\}$ be generated by Algorithm NT. Suppose that there exists $c > 0$ such that (4.1) holds for any k-sparse vector $x \in \mathbb{R}^n$. Then for any $\lambda \in (0.5, 1]$, we have*

$$\|y^* - y^{(p)}\| \leq (c_1)^p \|y^* - y^{(0)}\| + \Big(\sum_{i=0}^{p-1} c_1^i c_2\Big)\|r\|, \quad \forall p \in \mathcal{N},$$

where $c_1 = \frac{2\sqrt{1+\delta_{2k}}(\lambda\delta_{3k}+1-\lambda)}{\sqrt{1-\delta_{2k}}} + c\left(\frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}\right)$ *and* $c_2 = \frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}$.

Proof. Denote

$$\bar{b} := -h(x^*) + b. \tag{4.4}$$

Then, we have

$$\bar{b} = \mathscr{A}(x^*)^{m-1} + r - h(x^*) = My^* + r. \tag{4.5}$$

For any fixed iteration index $p \in \mathcal{N}$, let $x^{(p)}$ be the p-th step iteration point generated by Algorithm NT, and let $u^{(p)}$ be obtained by (3.5) when m is an even number, and by (3.6) when m is an odd number. From the definition of w^- in (3.13), it follows that $w^- \otimes u^{(p)} = \mathcal{H}_k(u^{(p)})$. By the definition of ϕ and the k-sparsity of w^+ and w^-, one has $\phi(w^+) = \phi(w^-) = \phi_{min}$. From the optimality of w^+ for (3.11) and the concavity of g_α^p, one has

$$\begin{aligned} g_\alpha^p(w^+) &\leq g_\alpha^p(w^-) + \nabla g_\alpha^p(w^-)^\top (w^+ - w^-) \\ &\leq g_\alpha^p(w^-) + \nabla g_\alpha^p(w^-)^\top (w^- - w^-) \\ &= g_\alpha^p(w^-). \end{aligned}$$

Combining with $\phi(w^+) = \phi(w^-)$, we have

$$\left\|\bar{b}_p - M\left(u^{(p)} \otimes w^+\right)\right\| \leq \left\|\bar{b}_p - M\left(u^{(p)} \otimes w^-\right)\right\|,$$

which, together with $y^{(p+1)} = w^+ \otimes u^{(p)}$, implies that

$$\|\bar{b}_p - My^{(p+1)}\| \leq \|\bar{b}_p - M\mathcal{H}_k(u^{(p)})\|. \tag{4.6}$$

Since both y^* and $y^{(p+1)}$ are k-sparse vectors, it follows that $y^* - y^{(p+1)}$ is a $2k$-sparse vector. By (4.5) and the triangular inequality, we have

$$\begin{aligned} \|\bar{b}_p - My^{(p+1)}\| &= \|\bar{b}_p - \bar{b} + \bar{b} - My^{(p+1)}\| \\ &= \left\|\bar{b}_p - \bar{b} + M\left(y^* - y^{(p+1)}\right) + r\right\| \\ &= \left\|h(x^{(p)}) - h(x^*) + M\left(y^* - y^{(p+1)}\right) + r\right\| \\ &\geq \left\|M\left(y^* - y^{(p+1)}\right)\right\| - \left\|h(x^{(p)}) - h(x^*) + r\right\| \end{aligned}$$

$$\geq \quad \sqrt{1-\delta_{2k}}\left\|y^* - y^{(p+1)}\right\| - \left\|h(x^{(p)}) - h(x^*) + r\right\|, \qquad (4.7)$$

where the second equality follows from (4.5), the third equality holds from (4.4) and (3.2), and the last inequality holds from (2.3). Similarly, we have

$$\begin{aligned}
\|\bar{b}_p - M\mathcal{H}_k(u^{(p)})\| &= \|\bar{b}_p - \bar{b} + \bar{b} - M\mathcal{H}_k(u^{(p)})\| \\
&= \left\|\bar{b}_p - \bar{b} + M\left(y^* - \mathcal{H}_k(u^{(p)})\right) + r\right\| \\
&= \left\|h(x^{(p)}) - h(x^*) + M\left(y^* - \mathcal{H}_k(u^{(p)})\right) + r\right\| \\
&\leq \left\|M\left(y^* - \mathcal{H}_k(u^{(p)})\right)\right\| + \left\|h(x^{(p)}) - h(x^*) + r\right\| \\
&\leq \sqrt{1+\delta_{2k}}\|y^* - \mathcal{H}_k(u^{(p)})\| + \|h(x^{(p)}) - h(x^*) + r\|. \qquad (4.8)
\end{aligned}$$

Combining (4.6) and (4.7) with (4.8), we can obtain that

$$\|y^* - y^{(p+1)}\| \leq \frac{\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|y^* - \mathcal{H}_k(u^{(p)})\| + \frac{2}{\sqrt{1-\delta_{2k}}}\|h(x^{(p)}) - h(x^*) + r\|. \qquad (4.9)$$

Define $S^* := \text{supp}(\mathcal{H}_k(u^{(p)}))$. By (4.3), one has

$$\|y^* - \mathcal{H}_k(u^{(p)})\| \leq 2\|(y^* - u^{(p)})_{S\cup S^*}\|. \qquad (4.10)$$

Since $|(S \cup \text{supp}(y^{(p)})) \cup (S \cup S^*)| = |S \cup \text{supp}(y^{(p)}) \cup S^*| \leq 3k$ holds from the k-sparsity of vectors y^*, $y^{(p)}$ and $\mathcal{H}_k(u^{(p)})$, it follows from Lemma 2.4 that

$$\left\|\left((I - M^\top M)(y^* - y^{(p)})\right)_{S^*\cup S}\right\| \leq \delta_{3k}\|y^* - y^{(p)}\|. \qquad (4.11)$$

In addition,

when m is an even number, by using (3.5), we have

$$\|(y^* - u^{(p)})_{S\cup S^*}\| = \left\|\left(y^* - y^{(p)} + M^\top(My^{(p)} - \bar{b}_p)\right)_{S\cup S^*}\right\|,$$

when m is an odd number, by using (3.5), we have

$$\left\|\left(y^* - u^{(p)}\right)_{S\cup S^*}\right\| = \left\|\left(y^* - \left(y^{(p)} - M^\top(My^{(p)} - \bar{b}_p)\right)_+\right)_{S\cup S^*}\right\|,$$

which, together with Corollary 4.4, implies that

$$\left\|\left(y^* - u^{(p)}\right)_{S\cup S^*}\right\| \leq \left\|\left(y^* - y^{(p)} + M^\top\left(My^{(p)} - \bar{b}_p\right)\right)_{S\cup S^*}\right\|.$$

Thus, whether m is an even number or an odd number, we always have

$$\begin{aligned}
&\left\|\left(y^* - u^{(p)}\right)_{S\cup S^*}\right\| \\
&\leq \left\|\left(y^* - y^{(p)} + \lambda M^\top\left(My^{(p)} - \bar{b}_p\right)\right)_{S\cup S^*}\right\| \\
&= \left\|\left(y^* - y^{(p)} + \lambda M^\top\left(My^{(p)} - \bar{b} + \bar{b} - \bar{b}_p\right)\right)_{S\cup S^*}\right\|
\end{aligned}$$

$$
\begin{aligned}
&= \left\| \left((I-\lambda M^\top M)\left(y^*-y^{(p)}\right)\right)_{S^*\cup S} + \left(\lambda M^\top \left(r+h(x^*)-h(x^{(p)})\right)\right)_{S^*\cup S} \right\| \\
&\le \left\| \left((I-\lambda M^\top M)\left(y^*-y^{(p)}\right)\right)_{S^*\cup S} \right\| + \left\| \lambda M^\top \left(r+h(x^*)-h(x^{(p)})\right) \right\| \\
&= \left\| \left(\lambda\left(I-M^\top M\right)+(1-\lambda)I\left(y^*-y^{(p)}\right)\right)_{S^*\cup S} \right\| + \left\| \lambda M^\top \left(r+h(x^*)-h(x^{(p)})\right) \right\| \\
&\le \|\lambda(I-M^\top M)(y^*-y^{(p)})_{S^*\cup S}+(1-\lambda)(y^*-y^{(p)})_{S^*\cup S}\| + \lambda\|M^\top(r+h(x^*)-h(x^{(p)}))\| \\
&\le (\lambda\delta_{3k}+1-\lambda)\|y^*-y^{(p)}\| + \lambda\left\|M^\top\left(r+h(x^*)-h(x^{(p)})\right)\right\| \\
&\le (\lambda\delta_{3k}+1-\lambda)\|y^*-y^{(p)}\| + \lambda\|M\|\left\|r+h(x^*)-h(x^{(p)})\right\|, \qquad (4.12)
\end{aligned}
$$

where the penultimate inequality holds because of (4.11) and $\lambda \in (0.5, 1]$.

Now, combining (4.9) and (4.10) with (4.12), we can obtain that

$$
\begin{aligned}
\|y^*-y^{(p+1)}\| \le{}& \frac{2\sqrt{1+\delta_{2k}}(\lambda\delta_{3k}+1-\lambda)}{\sqrt{1-\delta_{2k}}}\|y^*-y^{(p)}\| + \left(\frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}\right)\|r\| \\
&+ \left(\frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}\right)\left\|h(x^*)-h(x^{(p)})\right\|
\end{aligned}
$$

This, together with (4.1), implies that

$$
\begin{aligned}
\|y^*-y^{(p+1)}\| \le{}& \left(\frac{2\sqrt{1+\delta_{2k}}(\lambda\delta_{3k}+1-\lambda)}{\sqrt{1-\delta_{2k}}} + c\left(\frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}\right)\right)\left\|y^*-y^{(p)}\right\| \\
&+ \left(\frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}\right)\|r\| \\
:={}& c_1\|y^*-y^{(p)}\| + c_2\|r\| \qquad (4.13)
\end{aligned}
$$

holds for any fixed $p \in \mathcal{N}$.

Finally, by a recursive method, we can obtain from (4.13) that

$$
\|y^*-y^{(p)}\| \le (c_1)^p\|y^*-y^{(0)}\| + \Big(\sum_{i=0}^{p-1} c_1^i c_2\Big)\|r\|, \quad \forall p \in \mathcal{N}.
$$

This completes the proof. ◻

Corollary 4.7. *Suppose that all assumptions in Theorem 4.6 are satisfied, and $r = 0$. For any $\lambda \in (0.5, 1]$, if*

$$
4\lambda^2\delta_{3k}^3 + 4\lambda(2-\lambda)\delta_{3k}^2 + (5-4\lambda^2)\delta_{3k} + 4\lambda^2 - 8\lambda + 3 < 0 \qquad (4.14)
$$

and

$$
c < \frac{\sqrt{1-\delta_{3k}^2} - 2(1+\delta_{3k})(\lambda\delta_{3k}+1-\lambda)}{2\lambda(1+\delta_{3k})\|M\| + 2} \qquad (4.15)
$$

are satisfied, where c is the constant given in (4.1), the sequence $\left\{(x^{(p)})^{[m-1]}\right\}$ generated by Algorithm NT is globally linearly convergent to $(x^*)^{[m-1]}$.

Proof. Since $r = 0$, it follows from Theorem 4.6 that

$$
\|y^*-y^{(p)}\| \le (c_1)^p\|y^*-y^{(0)}\|, \quad \forall p \in \mathcal{N}, \qquad (4.16)
$$

where

$$c_1 = \frac{2\sqrt{1+\delta_{2k}}(\lambda\delta_{3k}+1-\lambda)}{\sqrt{1-\delta_{2k}}} + c\left(\frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}\right). \tag{4.17}$$

Noticed that $y^* = (x^*)^{[m-1]}$ and $y^{(p)} = (x^{(p)})^{[m-1]}$ for all $p \in \mathcal{N}$, in order to ensure that the sequence $\{(x^{(p)})^{[m-1]}\}$ is globally linearly convergent to $(x^*)^{[m-1]}$, we only make sure that the sequence $\{y^{(p)}\}$ is globally linearly convergent to y^*.

It is easy to see from (4.16) that $\{y^{(p)}\}$ is globally linearly convergent to y^* if $c_1 < 1$. To make sure that $c_1 < 1$ holds, from (4.17), it is enough if

$$c\left(\frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}\right) < 1 - \frac{2\sqrt{1+\delta_{2k}}(\lambda\delta_{3k}+1-\lambda)}{\sqrt{1-\delta_{2k}}}.$$

The above inequality holds if

$$\frac{2\sqrt{1+\delta_{2k}}(\lambda\delta_{3k}+1-\lambda)}{\sqrt{1-\delta_{2k}}} < 1 \tag{4.18}$$

and

$$c < \frac{\sqrt{1-\delta_{2k}^2} - 2(1+\delta_{2k})(\lambda\delta_{3k}+1-\lambda)}{2\lambda(1+\delta_{2k})\|M\|+2} \tag{4.19}$$

hold. For any $a \in [0,1)$, we denote $\kappa := 2\lambda\|M\| > 0$, and define $\varphi_1(a) := \sqrt{1+a}$, $\varphi_2(a) := \frac{1}{\sqrt{1-a}}$, $\varphi_3(a) := \sqrt{1-a^2}$ and $\varphi_4(a) := \frac{1}{\kappa(1+a)+2}$. Observe that $\varphi_1(a)$ and $\varphi_2(a)$ are increasing in $[0,1)$, then we have $\frac{\sqrt{1+a}}{\sqrt{1-a}} = \varphi_1(a)\varphi_2(a)$ is nonnegative and increasing in $[0,1)$. Similarly, since $\varphi_3(a)$ and $\varphi_4(a)$ are nonnegative and decreasing in $[0,1)$, it follows that $\frac{\sqrt{1-a^2}}{\kappa(1+a)+2} = \varphi_3(a)\varphi_4(a)$ is decreasing in $[0,1)$. Based on the above observation, by using $0 \le \delta_{2k} < \delta_{3k} \le 1$, we obtain that (4.18) holds if

$$\frac{2\sqrt{1+\delta_{3k}}(\lambda\delta_{3k}+1-\lambda)}{\sqrt{1-\delta_{3k}}} < 1 \tag{4.20}$$

holds, and (4.19) holds if

$$c < \frac{\sqrt{1-\delta_{3k}^2} - 2(1+\delta_{3k})(\lambda\delta_{3k}+1-\lambda)}{2(1+\delta_{3k})\|M\|+2} \tag{4.21}$$

holds. Furthermore, (4.20) holds if

$$4\lambda^2\delta_{3k}^3 + 4\lambda(2-\lambda)\delta_{3k}^2 + (5-4\lambda^2)\delta_{3k} + 4\lambda^2 - 8\lambda + 3 < 0,$$

which is equivalent to condition (4.14), and (4.21) is equivalent to condition (4.15). Note that (4.14) requires that $4\lambda^2 - 8\lambda + 3 < 0$, which holds if $0.5 < \lambda \le 1$. Thus, if the conditions (4.14) and (4.15) hold, then the sequence $\{(x^{(p)})^{[m-1]}\}$ generated by Algorithm NT is globally linearly convergent to $(x^*)^{[m-1]}$. That is, the result of the corollary holds. $\square$

Remark 4.8. (i) Although the condition (4.14) may seem complicated, it is easy to verify. For example, when $\lambda = 1$, if $\delta_{3k} < 0.34$, one has (4.14) holds. (ii) In Corollary 4.7, the condition $r = 0$ means that x^* is a k-sparse solution of $\mathscr{A}x^{m-1} = b$.

Remark 4.9. In Theorem 4.6 and Corollary 4.7, we require $\lambda \in (0.5, 1]$. In fact, these two results hold when $\lambda \in (0.5, 1.5)$. However, if $\lambda > 1$, the penultimate inequality in (4.12) turns to

$$\begin{array}{ll} \cdots & \\ \leq & (\lambda\delta_{3k} + \lambda - 1)\left\|y^* - y^{(p)}\right\| + \lambda\left\|M^\top\left(r + h(x^*) - h(x^{(p)})\right)\right\| \\ \cdots & , \end{array}$$

and hence, c_1 turns to $\frac{2\sqrt{1+\delta_{2k}}(\lambda\delta_{3k}+\lambda-1)}{\sqrt{1-\delta_{2k}}} + c\left(\frac{2\lambda\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|M\| + \frac{2}{\sqrt{1+\delta_{2k}}}\right)$. Accordingly, (4.14) turns to

$$4\lambda^2\delta_{3k}^3 + 4\lambda(3\lambda - 2)\delta_{3k}^2 + (12\lambda^2 - 16\lambda + 5)\delta_{3k} + 4\lambda^2 - 8\lambda + 3 < 0.$$

In this case, the requirement of δ_{3k} would be too strong. For example, when $\lambda = 1.2$, the convergence property holds if $\delta_{3k} < 0.15$. Such a requirement for δ_{3k} is stronger than the condition $\delta_{3k} < 0.34$ in the case of $\lambda = 1$. So, all things considered, we restrict $\lambda \in (0.5, 1]$ in Theorem 4.6 and Corollary 4.7.

Second, we show the convergence of Algorithm NTP. Since the next iteration point of Algorithm NTP generated by a projection map, we give this following lemma before discussing the convergence of Algorithm NTP.

Lemma 4.10. *Assume that x^* is an optimal solution of (1.4), and $b = \mathscr{A}(x^*)^{m-1} + r$, where r is measurement error. Denote $y^* := (x^*)^{[m-1]}$. Suppose that M is defined by (2.2) and it satisfies the RIP condition given in Definition 2.3. Let $x^{(p)}$ be the p-th step iteration point generated by Algorithm NTP and $y^{(p)} := (x^{(p)})^{[m-1]}$. Suppose that there exists $c > 0$ such that (4.1) holds for any k-sparse vector $x \in \mathbb{R}^n$, and $\bar{b}_p = b - h(x^{(p)})$ with $h(\cdot)$ being defined (3.1). Then, for any k-sparse vector $u \in \mathbb{R}^n$, the optimal solution*

$$z^* = \text{argmin}_z \left\{\|\bar{b}_p - Mz\|^2 : \ \text{supp}(z) \subseteq \text{supp}(u)\right\} \tag{4.22}$$

satisfies that

$$\|z^* - y^*\| \leq \frac{\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|y^* - u\| + \frac{2c}{\sqrt{1-\delta_{2k}}}\|y^{(p)} - y^*\| + \frac{2}{\sqrt{1-\delta_{2k}}}\|r\|. \tag{4.23}$$

Proof. On one hand, since $y^* - z^*$ is $2k$-sparse and M satisfies the RIP condition, one has

$$\|My^* - Mz^*\| = \|M\,(y^* - z^*)\| \geq \sqrt{1-\delta_{2k}}\,\|z^* - y^*\|\,. \tag{4.24}$$

On the other hand, one has

$$\begin{aligned} \|My^* - Mz^*\| &= \|Mz^* - (\bar{b} - r)\| \\ &= \|Mz^* - \bar{b}_p + (\bar{b}_p - \bar{b}) + r\| \\ &= \|Mz^* - \bar{b}_p + h(x^*) - h(x^{(p)} + r\| \\ &\leq \|Mu - \bar{b}_p\| + \|h(x^*) - h(x^{(p)}\| + \|r\| \\ &\leq \|Mu - \bar{b}_p\| + c\|y^{(p)} - y^*\| + \|r\|, \end{aligned} \tag{4.25}$$

where the first equality follows from (4.5), the third equality follows from (3.2) and (4.4), the first inequality follows from (4.22), and the second inequality follows from Assumption 4.1. Besides, one can also have

$$\|Mu - \bar{b}_p\| \quad = \quad \|Mu - \bar{b} + \bar{b} - \bar{b}_p\|$$

$$\begin{aligned} &= \|Mu - (My^* + r) + (h(x^*) - h(x^{(p)})\| \\ &\leq \sqrt{1+\delta_{2k}}\|u - y^*\| + c\|y^{(p)} - y^*\| + \|r\|. \end{aligned} \quad (4.26)$$

Combining (4.24) and (4.25) with (4.26), one has

$$\sqrt{1-\delta_{2k}}\|z^* - y^*\| \leq \sqrt{1+\delta_{2k}}\|u - y^*\| + 2c\|y^{(p)} - y^*\| + 2\|r\|,$$

which is equivalent to (4.23). □

Remark 4.11. In Lemma 4.10, if (4.22) is replaced by

$$z^* = \operatorname{argmin}_{z \in \mathbb{R}^n_+} \left\{ \|\bar{b}_p - Mz\|^2 : \operatorname{supp}(z) \subseteq \operatorname{supp}(u) \right\},$$

then it follows from the proof of Lemma 4.10 that the inequality (4.23) is still satisfied.

Theorem 4.12. *Suppose that all assumptions in Lemma 4.10 are satisfied. Let* $S := \operatorname{supp}(x^*)$ *and the sequence* $\{y^{(p)}\}$ *be generated by Algorithm NTP. Then for any* $\lambda \in (0.5, 1]$*, we have*

$$\|y^* - y^{(p)}\| \leq (\tilde{c}_1)^p \|y^* - y^{(0)}\| + \left(\sum_{i=0}^{p-1} \tilde{c}_1^i \tilde{c}_2\right)\|r\|, \quad \forall p \in \mathcal{N},$$

where $\tilde{c}_1 = \frac{2(\lambda\delta_{3k}+1-\lambda)(1+\delta_{2k})}{1-\delta_{2k}} + 2c\left(\frac{2}{\sqrt{1-\delta_{2k}}} + \frac{1+\delta_{2k}}{1-\delta_{2k}}\lambda\|M\|\right)$ *and* $\tilde{c}_2 = \frac{4}{\sqrt{1-\delta_{2k}}} + \frac{2(1+\delta_{2k})}{1-\delta_{2k}}\lambda\|M\|$.

Proof. For any fixed iteration index $p \in \mathcal{N}$, let $\hat{y} := u^{(p)} \otimes w^+$, and $y^{(p+1)}$ be the solution of the orthogonal projection (3.16) when m is an even number, and of the orthogonal projection (3.17) when m is an odd number. Then, by Lemma 4.10 and Remark 4.11, we have

$$\|y^{(p+1)} - y^*\| \leq \frac{\sqrt{1+\delta_{2k}}}{\sqrt{1-\delta_{2k}}}\|\hat{y} - y^*\| + \frac{2c}{\sqrt{1-\delta_{2k}}}\|y^{(p)} - y^*\| + \frac{2}{\sqrt{1-\delta_{2k}}}\|r\|. \quad (4.27)$$

It can be seen that the intermediate point $\hat{y}$ generated at the p-th step iteration of Algorithm NTP has exactly the same form as $y^{(p)}$ generated at the p-th step iteration of Algorithm NT. Thus, similar to the proof of Theorem 4.6, we can obtain that for any fixed iteration index $p \in \mathcal{N}$,

$$\|y^* - \hat{y}\| \leq c_1\|y^* - y^{(p)}\| + c_2\|r\|, \quad (4.28)$$

where c_1, c_2 are given in Theorem 4.6.

Combining (4.27) with (4.28), we can get that

$$\|y^{(p+1)} - y^*\| \leq \tilde{c}_1\|y^* - y^{(p)}\| + \tilde{c}_2\|r\|,$$

where

$$\tilde{c}_1 = \frac{c_1\sqrt{1+\delta_{2k}} + 2c}{\sqrt{1-\delta_{2k}}} = \frac{2(\lambda\delta_{3k}+1-\lambda)(1+\delta_{2k})}{1-\delta_{2k}} + 2c\left(\frac{2}{\sqrt{1-\delta_{2k}}} + \frac{1+\delta_{2k}}{1-\delta_{2k}}\lambda\|M\|\right).$$

and

$$\tilde{c}_2 = \frac{c_2\sqrt{1+\delta_{2k}} + 2}{\sqrt{1-\delta_{2k}}} = \frac{4}{\sqrt{1-\delta_{2k}}} + \frac{2(1+\delta_{2k})}{1-\delta_{2k}}\lambda\|M\|.$$

Thus, by a recursive method, we can obtain that the result of the theorem holds. □

Corollary 4.13. *Suppose that all assumptions in Theorem 4.12 hold, and $r = 0$. If*

$$2\lambda\delta_{3k}^2 + 3\delta_{3k} + 1 - 2\lambda < 0, \tag{4.29}$$

and

$$c < \frac{1 - \delta_{3k}^2 - 2(\lambda\delta_{3k} + 1 - \lambda)(1 + \delta_{3k})}{4\sqrt{1 - \delta_{3k}} + 2\lambda(1 + \delta_{3k})\|M\|} \tag{4.30}$$

are satisfied, then the sequence $\{(x^{(p)})^{[m-1]}\}$ generated by Algorithm NTP is globally linearly convergent to $(x^*)^{[m-1]}$.

Proof. Since $r = 0$, it follows from Theorem 4.12 that

$$\|y^* - y^{(p)}\| \leq (\tilde{c}_1)^p \|y^* - y^{(0)}\|, \quad \forall p \in \mathcal{N}.$$

In order to show that $\{(x^{(p)})^{[m-1]}\}$ is globally linearly convergent to $(x^*)^{[m-1]}$, it is enough to ensure $\tilde{c}_1 < 1$, i.e.,

$$2c\left(\frac{2}{\sqrt{1 - \delta_{2k}}} + \frac{1 + \delta_{2k}}{1 - \delta_{2k}}\lambda\|M\|\right) < 1 - \frac{2(\lambda\delta_{3k} + 1 - \lambda)(1 + \delta_{2k})}{1 - \delta_{2k}}.$$

Since $0 \leq \delta_{2k} < \delta_{3k}$, the above inequality holds if

$$2c\left(\frac{2}{\sqrt{1 - \delta_{3k}}} + \frac{1 + \delta_{3k}}{1 - \delta_{3k}}\lambda\|M\|\right) < 1 - \frac{2(\lambda\delta_{3k} + 1 - \lambda)(1 + \delta_{3k})}{1 - \delta_{3k}}.$$

This inequality holds if δ_{3k} and c satisfy

$$\frac{2(\lambda\delta_{3k} + 1 - \lambda)(1 + \delta_{3k})}{1 - \delta_{3k}} < 1 \quad \text{and} \quad c < \frac{1 - \delta_{3k}^2 - 2(\lambda\delta_{3k} + 1 - \lambda)(1 + \delta_{3k})}{4\sqrt{1 - \delta_{3k}} + 2\lambda(1 + \delta_{3k})\|M\|}. \tag{4.31}$$

The first inequality in (4.31) holds if

$$2\lambda\delta_{3k}^2 + 3\delta_{3k} + 1 - 2\lambda < 0,$$

which is equivalent to (4.29). In addition, it is easy to check that the second inequality in (4.31) is equivalent to inequality of (4.30). Thus, the desired result holds if (4.29) and (4.30) are satisfied. □

Remark 4.14. It is easy to verify the condition (4.29). For example, when $\lambda = 0.8$, (4.29) holds if $\delta_{3k} < 0.18$; when $\lambda = 0.9$, (4.29) holds if $\delta_{3k} < 0.23$; and when $\lambda = 1$, (4.29) holds if $\delta_{3k} < 0.28$. We use the result of $\lambda = 1$ later in the numerical experiments.

5 Numerical Experiments

In this section, we perform numerical experiments on a laptop (Windows 10, 64-bit, 8042 MB physical memory, Intel(R) Core (TM) i5-6200U CPU @ 2.30GHZ) by MATLAB(R2016a).

We only consider the numerical experiments for Algorithm NTP. In our experiments, th parameters are set as follows:

$$\text{tol} = 1e - 6, \quad \text{MaxIter} = 150, \quad \alpha = 3, \quad \lambda = 1.$$

We will display the iteration number, residual error and CPU time, where the residual error is defined as $\|\mathscr{A}x^{m-1} - b\|$ with x being the final output solution. We will divide our experiments into the following two parts.

Part 1. In this part, we test three examples, where $\mathscr{A} \in \mathbb{R}^{l\times[m-1,n]}$ and $b \in \mathbb{R}^l$ involved in (1.4) are given deterministically.

Example 5.1. Consider problem (1.4), where $\mathscr{A} \in \mathbb{R}^{6\times[3,10]}$ and $b \in \mathbb{R}^6$ are given as those in (ii) of Remark 4.2, and $k = 2$.

We use Algorithm NTP to test the problem in Example 5.1, and the numerical results are displayed in Table 1, where 'iniP', 'time', 'niter', 'res' and 'solu' denote the initial point, CPU time, iteration number, residual error and final output x, respectively; and $x_0^1 = (1,1,1,1,1,1,1,1,1,1)^\top$, $x_0^2 = (1,1,1,1,1,0,0,0,0,0)^\top$, $x_0^3 = (0,0,0,0,0,1,1,1,1,1)^\top$, $x_0^4 = (1,0,1,0,1,0,1,0,1,0)^\top$, x_0^5 =randn(10,1) (i.e., x_0^5 is generated randomly). For the case of the initial point being x_0^5, we test the problem 10 times, where we test the problem once for each randomly generated initial point x_0^5, and in this case, 'time', 'niter', 'res' denote the average CPU time, average iteration number and average residual error for 10 times testings.

Table 1: Numerical results of Example 5.1

iniP	time	niter	res	solu
x_0^1	0.0156	3	0	$(-2,0,0,0,0,1,0,0,0,0)^\top$
x_0^2	0.0938	3	0	$(-2,0,0,0,0,1,0,0,0,0)^\top$
x_0^3	0.0156	1	0	$(-2,0,0,0,0,1,0,0,0,0)^\top$
x_0^4	0.0156	3	0	$(-2,0,0,0,0,1,0,0,0,0)^\top$
x_0^5	0.0219	2.4	0	$(-2,0,0,0,0,1,0,0,0,0)^\top$

Example 5.2. Consider problem (1.4), where $\mathscr{A} \in \mathbb{R}^{6\times[3,10]}$ and $b \in \mathbb{R}^6$ are given as those in (iii) of Remark 4.2, and $k = 2$.

We use Algorithm NTP to test the problem in Example 5.2, and the numerical results are displayed in Table 2, where 'iniP', 'time', 'niter', 'res', 'solu', and the initial points $x_0^1, x_0^2, x_0^3, x_0^4, x_0^5$ are the same as those in Example 5.1.

Table 2: Numerical results of Example 5.2

iniP	time	niter	res	solu
x_0^1	0.0625	3	0	$(0,-2,0,0,0,0,1,0,0,0)^\top$
x_0^2	0.0313	3	0	$(0,-2,0,0,0,0,1,0,0,0)^\top$
x_0^3	0.0313	3	0	$(0,-2,0,0,0,0,1,0,0,0)^\top$
x_0^4	0.0313	3	0	$(0,-2,0,0,0,0,1,0,0,0)^\top$
x_0^5	0.0688	2.6	0	$(0,-2,0,0,0,0,1,0,0,0)^\top$

Example 5.3. Consider problem (1.4), where $\mathscr{A} \in \mathbb{R}^{6\times[2,10]}$ and $b \in \mathbb{R}^6$ are given as those in (iv) of Remark 4.2, and $k = 2$.

We use Algorithm NTP to test the problem in Example 5.3, and the numerical results are displayed in Table 3, where 'iniP', 'time', 'niter', 'res', 'solu', and the initial points $x_0^1, x_0^2, x_0^3, x_0^4, x_0^5$ are the same as those in Example 5.1.

From Tables 1, 2 and 3, one can see that Algorithm NTP can recover the desired sparse solution completely with low CPU time cost and few iteration steps for three specific problems given in Examples 5.1, 5.2 and 5.3.

Part 2. In this part, we test two examples, where $\mathscr{A} \in \mathbb{R}^{l\times[m-1,n]}$ and $b \in \mathbb{R}^l$ involved in (1.4) are generated randomly.

Table 3: Numerical results of Example 5.3

iniP	time	niter	res	solu
x_0^1	0.0625	2	0	$(2,0,0,0,0,1,0,0,0,0)^\top$
x_0^2	0.0625	2	0	$(2,0,0,0,0,1,0,0,0,0)^\top$
x_0^3	0.0156	2	0	$(2,0,0,0,0,1,0,0,0,0)^\top$
x_0^4	0.0313	1	0	$(2,0,0,0,0,1,0,0,0,0)^\top$
x_0^5	0.0156	2	0	$(2,0,0,0,0,1,0,0,0,0)^\top$

Example 5.4. Consider problem (1.4), where, for given positive integers m, n, and l, $\mathscr{A} \in \mathbb{R}^{l\times[m-1,n]}$, $b \in \mathbb{R}^l$, and the sparsity k are given as follows:

- Tensor $\mathscr{A}$: take $m = 4$, $l = 40$ and $n = 80$. Let U and V be $l \times l$ and $n \times n$ orthogonal matrices, respectively, and they are generated randomly in $[0, 1]$. Let $\Sigma := \big(D_{l\times l}, O_{l\times(n-l)}\big)$ be an $l \times n$ real matrix, where $D_{l\times l}$ and $O_{l\times(n-l)}$ denote $l \times l$ diagonal matrix and $l \times (n - l)$ zero matrix, respectively, and each entry of D is generated randomly in $[0.9, 1.1]$. Denote $M := U\Sigma V$. Suppose M is the majorization matrix of tensor $\mathscr{M} \in \mathbb{R}^{l\times[m-1,n]}$ and other entries of $\mathscr{M}$ are zero. Let $\mathscr{A} = \mathscr{M}$.
- Sparsity k: we choose the sparsity k such that $k = \lceil \rho n \rceil$, where $\lceil a \rceil$ represents the smallest integer greater than or equal to a, and $\rho \in [0.02, 0.12]$.
- Vector b: let $x^* = \text{zeros}(n, 1)$ and $x^*(k+1 : 2{*}k) = \text{randn}(k, 1)$, and let $b := \mathscr{A}(x^*)^{m-1}$.

Remark 5.5. We claim that Assumption 4.1 and the RIP condition given in Definition 2.3 are satisfied for each problem in Example 5.4. It is obvious that the former holds since $h(x) = 0$ for any $x \in \mathbb{R}^n$. We explain below why the latter is also true.

In order to satisfy the restriction of the RIC δ_{3k} of the majorization matrix M of $\mathscr{A}$ given in (4.29), we have to put some requirements on M. Let d_i denote the i-th diagonal entry of matrix D for any $i \in [n]$. Then, for any $z \in \mathbb{R}^n$, one has

$$\|Mz\|^2 = z^\top V^\top \Sigma^\top U^\top U\Sigma V z = \sum_{i=1}^n d_i^2 z_i^2.$$

Let d_{max} and d_{min} denote the maximal diagonal entry and the minimal diagonal entry of D, respectively. Then,

$$d_{min}^2\|z\|^2 \le \|Mz\|^2 \le d_{max}^2\|z\|^2, \quad \forall z \in \mathbb{R}^n.$$

In addition, if z is k-sparse, then, according to RIP condition, one has

$$(1 - \delta_k)\|z\|^2 \le \|Mz\|^2 \le (1 + \delta_k)\|z\|^2.$$

Since $0 \le \delta_k < \delta_{3k}$, the above inequality holds if

$$(1 - \delta_{3k})\|z\|^2 \le \|Mz\|^2 \le (1 + \delta_{3k})\|z\|^2.$$

Thus, in order to ensure that the RIP condition holds, it is enough if we choose d_i for all $i \in [n]$ such that

$$d_{min}^2 = 1 - \delta_{3k} \quad \text{and} \quad d_{max}^2 = 1 + \delta_{3k}.$$

By $\delta_{3k} < 0.28$ given in (4.29), we only need to choose d_i for all $i \in [n]$ such that

$$d_{min} \geq \sqrt{0.72} \quad \text{and} \quad d_{max} \leq \sqrt{1.28}.$$

In actual experiments, we will set the range of diagonal entries of D in a little bit smaller than $\left[\sqrt{0.72}, \sqrt{1.28}\right]$. This can be ensured by setting each diagonal entry of D be in $[0.9, 1.1]$.

We use Algorithm NTP to test problems in Example 5.4. For each problem (i.e., the case given a set of values m, l, n, k), we test 10 times, and display the numerical results in Table 4. For every test, the initial point is chosen as $x_0 = \text{zeros}(n, 1)$ and $x_0(1:k) = \text{randn}(k, 1)$. In Table 4, 'Atime', 'Aniter', and 'Ares' represent the average CPU time, average iteration number, and average residual error of 10 times tests, respectively.

Table 4: Numerical results of Example 5.4

k	Anitr	Atime	Ares
$\lceil 0.12n \rceil$	7	0.1156	$8.9912e-08$
$\lceil 0.1n \rceil$	5.8	0.0750	$1.9288e-09$
$\lceil 0.07n \rceil$	5.3	0.0656	$2.2543e-10$
$\lceil 0.05n \rceil$	4.7	0.0578	$1.0363e-15$
$\lceil 0.02n \rceil$	3.6	0.0578	$6.0177e-16$

Example 5.6. Consider problem (1.4), where, for given positive integers m, n, and l, $\mathscr{A} = \mathscr{M} - \mu\mathscr{B}$ with $\mathscr{M}$ being generated in the same way as the one in Example 5.4, $\mu = 0.001$ and all entries of $\mathscr{B}$ being generated randomly in $[0, 1]$, and $b \in \mathbb{R}^l$ is generated in the same way as the one in Example 5.4. For different values of m, l and n in our experiments, we choose $k = \lceil 0.05n \rceil$ or $\lceil 0.1n \rceil$.

We use Algorithm NTP to test problems in Example 5.6. We test each problem 10 times with the initial points being generated in the same way as those in Example 5.4, and display the numerical results in Table 5, where 'Atime', 'Aniter', and 'Ares' are the same as those in Example 5.4.

Table 5: Numerical results of Example 5.6

m	l	n	k	Anitr	Atime	Ares
4	4	5	1	1.3	0.0187	$2.9212e-17$
4	8	10	1	1.4	0.0297	$7.5876e-16$
4	12	15	1	1.4	0.0234	$1.4239e-16$
4	12	15	2	3.5	0.0328	$1.9781e-07$
6	8	10	1	1.6	0.0391	$1.0858e-09$

From Tables 4 and 5, one can see that Algorithm NTP can recover the desired sparse solution completely with low CPU time cost and few iteration steps for randomly generated problems given in Examples 5.4 and 5.6.

6 Concluding Remarks

In this paper, we proposed a model of sparse least squares problem for solving tensor equations and two linearized methods (i.e., Algorithm NT and Algorithm NTP) for solving it.

We showed the global linear convergence of the proposed method under two assumptions. The first assumption is that the majorization matrix of the tensor involved satisfies the RIP condition, which is commonly used in the field of sparse optimization. The second assumption is that the function h defined by (3.1) is the sparse $(m-1)$-power Lipschitz continuous with respect to a solution of the concerned tensor equations, which is new, but trivially true when the model we consider returns to the case of compressed sensing. The latter assumption plays a key role in obtaining the convergence of the proposed algorithm. How to weaken this condition is a topic worthy of further study. In addition, in the existing studies of tensor equations and their least squares problems, the tensors involved are assumed to be square tensors, which greatly limits their applications. The sparse tensor least squares problem proposed in this paper is a generalization of the classical compressed sensing model. Since the classical compressed sensing model has many practical applications, we believe that this new model will have many practical applications. Therefore, the effective algorithms to solve this model and its practical application problems are worthy of further study.

References

[1] T. Blumensath and M. Davies, Iterative hard thresholding for sparse approximation, *J. Fourier Anal. Appl.* 14 (2008) 629–654.

[2] T. Blumensath and M. Davies, Normalized iterative hard thresholding: Guaranteed stability and performance, *IEEE J. Sel. Top. Signal Process.* 4 (2010) 298–309.

[3] E.J. Candes and T. Tao, Decoding by linear programming, *IEEE Trans. Inf. Theory* 51 (2005) 4203–4215.

[4] P.F. Dai. A fixed point iterative method for tensor complementarity problems, *J. Sci. Comput.* 84 (2020): 49

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

[6] D.L. Donoho, De-noising by soft-thresholdinng, *IEEE Trans. Inf. Theory* 41 (1995) 613–627.

[7] D.L. Donoho and I. Johnstone, Ideal spatial adaptation by wavelet shrinkage, *Biomatrika* 81 (1994) 425–455.

[8] M. Elad, Why simple shrinkage is still relevant for redundant representations? *IEEE Trans. Inf. Theory* 52 (2006) 5559–5569.

[9] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, NY, 2010.

[10] Y.C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, Cambridge, UK, 2012.

[11] M. Fornasier and R. Rauhut, Iterative thresholding algorithms, *Appl. Comput. Harmon. Anal.* 25 (2008) 187–208.

[12] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Springer, NY, 2013.

[13] S. Foucart and S. Subramanian, Iterative hard thresholding for low-rank recovery from rank-one projections, *Linear Algebra Appl.* 572 (2019) 117–134.

[14] L. Han, A homotopy method for solving multilinear systems with M-tensors, *Appl. Math. Lett.* 69 (2017) 49–54.

[15] K. Herrity, A. Gilbert and J. Tropp, Sparse approximation via iterative thresholding, in: *IEEE ICASSP*, 2006, pp. 624–627.

[16] Z.H. Huang, Y.F. Li, and X.H. Miao, Finding the least element of a nonnegative solution set of a class of polynomial inequalities, *SIAM J. Matrix Anal. Appl.* 44 (2023) 530–558.

[17] Z.H. Huang, Y.F. Li, and Y. Wang, A fixed point iterative method for tensor complementarity problems with the implicit Z-tensors, *J. Global Optim.* 86 (2023) 495–520.

[18] Z.H. Huang and L. Qi, Tensor complementarity problems part I: basic theory, *J. Optim. Theory Appl.* 183 (2019) 1–23.

[19] Z.H. Huang and L. Qi, Tensor complementarity problems-part III: applications, *J. Optim. Theory Appl.* 183 (2019) 771–791.

[20] D.H. Li, S.L. Xie, and H.R. Xu, Splitting methods for tensor equations, *Numer. Linear Algebra Appl.* 24 (2017): e2102.

[21] X. Li, Z. Luo and Y. Chen, Sparse least squares solutions of multilinear equations, To appear in *Linear Multilinear Algebra* (2023).

[22] Z. Li, Y. Dai, and H. Gao, Alternating projection method for a class of tensor equations, *J. Comput. Appl. Math.* 346 (2019) 490–504.

[23] H. Liu and R.F. Barber, Between hard and soft thresholding: Optimal iterative thresholding algorithms, *Information and Inference: A Journal of the IMA* 9 (2020) 899-–933.

[24] Z. Luo, L. Qi, and N. Xiu, The sparsest solutions to Z-tensor complementarity problems, *Optim. Lett.* 11 (2017) 471–482.

[25] N. Meng and Y.-B. Zhao, Newton-step-based hard thresholding algorithms for sparse signal recovery, *IEEE Trans. Signal Process.* 68 (2020) 6594–6606.

[26] N. Meng, Y.B. Zhao, M. Kocvara, and Z.F. Sun, Partial gradient optimal thresholding algorithms for a class of sparse optimization problems, *J. Global Optim.* (2022), doi.org/10.1007/s10898-022-01143-1.

[27] L. Qi and Z.H. Huang, Tensor pomplementarity problems-part II: solution methods, *J. Optim. Theory Appl.* 183 (2019) 365–385.

[28] S.L. Xie, D.H. Li, and H.R. Xu, An iterative method for finding the least solution to the tensor complementarity problem, *J. Optim. Theory Appl.* 175 (2017) 119–136.

[29] Z. Xie, X. Jin, and Y. Wei, Tensor methods for solving symmetric M-tensor systems, *J. Sci. Comput.* 74 (2018) 412–425.

[30] H.R. Xu, D.H. Li, and S.L. Xie, An equivalent tensor equation to the tensor complementarity problem with positive semi-definite Z-tensor, *Optim. Lett.* 13 (2019) 685–694.

[31] J. Yan, Y. Xu, and Z.H. Huang, A homotopy method for solving multilinear systems with strong completely positive tensors, *Appl. Math. Lett.* 124 (2022): 107636.

[32] C. Zhao, N. Xiu, H. Qi, et al. A lagrange-Newton algorithm for sparse nonlinear programming, *Math. Program.* 195 (2022) 903–928.

[33] Y.B. Zhao, *Sparse Optimization Theory and Methods*, CRC Press, Boca Raton, FL, 2018.

[34] Y.B. Zhao, Optimal k-thresholding algorithms for sparse optimization problems, *SIAM J. Optim.* 30 (2020) 31–55.

[35] Y.B. Zhao and Z.Q. Luo, Analysis of optimal k-thresholding algorithms for compressed sensing, *Signal Process.* 187 (2021) 108–148.

[36] Y.B. Zhao and Z.Q. Luo, Natural thresholding algorithms for signal recovery with sparsity, *IEEE Open J. Signal Process.* 3 (2022) 417–431.

[37] X.H. Zheng, Y. Wang and Z.H. Huang, A linearized method for solving tensor complementarity problems with implicit Z-tensors, *Optim. Lett.* 2023, https://doi.org/10.1007/s11590-023-02043-3

[38] S. Zhou, N. Xiu and H. Qi, Global and quadratic convergence of newton hard-thresholding pursuit, *J. Mach. Learn. Res.* 22 (2021) 1–45.

Manuscript received 11 September 2023
revised 11 December 2023
accepted for publication 25 December 2023

ZIXIN DENG
School of Mathematics, Tianjin University
Tianjin 300350, P.R. China
E-mail address: dengzx@tju.edu.cn

ZHENG-HAI HUANG
School of Mathematics, Tianjin University
Tianjin 300350, P.R. China
E-mail address: huangzhenghai@tju.edu.cn