Newton’s Method for M-Tensor Equations

Dong-Hui Li1 · Jie-Feng Xu1 · Hong-Bo Guan1,2

Received: 26 January 2021 / Accepted: 28 June 2021 / Published online: 12 July 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
We are concerned with the tensor equations whose coefficient tensors are M-tensors. We first propose a Newton method for solving the equation with a positive constant term and establish its global and quadratic convergence. Then we extend the method to solve the equation with a nonnegative constant term and establish its convergence. At last, we do numerical experiments to test the proposed methods. The results show that the proposed methods are quite efficient.

Keywords M-tensor equation · Newton method · Quadratic convergence

Mathematics Subject Classification 65H10 · 65K10 · 90C33

1 Introduction

Tensor equation is also called multi-linear equation. It appears in many practical fields including data mining and numerical partial differential equations [4,7–9,13,16,18,26].

The study on numerical methods for solving tensor equation has begun only a few years ago. Most of them focus on solving the M-tensor equation (abbreviated as M-Teq). The existing methods for solving M-Teq focus on finding a positive solution under the restriction that the constant term is positive. Those methods include...
the Jacobian and Gauss–Seidel methods [7], the homotopy method [11], the tensor splitting methods [19], the Newton-type methods [12], and the continuous time neural network methods [25].

Recently, Bai et al. [2] proposed a nonnegativity preserving algorithm to solve M-Teq with a nonnegative constant term. Li et al. [15] proposed a monotone iterative method to solve the M-Teq with an arbitrary constant term. Li et al. [14] proposed an inexact Newton method with a positive constant term and extended the method to solve the M-Teq with a nonnegative constant term.

There are a few methods to solve the tensor equation with other structure tensors or more general tensors. Li et al. [16] extended the classic splitting methods for solving system of linear equations to solve tensor equations with symmetric tensors. Li et al. [17] proposed an alternating projection method for solving tensor equations with a special third-order tensor. Other related works can also be found in [3–5,10,18,20–22,26–29].

In this paper, we further study numerical methods for solving M-Teq. Our purpose is to find a nonnegative solution of the equation with a nonnegative constant term. As we know in [14], finding a nonnegative solution of the M-tensor equation can be done by finding a positive solution of a lower-dimensional M-tensor equation with a nonnegative constant term. It is noting that the constant term of that lower dimensional equation is still not guaranteed to be positive. So most of the existing methods are not applicable. We will propose a Newton method to get a positive solution of the equation and prove its global convergence and quadratic convergence. Our numerical results show that the proposed Newton method is very efficient.

In the next section, we do some preliminaries. In Sect. 3, we propose a Newton method to find the unique positive solution to an M-Teq with a positive constant term. We will also establish its global and quadratic convergence in Sect. 3. In Sect. 4, we extend the idea of the method proposed in Sect. 3 to get a nonnegative solution of an M-tensor equation with a nonnegative constant term and establish its convergence. It should be pointed out that such an extension is not trivial because the M-tensor equation with a positive and the M-Teq with a nonnegative constant terms are quite different. In Sect. 5, we do numerical experiments to test the proposed methods and compare their performance with some existing Newton methods. At last, we conclude the paper by giving some final remarks in Sect. 6.

2 Preliminaries

In this paper, we will consider the following general tensor equation

\[ F(x) = Ax^{m-1} - b = 0, \quad (1) \]

where \( x, b \in \mathbb{R}^n \) and \( \mathcal{A} \) is an \( m \)-th order \( n \)-dimensional tensor consisting of \( n^m \) elements:

\[ \mathcal{A} = (a_{i_1i_2\ldots i_m}), \quad a_{i_1i_2\ldots i_m} \in \mathbb{R}, \quad 1 \leq i_1, i_2, \ldots, i_m \leq n, \]
and \( Ax^{m-1} \in \mathbb{R}^n \) with elements

\[
(AX^{m-1})_i = \sum_{i_2,\ldots,i_m=1}^n a_{i_2\ldots,i_m}x_{i_2}\ldots x_{i_m}, \quad i = 1, 2, \ldots, n.
\]

We will pay particular attention to the M-Teq (1) in which \( A \) is an M-tensor. To give a definition of the M-tensor, we first introduce some concepts. We refer to two recent books \[23,24\] for details.

We denote the set of all \( m \)-th order \( n \)-dimensional tensors by \( T(m,n) \) and \( [n] = \{1,2,\ldots,n\} \).

A tensor \( A = (a_{i_1i_2\ldots i_m}) \in T(m,n) \) is called nonnegative tensor, denoted by \( A \geq 0 \), if all its elements are nonnegative, i.e., \( a_{i_1i_2\ldots i_m} \geq 0, \forall i_1, \ldots, i_m \in [n] \). \( A \) is called the identity tensor, denoted by \( I \), if its diagonal elements are all ones and other elements are zeros, i.e., all \( a_{i_1i_2\ldots i_m} = 0 \) except \( a_{ii\ldots i} = 1, \forall i, i_1, \ldots, i_m \in [n] \).

If a real number \( \lambda \) and a nonzero real vector \( x \in \mathbb{R}^n \) satisfy

\[
Ax^{m-1} = \lambda x^{[m-1]},
\]

then \( \lambda \) is called an H-eigenvalue of \( A \) and \( x \) is an H-eigenvector of \( A \) associated with \( \lambda \). Here, for a real scalar \( \alpha \), \( x^{[\alpha]} = (x_1^{\alpha}, x_2^{\alpha}, \ldots, x_n^{\alpha})^T \) whenever it is meaningful.

A tensor \( A = (a_{i_1i_2\ldots i_m}) \in T(m,n) \) is symmetric if its elements \( a_{i_1i_2\ldots i_m} \) are invariant under any permutation of their indices. The set of all \( m \)-th order \( n \)-dimensional symmetric tensors is denoted by \( ST(m,n) \). \( A \) is called semi-symmetric if for any \( i \in [n] \), the sub-tensor \( A_i := (a_{i_1i_2\ldots i_m})_{1\leq i_2,\ldots,i_m \leq n} \) is symmetric. In the case \( A \in ST(m,n) \), we have

\[
\nabla (Ax^m) = mAx^{m-1},
\]

where \( \nabla \) denotes the gradient operator. In the case \( A \in T(m,n) \) is semi-symmetric, we have

\[
\nabla (Ax^{m-1}) = (m-1)Ax^{m-2}.
\]

The definition of M-tensor is introduced in \[6,24,30\].

**Definition 2.1** A tensor \( A \in T(m,n) \) is called an M-tensor, if it can be written as

\[
A = sI - B, \quad B \geq 0, \quad s \geq \rho(B), \tag{2}
\]

where \( \rho(B) \) is the spectral radius of tensor \( B \), that is

\[
\rho(B) = \max \{ |\lambda| : \lambda \text{ is an eigenvalue of } B \}.
\]

If \( s > \rho(B) \), then \( A \) is called a strong or nonsingular M-tensor.
For \( x, y \in \mathbb{R}^n \), we use \( x \circ y \) to denote their Hadamard product defined by
\[
x \circ y = (x_1 y_1, \ldots, x_n y_n)^T.
\]
We use \( \mathbb{R}^n_+ \) and \( \mathbb{R}^n_{++} \) to denote the sets of all nonnegative vectors and positive vectors in \( \mathbb{R}^n \). That is,
\[
\mathbb{R}^n_+ = \{ x \in \mathbb{R}^n \mid x \geq 0 \} \quad \text{and} \quad \mathbb{R}^n_{++} = \{ x \in \mathbb{R}^n \mid x > 0 \}.
\]

3 A Newton Method for the M-Tensor Equation (1) with \( b > 0 \)

Newton’s method is a famous iterative method for solving nonlinear equations. An attractive property of the method is its superlinear/quadratic convergence if the Jacobian of the residual function is nonsingular at the solution. However, when the Jacobian of the function is singular, the method may lose its fast convergence property or even fails to work. For example, we consider the following system of nonlinear equations
\[
F_i(x) = \sum_{j=1}^{n} a_{ij} x_j^{m-1} - b_i = 0, \quad i = 1, 2, \ldots, n.
\]
(3)

It is a special tensor equation. When matrix \( A = (a_{ij}) \) is nonsingular and \( b = (b_1, \ldots, b_n)^T \geq 0 \), the equation has a unique solution satisfying \( \bar{x}_i = (A^{-1} b)_i^{1/(m-1)} \), \( i = 1, 2, \ldots, n \). If there is some \( \bar{x}_i = 0 \), then the Jacobian \( F'(\bar{x}) \) is singular. As a result, the Newton method may lose its superlinear/quadratic convergence or even fails.

In this section, we propose a Newton method to find the unique positive solution to the M-Teq (1) with \( b > 0 \). Throughout this section, without specification, we always suppose that the following assumption holds.

**Assumption 3.1** Tensor \( A \) in (1) is a semi-symmetric and strong M-tensor, and \( b > 0 \).

Since our purpose is to get a positive solution of the M-Teq (1), we restrict \( x \in \mathbb{R}^n_{++} \). Making a variable transformation \( y = x^{[m-1]} \), we formulate the M-Teq (1) as
\[
f(y) = F \left( y^{\left[ \frac{1}{m-1} \right]} \right) = A \left( y^{\left[ \frac{1}{m-1} \right]} \right)^{m-1} - b = 0.
\]
(4)

A direct computation gives
\[
f'(y) = A \left( y^{\left[ \frac{1}{m-1} \right]} \right)^{m-2} \text{diag} \left( y^{\left[ \frac{1}{m-1} - 1 \right]} \right).
\]

It follows that
\[
f'(y)y = A \left( y^{\left[ \frac{1}{m-1} \right]} \right)^{m-2} \text{diag} \left( y^{\left[ \frac{1}{m-1} - 1 \right]} \right) y = A \left( y^{\left[ \frac{1}{m-1} \right]} \right)^{m-1} = f(y) + b.
\]
For $\epsilon \in (0, 1)$ and $\beta > 0$, define

$$F_\epsilon = \left\{ x \in \mathbb{R}^n_+ : Ax^{m-1} \geq \epsilon b \right\} = \left\{ y \in \mathbb{R}^n_+ : A \left( y \left[ \frac{1}{m-1} \right] \right)^{m-1} \geq \epsilon b \right\}$$

and

$$\Omega_\beta = \left\{ x \in \mathbb{R}^n : \| Ax^{m-1} - b \| \leq \beta \right\} = \left\{ y \in \mathbb{R}^n : \| A \left( y \left[ \frac{1}{m-1} \right] \right)^{m-1} - b \| \leq \beta \right\}.$$

It is easy to see that for any $\epsilon \in (0, 1]$, the positive solution of the equation (1) is contained in $F_\epsilon$. In addition, the Jacobian matrices $F'(x)$ and $f'(y)$ are nonsingular M-matrices for any $x, y \in F_\epsilon$.

**Lemma 3.1** The following statements are true.

(i) If $A = (a_{i_1...i_m})$ is a Z-tensor and $b > 0$, then for any $\epsilon > 0$, the set $F_\epsilon$ is bounded away from zero. That is, there is a constant $\delta > 0$ such that $x \geq \delta e$, $\forall x \in F_\epsilon$, where $e = (1, 1, \ldots, 1)^T$.

(ii) If $A$ is a strong M-tensor, then for any $\beta \in \mathbb{R}$, the level set $\Omega_\beta$ is bounded.

**Proof** We prove the lemma by contradiction.

(i) Suppose conclusion (i) is not true. Then, there is a sequence $\{x_k\} \subset F_\epsilon$ and an index $i \in [n]$ such that $\{(x_k)_i\} \to 0$. Since $x_k \in F_\epsilon$, it holds that

$$(\epsilon - 1)b_i \leq a_{i...i} (x_k)_i^{m-1} + \sum_{(i_2, \ldots, i_m) \neq (i, \ldots, i)} a_{i_2...i_m} (x_k)_{i_2} \ldots (x_k)_{i_m} - b_i \leq a_{i...i} (x_k)_i^{m-1} - b_i.$$

Taking limits in both sides of the last inequality, we get $\epsilon b_i \leq 0$. It is a contradiction. Consequently, the set $F_\epsilon$ is bounded away from zero.

(ii) Suppose that for some $\beta \in \mathbb{R}$, the level set $\Omega_\beta$ is not bounded. Then there is a sequence $\{x_k\} \subset \Omega_\beta$ satisfying $\|x_k\| \to \infty$, as $k \to \infty$. However, we obviously have

$$\frac{\beta}{\| Ax_k^{m-1} - b \|} \geq \frac{\| Ax_k^{m-1} - b \|}{\| Ax_k^{m-1} \|} \geq \frac{\| A (x_k / \| x_k \|)^{m-1} - b \|}{\| x_k^{m-1} \|},$$

Suppose that the subsequence $\{x_k / \| x_k \|\}_K$ converges to some $\tilde{u} \neq 0$. Taking limits as $k \to \infty$ with $k \in K$ in both sides of the last inequality, we get $A\tilde{u}^{m-1} = 0$. Since $A$ is a strong M-tensor, from Theorem 2.3 in [14], we get a contradiction. $\square$

The idea to develop Newton’s method is described as follows. Starting from some $y_0 = x_0^{[m-1]}$ satisfying $x_0 \in F_\epsilon$ with some given small $\epsilon \in (0, 1)$, the method
generates a sequence of iterates \( \{ x_k \} \subset \mathcal{F} \) by a damped Newton iteration such that the residual sequence \( \{ \| f(y_k) \| \} \) is decreasing.

We first show the following lemma.

**Lemma 3.2** Suppose that \( A \) is a strong M-tensor and \( b > 0 \). Let \( d \) be the Newton direction that is the unique solution of the system of linear equations

\[
f'(y)d + f(y) = 0.
\]

Then there is a constant \( L > 0 \) such that the inequality

\[
A[(y + \alpha d)\frac{1}{m-1}]^{m-1} \geq \epsilon b + \alpha \left( (1-\epsilon)b - \frac{1}{2}L\alpha \|d\|^2 e \right), \quad \forall \alpha > 0, \forall y \in \mathcal{F} \cap \Omega_\beta,
\]

where \( e = (1, 1, \ldots, 1)^T \).

**Proof** It follows from Lemma 3.1 that the set \( D = \mathcal{F} \cap \Omega_\beta \) has positive lower and upper bounds. It is also clear that function \( f(y) \) is twice continuously differentiable on \( D \). Denoted by \( L \) the bound of \( \|f''(y)\| \) on \( D \). By the use of the mean value theorem, for any \( y \in D \) and \( \alpha > 0 \), we obtain

\[
A[(y + \alpha d)\frac{1}{m-1}]^{m-1} = f(y + \alpha d) + b \\
= f(y) + \alpha f'(y)d + \alpha \int_0^1 [f'(y + \alpha \tau d) - f'(y)]d \cdot d\tau + b \\
\geq (1 - \alpha) f(y) + b - \frac{1}{2}L\alpha^2 \|d\|^2 e \\
= (1 - \alpha)A[(y)\frac{1}{m-1}]^{m-1} + \alpha b - \frac{1}{2}L\alpha^2 \|d\|^2 e \\
\geq (1 - \alpha)\epsilon b + \alpha b - \frac{1}{2}L\alpha^2 \|d\|^2 e \\
= \epsilon b + \alpha \left( (1 - \epsilon)b - \frac{1}{2}L\alpha \|d\|^2 e \right).
\]

The proof is complete. \( \square \)

Denote

\[
\tilde{\alpha} = \min \left\{ \frac{2(1 - \epsilon)b_i}{L\|d\|^2} : i \in [n] \right\}.
\]

It follows from the last lemma that if \( y \in D \), then it holds that

\[
A[(y + \alpha d)\frac{1}{m-1}]^{m-1} \geq \epsilon b, \quad \forall \alpha \in (0, \tilde{\alpha}).
\]

The steps of the Newton method are stated as follows.
Algorithm 3.1 (Newton’s Method)

**Initial.** Given a small constant $\epsilon \in (0, 1)$ and constants $\sigma \in (0, \frac{1}{2})$, $\eta, \rho \in (0, 1)$.
Select an initial point $x_0 \in \mathcal{F}_\epsilon$. Let $y_0 = x_0^{[m-1]}$ and $k = 0$.

**Step 1.** Stop if $\|f(y_k)\| \leq \eta$.

**Step 2.** Solve the system of linear equations
\begin{equation}
    f'(y_k)d_k + f(y_k) = 0 \tag{6}
\end{equation}
to get $d_k$.

**Step 3.** Determine a steplength $\alpha_k = \max\{\rho^i : i = 0, 1, \ldots\}$ such that $y_k + \alpha_k d_k \in \mathcal{F}_\epsilon$ and that the inequality
\begin{equation}
    \|f(y_k + \alpha_k d_k)\|^2 \leq (1 - 2\sigma \alpha_k)\|f(y_k)\|^2 \tag{7}
\end{equation}
is satisfied.

**Step 4.** Let $y_{k+1} = y_k + \alpha_k d_k$ and $x_{k+1} = y_k^{[m-1]}$. Go to Step 1.

**Remark 3.1** – In [14], the authors also proposed a Newton method for solving the M-Teq (1). The authors first reformulate the M-Teq as the following system of nonlinear equations:
\begin{align*}
    E(y) \triangleq \text{diag}\left(y^{[1-1]}\right) f(y) = \left(y^{[1-1]} \circ f(y)\right) &= 0 \\
\end{align*}
and then solve it by Newton’s method. Taking into account the relation between $E(y)$ and $f(y)$, the subproblem in [14] can be written as
\begin{equation}
    f'(y_k)d_k + f(y_k) = r_k, \tag{9}
\end{equation}
where
\begin{align*}
    r_k &= \text{diag}\left(y_k^{[1-1]} \circ f(y_k)\right)d_k = O\left(\|f(y_k)\| \|d_k\|\right). \\
\end{align*}
Consequently, the method there is indeed an inexact Newton method.

– It is easy to see that the last method is very similar to the standard damped Newton method except the line search step where we need to ensure $x_{k+1} \in \mathcal{F}_\epsilon$. If $y_k + d_k \in \mathcal{F}_\epsilon$, then the last method is equivalent to the standard Newton method for solving nonlinear equation $f(y) = 0$.

– The steps of the last method ensure that the generated sequence of iterates $\{x_k\} \subset \mathcal{F}_\epsilon$. As a result, $f'(y_k)$ is a strong M-matrix and hence the method is well defined. Moreover, the residual sequence $\{\|f(y_k)\|\}$ is decreasing. It then follows from Lemma 3.1 that there are positive constants $c$ and $C$ satisfying $c \leq C$ such that
\begin{equation}
    ce \leq y_k \leq Ce. \tag{8}
\end{equation}

It particularly shows that the method is positive preserving.
It follows from Lemma 3.2 that if \( y_k \in \mathcal{F}_\epsilon \), then
\[
y_k + \alpha_k d_k \in \mathcal{F}_\epsilon, \forall \alpha_k \in (0, \bar{\alpha})
\]
where
\[
\bar{\alpha}_k = \min \left\{ \frac{2(1 - \epsilon)b_i}{L \|d_k\|^2} : i \in [n] \right\} \cap \left\{ \frac{-(y_k)_i}{(d_k)_i} : (d_k)_i < 0 \right\}
\]
(9)
and \( L \) is the bound of \( f''(y) \) on the set compact set \( \mathcal{F}_\epsilon \cap \Omega \|f(y_0)\| \).

Let \( x^* \) be the unique positive solution to the M-Teq (1) and \( y^* = (x^*)^{[m-1]} \). It is easy to see that for any \( \epsilon \in (0, 1) \), \( x^* \in \mathcal{F}_\epsilon \). Consequently, the matrix \( f'(y^*) \) is a nonsingular M-matrix. As a result, the full-step Newton method is locally quadratically convergent.

In what follows, we are going to show that Algorithm 3.1 is globally convergent and that after a finite number of iterations, the method reduces to the full-step Newton method. Consequently, it is quadratically convergent. We first show the following lemma.

**Lemma 3.3** Suppose that \( A \) is a strong M-tensor and \( b > 0 \). Then the sequence \( \{y_k\} \) and \( \{d_k\} \) generated by Algorithm 3.1 are bounded. In addition, there is a positive constant \( \bar{\alpha} \) such that
\[
y_k + \alpha_k d_k \in \mathcal{F}_\epsilon, \quad \forall \alpha_k \in (0, \bar{\alpha}).
\]
(10)

**Proof** By the steps of the algorithm, it is easy to see that the sequence \( \{y_k\} \) is contained in the compact set \( D = \mathcal{F}_\epsilon \cap \Omega \|f(y_0)\| \) and hence bounded. Since \( f'(y_k) \) is a nonsingular M-matrix and \( D \) is compact, the sequence \( \{d_k\}_K \) is bounded too. Note that \( b > 0 \) and \( \{y_k\} \) has a positive lower bound, the scalar \( \bar{\alpha}_k \) defined by (9) has a positive lower bound. This, together with the boundedness of \( \|d_k\| \) implies that \( \bar{\alpha}_k \) has a positive lower bound. Consequently, (10) is satisfied with some positive \( \bar{\alpha} \). \( \Box \)

The following theorem establishes the global convergence of the proposed method.

**Theorem 3.1** Suppose that \( A \) is a strong M-tensor and \( b > 0 \). Then the sequence of iterates \( \{x_k\} \) generated by Algorithm 3.1 converges to the unique positive solution to the M-Teq (1).

**Proof** It suffices to show that there is an accumulation point \( \bar{y} \) of \( \{y_k\} \) satisfying \( f(\bar{y}) = 0 \). Let the subsequence \( \{y_k\}_K \) converge to \( \bar{y} \). Without loss of generality, we suppose that the subsequence \( \{d_k\}_K \) converges to some \( \bar{d} \).

Denote \( \bar{\alpha} = \lim \inf_{k \to \infty, k \in K} \alpha_k \). If \( \bar{\alpha} > 0 \), the inequality (7) implies \( f(\bar{y}) = 0 \). Consider the case \( \bar{\alpha} = 0 \). By the line search rule, when \( k \in K \) is sufficiently large, \( \alpha'_k = \rho^{-1} \alpha_k \) will not satisfy (7), i.e.,
\[
\|f(y_k + \alpha'_k d_k)\|^2 - \|f(y_k)\|^2 > -2\sigma \alpha'_k \|f(y_k)\|^2
\]
Dividing both sizes of the last inequality by $\alpha'_k$ and then taking limits as $k \to \infty$ with $k \in K$, we get

$$2 f(\bar{y})^T f'(\bar{y}) \bar{d} \geq -2\sigma \| f(\bar{y}) \|^2$$  \hspace{1cm} (11)

On the other hand, by taking limits in (6), we can obtain $f'(\bar{y}) \bar{d} + f(\bar{y}) = 0$. It together with (11) and the fact $\sigma \in (0, 1)$ yields $f(\bar{y}) = 0$. The proof is complete. \hfill \Box

The last theorem established the global convergence of the proposed Newton method. Moreover, we see from (5) that $y_k + d_k \in F_\epsilon$ for all $k$ sufficiently large because $\{d_k\} \to 0$. Consequently, the method locally reduces to a standard damped Newton method. Following a standard discussion as the proof of the quadratic convergence of a damped Newton method, it is not difficult to prove the quadratic convergence of the method. We give the result but omit the proof.

**Theorem 3.2** Let the conditions in Theorem 3.1 hold. Then the convergence rate of the sequence $\{y_k\}$ generated by Algorithm 3.1 is quadratic.

**4 An Extension**

In this section, we extend the Newton method proposed in the last section to the M-Teq (1) with $b \geq 0$. In the case $b$ has zero elements, the M-Teq may have multiple nonnegative or positive solutions. Our purpose is to find one nonnegative or positive solution of the equation. By Theorem 2.6 in [14], a nonnegative solution of (1) has zero elements if and only if $A$ is reducible with respect to some $I \subseteq I_0$, where

$$I_0 = \{i \in [n] \mid b_i = 0\}.$$

Since justifying the reducibility is an easy task, without loss of generality, we suppose that the nonnegative solutions of the M-Teq are positive.

**Assumption 4.1** Suppose $b \geq 0$ and that tensor $A$ is a strong M-tensor and irreducible with respect to $I_0$. Suppose further that for each $i \in I_0$, there is an element $a_{i_2 \ldots i_m} \neq 0$ with, $i_2, \ldots, i_m \in I_+$. 

Under the conditions of Assumption 4.1, we have

$$A^{c}(y^{\left[1 \ldots m-1\right]i}^{m-1})_{I_0} < 0, \quad \forall y \in \mathbb{R}_{++}^n.$$

For the sake of convenience, we introduce some notations. Denote $I_+ = \{i : b_i > 0\}$ and $I_0 = \{i : b_i = 0\}$. For given constants $1 > \epsilon > \epsilon' > 0$, we define

$$\bar{F}_{\epsilon, \epsilon'} = \bar{F}_{\epsilon}^1 \cap \bar{F}_{\epsilon'}^2$$

with

$$\bar{F}_{\epsilon}^1 = \{y \in \mathbb{R}_{++}^n : \left(A\left(y^{\left[1 \ldots m-1\right]}^{(m-1)}\right)_{I_+} \geq \epsilon b_{I_+}\right)\}.$$
and
\[
\overline{F}_{e'}^2 = \left\{ y \in \mathbb{R}^n_{++} : \left( A \left( y^l_{m-1} \right)^{m-1} \right)_{I_0} \geq e' f'(y) I_0 I_+ f'(y)^{-1} I_+ b I_+ \right\}.
\]

It is easy to see that every solution \( \bar{x} \in \overline{F}_e \).
For \( y \in \mathbb{R}^n_{++} \), we split \( f'(y) \) into
\[
f'(y) = \begin{pmatrix} f'_{I_+ I_+} (y) & f'_{I_+ I_0} (y) \\ f'_{I_0 I_+} (y) & f'_{I_0 I_0} (y) \end{pmatrix}.
\]

It is easy to see that \( f'(y) \) is a Z-matrix.

The next theorem shows that for any \( y \in \overline{F}_e, f'(y) \) is a nonsingular M-matrix. As a result, the set \( \overline{F}_e \) is well defined.

**Theorem 4.1** Let \( 1 > \epsilon > \epsilon' > 0 \). For any \( y \in \overline{F}_{e', e'} \), \( f'(y) \) is a nonsingular M-matrix.

**Proof** By direct computation, we get \( f'(y) = A \left( y^l_{m-1} \right)^{m-1} \). The condition \( y \in \overline{F}_{e}^1 \) yields
\[
0 < \epsilon b_{I_+} \leq \left( A \left( y^l_{m-1} \right)^{m-1} \right)_{I_+} = f'(y) I_+ I_+ y_{I_+} + f'(y) I_+ I_0 y_{I_0} \leq f'(y) I_+ I_+ y_{I_+}.
\]

Consequently, \( f'(y) I_+ I_+ \) is a nonsingular M-matrix. We are going to show that the Schur complement
\[
f'(y) I_0 I_0 - f'(y) I_0 I_+ f'(y)^{-1} I_+ I_+ f'(y) I_+ I_0
\]
is also a nonsingular M-matrix.

Observing \( f'(y) I_0 I_+ \leq 0 \), we get from the condition \( y \in \overline{F}_{e'}^2 \),
\[
f'(y) I_0 I_0 y_{I_0} \geq \epsilon' f'(y) I_0 I_+ f'(y)^{-1} I_+ I_+ b I_+ + f'(y) I_0 I_0 y_{I_+}
\]
\[
\geq \epsilon' f'(y) I_0 I_+ f'(y)^{-1} I_+ I_+ b I_+ + f'(y) I_0 I_+ f'(y)^{-1} I_+ I_+ (\epsilon b_{I_+} - f'(y) I_0 I_0 y_{I_0})
\]
\[
= -(\epsilon - \epsilon') f'(y) I_0 I_+ f'(y)^{-1} I_+ I_+ b I_+ + f'(y) I_0 I_+ f'(y)^{-1} I_+ I_+ f'(y) I_0 I_0 y_{I_0},
\]
which implies
\[
\left( f'(y) I_0 I_0 - f'(y) I_0 I_+ f'(y)^{-1} I_+ I_+ f'(y) I_+ I_0 \right) y_{I_0} \geq -(\epsilon - \epsilon') f'(y) I_0 I_+ f'(y)^{-1} I_+ I_+ b I_+ > 0.
\]
The last condition ensures that \( f'(y) \) is a nonsingular M-matrix. \( \Box \)

Similar to Lemma 3.1, we have the following lemma.

**Lemma 4.1** If \( A = (a_{i_1 \ldots i_m}) \) is a Z-tensor and \( b \geq 0 \), then for any \( \epsilon > 0 \), the set \( \overline{F}_e \) is bounded away from zero. That is, there is a constant \( \delta > 0 \) such that
\[
y \geq \delta e, \quad \forall y \in \overline{F}_{e, e'}.
\]
Proof First, following the same arguments as the proof of Lemma 3.1 (i), it is easy to show that $y_1$ has a positive lower bound. We only need to prove that $y_0$ has a positive lower bound too.

It is easy to see by the definition of $F_1$ that each $y \in F_1$ satisfies

$$
\epsilon b_1 \leq (A(y [n-1])^{m-1})_{1+} = (f'(y)y)_{1+} \leq f'(y)_{1+} b_{1+},
$$

which implies

$$
y_{1+} \geq f'(y)_{1+}^{-1} b_{1+} > f'(y)_{1+}^{-1} b_{1+}.
$$

The condition $y \in F_0$ implies

$$
0 \leq f'_{l_0}(y) \left(y_{1+} - \epsilon' f'(y)_{1+}^{-1} b_{1+}\right) + f_{l_0}(y) y_{l_0}.
$$

By the condition of Assumption 4.1 and the fact that $y_{1+}$ has a positive lower bound, we claim that the vector $f'_{l_0}(y) \left(y_{1+} - \epsilon' f'(y)_{1+}^{-1} b_{1+}\right)$ is bounded away from zero. Taking into account that $f'_{l_0}(y)$ is a Z-matrix and $y_{l_0} > 0$, it is easy to see that $y_{l_0}$ is bounded away from zero too. \hfill \Box

In what follows, we propose a Newton method for finding a positive solution to the M-Teq (1) with $b \geq 0$ as follows.

**Algorithm 4.1 (Extended Newton Method for (1) with $b \geq 0$)**

**Initial.** Given constants $\epsilon, \epsilon', \rho, \eta, \sigma \in (0, 1)$ satisfying $\epsilon' < \epsilon$. Select an initial point $y_0 \in F_{\epsilon, \epsilon'}$. Let $k = 0$.

**Step 1.** Stop if $\|f(y_k)\| \leq \eta$.

**Step 2.** Solve the system of linear equations

$$
f'(y_k) d_k + f(y_k) = 0. \tag{12}
$$

to get $d_k$.

**Step 3.** Determine a steplength $\alpha_k = \max\{\rho^i : i = 0, 1, \ldots\}$ such that $y_k + \alpha_k d_k \in F_{\epsilon, \epsilon'}$ and that the inequality

$$
\|f(y_k + \alpha_k d_k)\|^2 \leq (1 - 2\sigma \alpha_k) \|f(y_k)\|^2 \tag{13}
$$

is satisfied.

**Step 4.** Let $y_{k+1} = y_k + \alpha_k d_k$. Go to Step 1.

In what follows, we show that the algorithm above is well-defined.

**Proposition 4.1** Let the conditions in Assumption 4.1 hold. Then Algorithm 4.1 is well defined.
Proof It suffices to verify that the relation \( y_k + \alpha_k d_k \in \overline{\mathcal{F}}_{\epsilon, \epsilon'} \) is satisfied for all \( \alpha > 0 \) sufficiently small. Indeed, we have

\[
A \left( (y_k + \alpha d_k)^{1 \over m-1} \right)^{m-1} = f(y_k + \alpha d_k) + b \\
= f(y_k) + \alpha f'(y_k) d_k + O(\|\alpha d_k\|^2) + b \\
= (1 - \alpha) f(y_k) + b + O(\|\alpha d_k\|^2) \\
= (1 - \alpha) A \left( y_k^{1 \over m-1} \right)^{m-1} + \alpha b + O(\|\alpha d_k\|^2).
\]

Since \( y_k \in \overline{\mathcal{F}}_{\epsilon}^{1} \) and \( b_{I^+} > 0 \), we get from the last equality

\[
A \left( (y_k + \alpha d_k)^{1 \over m-1} \right)^{m-1} \bigg|_{I^+} \geq \epsilon b_{I^+} + \alpha (1 - \epsilon) b_{I^+} + O(\alpha \|d_k\|^2),
\]

which implies \( y_k + \alpha d_k \in \overline{\mathcal{F}}_{\epsilon}^{1} \) for all \( \alpha > 0 \) sufficiently small.

Similarly, we have by the fact \( y_k \in \overline{\mathcal{F}}_{\epsilon}^{2} \) and \( b_{I_0} = 0 \)

\[
\left. \left( A \left( (y_k + \alpha d_k)^{1 \over m-1} \right)^{m-1} \right) \right|_{I_0} \geq \epsilon' f'(y_k) I_0 + f'(y_k I_0 I_+, b_{I^+} \\
+ \alpha \left( - \epsilon' f'(y_k) I_0 I_+ + f'(y) I_0 I_+ b_{I^+} + O(\alpha \|d_k\|^2) \right).
\]

By the condition of Assumption 4.1, it is not difficult to see from the last inequality that we claim that the inequality

\[
A \left( (y_k + \alpha d_k)^{1 \over m-1} \right)^{m-1} \bigg|_{I_0} \geq \epsilon' f'(y) I_0 I_+ f'(y I_0 I_+) b_{I^+}
\]

is satisfied for all \( \alpha > 0 \) sufficiently small. \( \square \)

It is easy to show that Lemma 3.1 holds true for the case \( b \geq 0 \). As a result, the sequence generated by Algorithm 4.1 is bounded. Consequently, the inequalities (14) and (15) ensure that there is a positive constant \( \tilde{\alpha} > 0 \) such that \( x_k + \alpha d_k \in \overline{\mathcal{F}}_{\epsilon, \epsilon'} \), \( \forall \alpha \in (0, \tilde{\alpha}] \).

Similar to the proof of Theorem 3.1, we can prove the global convergence of Algorithm 4.1.

Theorem 4.2 Let the conditions in Assumption 4.1 hold. Then the sequence of iterates \( \{y_k\} \) generated by Algorithm 4.1 is bounded. Moreover, every accumulation point of the iterates \( \{y_k\} \) is a positive solution to the \( M \)-tensor equation \( f(y) = 0 \).

The remainder of this section is devoted to the proof of the quadratic convergence of Algorithm 4.1. It should be pointed out that the unit steplength may not be acceptable due to the existence of zero elements in \( b \). To ensure the quadratic convergence of the method, we need to make a slight modification to Step 3 of the algorithm. Specifically, we use the following Step 3' instead of Step 3 in Algorithm 4.1.
Step 3’. If $\alpha_k = 1$ satisfies $y_k + \alpha_k d_k \in \overline{F}_{\epsilon', e'}$ and (13), then we let $\alpha_k = 1$. Otherwise, for given constant $c > 0$, we let $\beta_k = 1 - c\|f(y_k)\|$. If $\beta_k \leq 0$, we let $\beta_k = 1$. Determine a steplength $\alpha_k = \max\{\beta_k \rho^i : i = 0, 1, \ldots\}$ such that $y_k + \alpha_k d_k \in \overline{F}_{\epsilon', e'}$ and that the inequality (13) is satisfied.

It is not difficult to see that the global convergence theorem still remains true if Step 3 is replaced by Step 3’. Moreover, since $\{d_k\} \to 0$, it is easy to prove from (14), (15) and (13) that for all $k$ sufficiently large, the step $\alpha_k = \beta_k = 1 - c\|f(y_k)\|$ will be accepted. In this case, the sequence of iterates $\{y_k\}$ satisfies $y_{k+1} = y_k + \tilde{d}_k$, with $\tilde{d}_k = \beta_k d_k = (1 - c\|f(y_k)\|)d_k$ satisfying

$$f'(y_k)\tilde{d}_k + f(y_k) = f'(y_k) d_k + f(y_k) - c\|f(y_k)\|d_k = -c\|f(y_k)\|d_k.$$ 

If $y_{k+1} = y_k + \tilde{d}_k \in \overline{F}_{\epsilon', e'}$, then when $k$ is sufficiently large, $y_k$ can be regarded as the sequence generated by a full-step inexact Newton method. Consequently, the quadratic convergence becomes well-known.

**Theorem 4.3** Let the conditions in Assumption 4.1 hold. Suppose that the sequence of iterates $\{y_k\}$ generated by Algorithm 4.1 converges to a positive solution $y^*$ to the M-tensor equation $f(y) = 0$. Then the convergence rate of $\{y_k\}$ is quadratic.

**Proof** We only need to verify

$$y_{k+1} = y_k + \tilde{d}_k = y_k + \left(1 - c\|f(y_k)\|\right)d_k \in \overline{F}_{\epsilon', e'}.$$  \tag{16}

It is not difficult to show from (12) that

$$\|d_k\| = O(\|f(y_k)\|) = O(\|f(y_k)\|) = O(\|x - x^*\|).$$

Similar to the proof of (14), we can derive

$$\left(A\left((y_k + \beta_k d_k)^{\frac{1}{m-1}}\right)^{m-1}\right)_{I_+} \geq \epsilon b_{I_+} + \beta_k (1 - \epsilon) b_{I_+} + O(\beta_k \|d_k\|^2).$$

Since $\{\beta_k\} \to 1$ and $\{d_k\} \to 0$, the last inequality implies $y_k + \beta_k d_k \in \overline{F}_{\epsilon, e}^{-1}$.

We can also obtain

$$\left(A\left((y_k + \beta_k d_k)^{\frac{1}{m-1}}\right)^{m-1}\right)_{I_0} \geq \epsilon' f'(y_k)_{I_0} b_{I_+} + f'(y_k)_{I_+} b_{I_+} + \beta_k \left(-\epsilon' f'(y_k)_{I_0} b_{I_+} f'(y_k)_{I_+} b_{I_+} + O(\beta_k \|d_k\|^2)\right).$$

By the condition of Assumption 4.1, it is clear that $f'(y_k)_{I_0} b_{I_+} + f'(y_k)_{I_+} b_{I_+} < 0$. Consequently, the last inequality implies $y_k + \beta_k d_k \in \overline{F}_{\epsilon, e}^2$. \hfill $\Box$

\( \text{ Springer} \)
5 Numerical Results

In this section, we do numerical experiments to test the effectiveness of the proposed methods. We implemented our methods in Matlab R2019a and ran the codes on a computer with Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz 2.30 GHz and 16.0 GB RAM. We used a tensor toolbox [1] to proceed some tensor computation.

The test problems are from [7,14,16,26].

Problem 1. We solve tensor equation (1) where $A$ is a symmetric strong M-tensor of order $m$ ($m = 3, 4, 5$) in the form $A = sI - B$, where tensor $B$ is symmetric tensor whose entries are uniformly distributed in $(0, 1)$, and

$$s = (1 + 0.01) \cdot \max_{i=1,2,...,n} (Be^m)_i,$$

where $e = (1, 1, \ldots, 1)^T$.

Problem 2. We solve tensor equation (1) where $A$ is a symmetric strong M-tensor of order $m$ ($m = 3, 4, 5$) in the form $A = sI - B$, and tensor $B$ is a nonnegative tensor with

$$b_{i_1i_2\ldots i_m} = |\sin(i_1 + i_2 + \ldots + i_m)|,$$

and $s = n^{m-1}$.

Problem 3. Consider the ordinary differential equation

$$\frac{d^2x(t)}{dt^2} = -\frac{GM}{x(t)^2}, \quad t \in (0, 1),$$

with Dirichlet’s boundary conditions

$$x(0) = c_0, \quad x(1) = c_1,$$

where $G \approx 6.67 \times 10^{-11} N m^2/kg^2$ and $M \approx 5.98 \times 10^{24}$ is the gravitational constant and the mass of the earth.

Discretize the above equation, we have

$$\begin{cases}
  x_1^3 = c_0^3, \\
  2x_i^3 - x_{i-1}^3 - x_{i+1}^3 = \frac{GM}{(n-1)^2}, \quad i = 2, 3, \ldots, n - 1, \\
  x_n^3 = c_1^3.
\end{cases}$$

It is a tensor equation, i.e.,

$$Ax^3 = b,$$
where \( \mathcal{A} \) is a fourth-order M-tensor whose entries are

\[
\begin{aligned}
a_{1111} &= a_{n n n n} = 1, \\
a_{i i i i} &= 2, \quad i = 2, 3, \ldots, n - 1, \\
a_{i(i-1)ii} &= a_{ii(i-1)i} = a_{ii(i-1)} = -1/3, \quad i = 2, 3, \ldots, n - 1, \\
a_{i(i+1)ii} &= a_{ii(i+1)i} = a_{ii(i+1)} = -1/3, \quad i = 2, 3, \ldots, n - 1,
\end{aligned}
\]

and \( b \) is a positive vector with

\[
\begin{aligned}
b_1 &= c_0^3, \\
b_i &= \frac{G_M}{(n-1)^2}, \quad i = 2, 3, \ldots, n - 1, \\
b_n &= c_1^3.
\end{aligned}
\]

**Problem 4.** We solve tensor equation (1) where \( \mathcal{A} \) is a non-symmetric strong M-tensor of order \( m \) (\( m = 3, 4, 5 \)) in the form \( \mathcal{A} = s \mathcal{I} - \mathcal{B} \), and tensor \( \mathcal{B} \) is nonnegative tensor whose entries are uniformly distributed in \((0, 1)\). The parameter \( s \) is set to

\[
s = (1 + 0.01) \cdot \max_{i=1,2,\ldots,n} (\mathcal{B}e_i)^{m-1}.
\]

**Problem 5.** We solve tensor equation (1) where \( \mathcal{A} \) is a lower triangle strong M-tensor of order \( m \) (\( m = 3, 4, 5 \)) in the form \( \mathcal{A} = s \mathcal{I} - \mathcal{B} \), and tensor \( \mathcal{B} \) is a strictly lower triangular nonnegative tensor whose entries are uniformly distributed in \((0, 1)\). The parameter \( s \) is set to

\[
s = (1 - 0.5) \cdot \max_{i=1,2,\ldots,n} (\mathcal{B}e_i)^{m-1}.
\]

For Problems 1, 2, 4 and 5, similar to [11,12], we solved the tensor equation

\[
\hat{F}(x) = \hat{A}x^{m-1} - \hat{b} = 0
\]

instead of the tensor equation (1), where \( \hat{A} := \mathcal{A}/\omega \) and \( \hat{b} := b/\omega \) with \( \omega \) is the largest value among the absolute values of components of \( \mathcal{A} \) and \( b \). The stopping criterion is set to

\[
\| \hat{F}(x_k) \| \leq 10^{-10}.
\]

And for Problem 3, the stopping criterion is set to

\[
\frac{\| \mathcal{A}x^{m-1} - b \|}{\| b \|} \leq 10^{-10}.
\]

We also stop the tested algorithms if the number of iteration reaches to 300, which means that the method fails in solving the problem.
Table 1  Comparison on Problem 1

| $(m, n)$ | NM Iter/Time/Res | INM Iter/Time/Res | SNM Iter/Time/Res | QCA Iter/Time/Res |
|----------|------------------|-------------------|-------------------|-------------------|
| (3, 200) | 2/0.011/6.7E−13 | 11/0.037/9.1E−12 | 8/0.027/7.3E−12 | 8.4/0.037/5.3E−12 |
| (3, 401) | 2/0.081/1.3E−13 | 12/0.38/7.3E−12 | 9/0.3/4.0E−13 | 8.7/0.33/7.5E−12 |
| (3, 650) | 2/0.4/5.4E−14  | 13/1.8/6.8E−12  | 10/1.5/5.3E−16 | 9.1/1.6/1.1E−11  |
| (4, 40)  | 2/0.0035/2.8E−13| 8.4/0.0071/8.0E−12| 9/0.0064/5.4E−12| 8.0/0.0078/9.4E−12|
| (4, 71)  | 2/0.028/4.1E−14 | 9.2/0.1/7.7E−12 | 10/0.12/3.4E−11| 8.4/0.13/7.7E−12 |
| (4, 100) | 2/0.12/1.8E−14 | 9.6/0.44/7.3E−12| 11/0.58/2.5E−12| 9.1/0.58/1.2E−11 |
| (4, 130) | 2/0.38/7.9E−15 | 10/1.3/1.4E−11 | 12/1.78/6.6E−15| 9.1/1.5/9.3E−12 |
| (5, 30)  | 2/0.027/1.9E−14 | 7.6/0.081/1.5E−11| 11/0.12/4.0E−11| 8.3/0.11/1.5E−11 |
| (5, 48)  | 2/0.39/2.6E−15 | 8.3/1.3/1.7E−11 | 13/2.4/1.2E−13 | 9/1.9/1.6E−11 |

Remark 5.1 Since $A$ is a strong M-tensor, there exists a positive vector $u$ such that $Au^{m−1} > 0$. This vector $u$ can be obtained in a certain iteration of solving $Ax^{m−1} = e$ by the existing methods proposed in [7,11,12]. Then we can get an initial point of Algorithm 3.1 or Algorithm 4.1 by letting $x_0 = tu$ and $y_0 = x_0^{[m−1]}$ with sufficient large constant $t$. Particularly, if $A$ is a diagonally dominant M-tensor, we can simply let $u = e$.

Note that strong M-tensors constructed in Problems 1, 2 and 4 are diagonally dominant M-tensors, whereas both Problems 3 and 5 are non-diagonally dominant M-tensors.

We first test the performance of Algorithm 3.1 (denoted by ‘NM’). In order to test the effectiveness of the proposed method, we compare the proposed Newton method with the Newton Method for symmetric M-equations (denoted by ‘SNM’) proposed in [7], the globally and quadratically convergent algorithm (denoted by ‘QCA’) proposed in [12] and the Inexact Newton Method (denoted by ‘INM’) proposed in [14]. We take the parameter of NM as $\epsilon = 0.1$, $\sigma = 0.1$ and $\rho = 0.5$. And let parameters of QCA and INM the same as in [12,14]. We set the initial point for INM as in [14], i.e., $y_0 = te > 0$ such that $f(y_0) \leq b$, where $t$ is a sufficient small positive constant. We use INM to find an initial point of NM and SNM for Problems 3 and 5 by Remark 5.1. The initial point of QCA is set to $e$.

For the stability of numerical results, we test the problems with different sizes. For each pair $(m, n)$, we randomly generate 50 tensors $A$ and $b \in (0, 1)$. The average results are listed in Tables 1, 2, 3, 4 and 5, where ‘Iter’ represents the average number of iterations; ‘Time’ denotes the computing time (in seconds) including initial time to find an approximate solution; ‘Res’ denotes the average residual at termination of algorithms.

It can be seen from Tables 1, 2, 3, 4 and 5 that the proposed NM method has an advantage over the other three methods both in the number of iteration and the CPU time. We notice that for Problems 1, 2 and 4, since the coefficient tensors are diagonally dominant, it is easy for NM to get an initial point. For those problems, the performance of the NM is significantly better than other methods. On the other hand, for Problems 3 and 5, although the number of iterations in NM was significantly less than that in
Table 2 Comparison on Problem 2

| (m, n) | NM Iter/Time/Res | INM Iter/Time/Res | SNM Iter/Time/Res | QCA Iter/Time/Res |
|--------|------------------|-------------------|-------------------|-------------------|
| (3, 200) | 3/0.012/3.8E-16 | 11/0.035/1.1E-11 | 11/0.037/3.9E-12 | 12/0.047/1.4E-11 |
| (3, 401) | 3/0.11/1.5E-16 | 12/0.4/9.7E-12 | 12/0.4/1.5E-12 | 12/0.44/1.1E-11 |
| (3, 650) | 3/0.48/6.4E-17 | 12/1.9/1.1E-11 | 13/2.9/7.6E-15 | 12/1.9/1.1E-11 |
| (4, 40) | 3/0.0036/2.1E-15 | 9.4/0.0074/7.0E-12 | 13/0.0094/1.9E-14 | 11/0.0095/9.4E-12 |
| (4, 71) | 3/0.036/4.5E-16 | 9.3/0.11/1.5E-11 | 14/2.8/5.6E-13 | 12/0.17/1.6E-11 |
| (4, 100) | 2.7/0.14/2.6E-11 | 9.1/0.52/1.2E-11 | 15/0.87/4.0E-14 | 12/0.72/1.8E-11 |
| (4, 130) | 2.0/0.36/5.6E-11 | 9.7/1.5/1.8E-11 | 15/2.4/1.6E-11 | 11/1.9/1.7E-11 |
| (5, 30) | 2.4/0.03/3.7E-11 | 8.4/0.089/9.2E-12 | 15/0.18/1.2E-11 | 11/0.15/1.5E-11 |
| (5, 48) | 2.0/0.37/2.5E-11 | 8/1.6/1.5E-11 | 17/3.3/7.1E-12 | 12/2.5/1.5E-11 |

Table 3 Comparison on Problem 3

| (m, n) | NM Iter/Time/Res | INM Iter/Time/Res | SNM Iter/Time/Res | QCA Iter/Time/Res |
|--------|------------------|-------------------|-------------------|-------------------|
| (4, 40) | 1/0.0029/1.5E-15 | 5/0.0046/6.6E-12 | 4/0.0046/1.0E-15 | 5/0.0055/2.1E-11 |
| (4, 71) | 1/0.041/1.6E-15 | 5/0.081/6.6E-12 | 4/0.083/1.4E-15 | 5/0.096/8.7E-11 |
| (4, 100) | 1/0.16/2.0E-15 | 5/0.32/6.6E-12 | 4/0.34/1.6E-15 | 8/0.47/1.8E-15 |
| (4, 130) | 1/0.45/2.0E-15 | 5/0.93/6.6E-12 | 4/0.93/1.8E-15 | 8/1.4/1.2E-14 |

Table 4 Comparison on Problem 4

| (m, n) | NM Iter/Time/Res | INM Iter/Time/Res | SNM Iter/Time/Res | QCA Iter/Time/Res |
|--------|------------------|-------------------|-------------------|-------------------|
| (3, 200) | 2/0.0089/1.3E-12 | 11/0.036/1.2E-11 | 8.1/0.028/4.1E-11 | 8.4/0.035/1.1E-11 |
| (3, 401) | 2/0.081/2.1E-13 | 12/0.38/5.3E-12 | 9/0.31/1.9E-12 | 9.4/0.36/3.7E-12 |
| (3, 650) | 2/0.36/7.1E-14 | 12/1.9/7.9E-12 | 10/1.6/2.0E-15 | 9.5/1.7/1.0E-11 |
| (4, 40) | 2/0.0027/4.5E-13 | 8.7/0.0065/1.5E-11 | 9.1/0.0064/2.4E-11 | 8.1/0.0072/1.3E-11 |
| (4, 71) | 2/0.027/6.0E-14 | 9.4/0.11/5.3E-12 | 10/1.4/3.5E-11 | 8.6/0.14/9.7E-12 |
| (4, 100) | 2/0.11/1.9E-14 | 9.4/0.53/1.4E-11 | 11/0.65/4.0E-12 | 8.8/0.58/7.9E-12 |
| (4, 130) | 2/0.35/9.8E-15 | 9.9/1.6/1.6E-11 | 12/1.9/1.8E-14 | 9.5/1.7/1.0E-11 |
| (5, 30) | 2/0.026/2.0E-14 | 7.7/0.08/1.7E-11 | 11/0.13/3.7E-11 | 8.5/0.12/1.2E-11 |
| (5, 48) | 2/0.38/3.3E-15 | 8.4/1.7/1.5E-11 | 13/2.6/2.3E-13 | 9/2/2.1E-11 |

INM, SNM and QCA, the reduction in the required CPU time was not proportional to the numbers of iterations because it took some CPU time to get an initial point.

We then tested the performance of Algorithm 4.1 (denoted by ‘ENM’) and compared it with the Regularized Newton Method (denoted by ‘RNM’) proposed in [14]. The parameters of ENM are set to $\epsilon = 0.1$, $\epsilon' = 0.05$, $\sigma = 0.1$ and $\rho = 0.5$. And the
Table 5 Comparison on Problem 5

| (m, n)   | NM | INM | SNM | QCA |
|----------|----|-----|-----|-----|
|          | Iter/Time/Res | Iter/Time/Res | Iter/Time/Res | Iter/Time/Res |
| (3, 200) | 2.9/0.03/7.9E−12 | 12/0.035/3.8E−12 | 3.4/0.034/9.7E−12 | 11/0.04/6.7E−12 |
| (3, 401) | 2.9/0.34/1.6E−12 | 12/0.43/6.7E−12 | 3.5/0.4/1.1E−11     | 11/0.39/8.7E−12 |
| (3, 650) | 2.9/1.8/2.1E−12   | 13/2.1/9.7E−12   | 3.7/2/6.4E−12       | 12/2/3.5E−12    |
| (4, 40)  | 2.9/0.0064/2.6E−12 | 9.1/0.0066/8.5E−12 | 3.6/0.0061/9.3E−12 | 8.4/0.0068/5.8E−12 |
| (4, 71)  | 2.8/0.092/5.6E−12 | 9.6/0.14/9.1E−12 | 3.5/0.14/4.9E−12   | 8.9/0.14/9.3E−12 |
| (4, 100) | 2.9/0.46/1.8E−12 | 10/0.6/4.1E−12   | 3.6/0.59/5.1E−12  | 9.4/0.59/1.0E−11 |
| (4, 130) | 2.9/1.5/2.3E−12 | 11/1.9/6.1E−12 | 3.7/1.8/3.6E−12   | 9.8/1.7/6.5E−12 |
| (5, 30)  | 2.8/0.074/3.7E−12 | 8.7/0.11/6.8E−12 | 3.6/0.11/6.6E−12 | 7.9/0.1/9.7E−12 |
| (5, 48)  | 2.8/1.4/7.0E−12 | 9.1/1.9/8.7E−12 | 3.6/1.8/1.9E−12   | 8.2/1.8/1.1E−11 |

parameters in RNM are the same as in [14], i.e., $\sigma = 0.1$, $\rho = 0.8$, $\gamma = 0.9$ and $\bar{t} = 0.01$.

For Problems 1, 2, 4 and 5, we generated $b \in \mathbb{R}^n_+$ randomly and then let some but not all elements of $b$ be 0 randomly. We let $b_1 > 0$ in Problem 5. It is easy to see that the strong M-tensor $A$ and $b$ constructed in our tested problems satisfy the conditions in Assumption 3.1. For Problem 5, we also use INM to find an initial point of ENM. The results are summarized in Tables 6, 7, 8 and 9, where ‘Time-Int’ denotes the average time to find an initial point of ENM and the ratio signs are denoted bellow.

| R-Int | RI | RT | RT1 |
|-------|----|----|-----|
| Time-Int | Iter of ENM | Time of ENM | Time of RNM |

It can be seen from the data that the proposed ENM is not only effective for solving the M-Teq with $b \geq 0$, but also more efficient than RNM to a certain extent.

6 Conclusions

We proposed a Newton method for solving the M-tensor equation with a positive constant term and establish its global and quadratic convergence. Then we extended the Newton method to solve the M-tensor equation with a nonnegative constant term and establish its convergence. Our numerical results show that the proposed methods are very efficient.

Acknowledgements This work was supported by the NSF of China grant No.11771157, 11801184, the NSF of Guangdong Province grant No.2020B1515310013 and the Education Department of Hunan Province grant No.20C0559.
## Table 6  Comparison on Problem 1

| $(m, n)$ | ENM | RNM | Rate |
|----------|-----|-----|------|
|          | Iter | Time | Iter | Time | RI (%) | RT (%) |
| (3, 200) | 2.4  | 0.01286 | 7.4  | 0.03194 | 32.3  | 40.3  |
| (3, 350) | 2.3  | 0.05659 | 7.2  | 0.18505 | 32.0  | 30.6  |
| (3, 500) | 2.2  | 0.16599 | 7.7  | 0.60120 | 28.1  | 27.6  |
| (3, 650) | 2.2  | 0.43180 | 7.8  | 1.38419 | 27.8  | 31.2  |
| (4, 40)  | 2.3  | 0.00467 | 7.4  | 0.00732 | 31.1  | 63.8  |
| (4, 90)  | 2.1  | 0.07785 | 8    | 0.27286 | 26.4  | 28.5  |
| (4, 130) | 2    | 0.38707 | 8.1  | 1.20770 | 25.2  | 32.1  |
| (5, 30)  | 2.1  | 0.02927 | 7.9  | 0.09237 | 26.2  | 31.7  |
| (5, 48)  | 2    | 0.38698 | 8.1  | 1.42453 | 24.6  | 27.2  |

## Table 7  Comparison on Problem 2

| $(m, n)$ | ENM | RNM | Rate |
|----------|-----|-----|------|
|          | Iter | Time | Iter | Time | RI (%) | RT (%) |
| (3, 200) | 3.5  | 0.01525 | 9.6  | 0.04179 | 36.7  | 36.5  |
| (3, 350) | 3.2  | 0.07255 | 10   | 0.23803 | 32.4  | 30.5  |
| (3, 500) | 3.2  | 0.22577 | 10   | 0.75133 | 32.3  | 30.0  |
| (3, 650) | 3.3  | 0.53518 | 10   | 1.82259 | 32.6  | 29.4  |
| (4, 40)  | 3.4  | 0.00394 | 9.6  | 0.00838 | 35.3  | 47.0  |
| (4, 90)  | 3.3  | 0.10645 | 10   | 0.40368 | 31.7  | 26.4  |
| (4, 130) | 3.2  | 0.48901 | 10   | 1.70984 | 30.9  | 28.6  |
| (5, 30)  | 3.3  | 0.03658 | 10   | 0.12137 | 32.7  | 30.1  |
| (5, 48)  | 3    | 0.52140 | 11   | 2.20545 | 28.1  | 23.6  |

## Table 8  Comparison on Problem 4

| $(m, n)$ | ENM | RNM | Rate |
|----------|-----|-----|------|
|          | Iter | Time | Iter | Time | RI (%) | RT (%) |
| (3, 200) | 2.6  | 0.01099 | 7.5  | 0.03303 | 34.5  | 33.3  |
| (3, 350) | 2.2  | 0.05440 | 7.6  | 0.19408 | 29.1  | 28.0  |
| (3, 500) | 2.2  | 0.16818 | 8    | 0.63763 | 27.4  | 26.4  |
| (3, 650) | 2.1  | 0.40086 | 8.1  | 1.64549 | 25.8  | 24.4  |
| (4, 40)  | 2.3  | 0.00477 | 7.4  | 0.01114 | 30.6  | 42.8  |
| (4, 90)  | 2    | 0.09399 | 7.7  | 0.28420 | 26.6  | 33.1  |
| (4, 130) | 2    | 0.35984 | 8.1  | 1.39153 | 24.9  | 25.9  |
| (5, 30)  | 2    | 0.02607 | 7.8  | 0.09134 | 26.0  | 28.5  |
| (5, 48)  | 2    | 0.37042 | 8.4  | 1.82102 | 24.1  | 20.3  |
Table 9 Comparison on Problem 5

| (m, n) | ENM | RNM | Rate |
|--------|-----|-----|------|
|        | Iter | Time | Time-Int | R-Int (%) | Iter | Time | RI (%) | RT (%) | RT1 (%) |
| (3, 200) | 4.3 | 0.02343 | 0.01179 | 50.3 | 10 | 0.03949 | 42.8 | 59.3 | 29.5 |
| (3, 350) | 4.2 | 0.14555 | 0.06743 | 46.3 | 10 | 0.24752 | 40.5 | 58.8 | 31.6 |
| (3, 500) | 4.3 | 0.48284 | 0.21440 | 44.4 | 11 | 0.80104 | 39.7 | 60.3 | 33.5 |
| (3, 650) | 4.1 | 1.10155 | 0.49533 | 45.0 | 11 | 1.87385 | 38.3 | 58.8 | 32.4 |
| (4, 40) | 4.4 | 0.00624 | 0.00328 | 52.5 | 8.3 | 0.00636 | 52.5 | 98.2 | 46.6 |
| (4, 90) | 4.6 | 0.25343 | 0.10136 | 40.0 | 9.2 | 0.39225 | 49.5 | 64.6 | 38.8 |
| (4, 130) | 4.4 | 1.13300 | 0.48959 | 43.2 | 9.5 | 1.64542 | 46.6 | 68.9 | 39.1 |
| (5, 30) | 4.4 | 0.07540 | 0.03641 | 48.3 | 8 | 0.10592 | 54.4 | 71.2 | 36.8 |
| (5, 48) | 4.5 | 1.40943 | 0.60182 | 42.7 | 8.5 | 1.86548 | 53.2 | 75.6 | 43.3 |
Data Availability Statements  All data generated or analyzed during this study are included in this published article.

References

1. Bader, B.W., Kolda, T.G., et al.: MATLAB tensor toolbox version 2.6 (2015)
2. Bai, X., He, H., Ling, C., Zhou, G.: An nonnegativity preserving algorithm for multilinear systems with nonsingular M-tensors. Numer. Algorithms. 87(3), 1301–1320 (2021)
3. Berman, A., Plummer, R.: Nonnegative Matrices in the Mathematical Sciences. SIAM, Philadelphia (1994)
4. Brazell, M., Li, N., Navasca, C., Tamon, C.: Solving multilinear systems via tensor inversion. SIAM J. Matrix Anal. Appl. 34(2), 542–570 (2013)
5. Chang, K.C., Pearson, K., Zhang, T.: Primitivity, the convergence of the NQZ method, and the largest eigenvalue for nonnegative tensors. SIAM. J. Matrix Anal. Appl. 32(3), 806–819 (2011)
6. Ding, W., Qi, L., Wei, Y.: M-tensors and nonsingular M-tensors. Linear Algebra Appl. 439 (10), 3264–3278 (2013)
7. Ding, W., Wei, Y.: Solving multi-linear systems with M-tensors. J. Sci. Comput. 68(2), 683–715 (2016)
8. Facchinei, F., Kanzow, C.: Beyond monotonicity in regularization methods for nonlinear complementarity problems. SIAM J. Control Optim. 37(4), 1150–1161 (1999)
9. Fan, H.Y., Zhang, L., Chu, E.K., Wei, Y.: Numerical solution to a linear equation with tensor product structure. Numer. Linear Algebra Appl. 24(6), e2106 (2017)
10. Gowda, M.S., Luo, Z., Qi, L., Xu, N.: Z-tensor and complementarity problems (2015). arXiv:1510.07933
11. Han, L.: A homotopy method for solving multilinear systems with M-tensors. Appl. Math. Lett. 69, 49–54 (2017)
12. He, H., Ling, C., Qi, L., Zhou, G.: A globally and quadratically convergent algorithm for solving multilinear systems with M-tensors. J. Sci. Comput. 76(3), 1718–1741 (2018)
13. Kressner, D., Tobler, C.: Krylov subspace methods for linear systems with tensor product structure. SIAM J. Matrix Anal. Appl. 31(4), 1688–1714 (2010)
14. Li, D.H., Guan, H.B., Xu, J.F.: Inexact Newton method for M-Tensor equations (2020). arXiv:2007.13324
15. Li, D.H., Guan, H.B., Wang, X.Z.: Finding a nonnegative solution to an M-tensor equation. Pac. J. Optim. 16(3), 419–440 (2020)
16. Li, D.H., Xie, S., Xu, H.R.: Splitting methods for tensor equations. Numer. Linear Algebra Appl. 24(5), e2102 (2017)
17. Li, Z., Dai, Y.-H., Gao, H.: Alternating projection method for a class of tensor equations. J. Comput. Appl. Math. 346, 490–504 (2019)
18. Li, X., Ng, M.K.: Solving sparse non-negative tensor equations: algorithms and applications. Front. Math. China 10(3), 649–680 (2015)
19. Liu, D., Li, W., Vong, S.W.: The tensor splitting with application to solve multi-linear systems. J. Comput. Appl. Math. 330, 75–94 (2018)
20. Lim, L.H.: Singular values and eigenvalues of tensors, a variational approach. In: Proceedings of the 1st IEEE International Workshop on Computational Advances of Multi-tensor Adaptive Processing, vol. 1, pp. 129–132 (2005)
21. Lv, C.Q., Ma, C.F.: A Levenberg–Marquardt method for solving semi-symmetric tensor equations. J. Comput. Appl. Math. 332, 13–25 (2018)
22. Qi, L.: Eigenvalues of a real supersymmetric tensor. J. Symbolic Comput. 40(6), 1302–1324 (2005)
23. Qi, L., Chen, H., Chen, Y.: Tensor Eigenvalues and Their Applications. Springer, Berlin (2018)
24. Qi, L., Luo, Z.: Tensor Analysis. Spectral Theory and Special Tensors. SIAM, Philadelphia (2017)
25. Wang, X., Che, M., Wei, Y.: Neural networks based approach solving multi-linear systems with M-tensors. Neural Comput. 351, 33–42 (2019)
26. Xie, Z.J., Jin, X.Q., Wei, Y.M.: Tensor methods for solving symmetric M-tensor systems. J. Sci. Comput. 74(1), 412–425 (2018)
27. Xie, Z.J., Jin, X.Q., Wei, Y.M.: A fast algorithm for solving circulant tensor systems. Linear Multilinear Algebra. 65(9), 1894–1904 (2017)

Springer
28. Xu, Y., Gu, W., Huang, Z.H.: Properties of the nonnegative solution set of multi-linear equations. Pac. J. Optim. 15(3), 441–456 (2019)
29. Yan, W., Ling, C., Ling, L., He, H.: Generalized tensor equations with leading structured tensors. Appl. Math. Comput. 361, 311–324 (2019)
30. Zhang, L., Qi, L., Zhou, G.: M-tensor and some applications. SIAM J. Matrix Anal. Appl. 35(2), 437–452 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.