Numer Algor (2019) 81:407–420
https://doi.org/10.1007/s11075-018-0554-7

ORIGINAL PAPER

A self-adaptive trust region method for extreme $\mathcal{B}$-eigenvalues of symmetric tensors

Mingyuan Cao[1] · **Qingdao Huang**[1] · **Yueting Yang**[2]

Received: 3 March 2017 / Accepted: 1 June 2018 / Published online: 14 June 2018

Abstract A self-adaptive trust region method is presented for finding the largest or smallest $\mathcal{B}$-eigenvalues of symmetric tensors. One of the important features of this method is that $\mathcal{B}$-eigenvalues problem of symmetric tensors is transformed into a homogenous polynomial optimization. Global convergence of the proposed algorithm and second-order necessary conditions of the optimal solutions are established, respectively. Numerical experiments are listed to illustrate the efficiency of the proposed method.

Keywords Symmetric tensors · Eigenvalues of tensors · Polynomial optimization · Trust region method · Global convergence

Mathematics Subject Classification (2010) 15A18 · 15A69 · 90C55

1 Introduction

Eigenvalues and eigenvectors of symmetric tensors were introduced in [14, 17, 19, 22–27, 29]. They have been applied in molecular conformation [7], blind source separation [15], quantum physics [31], high-order Markov chains [4], etc.

✉ Qingdao Huang
huangqd@jlu.edu.cn

Mingyuan Cao
caomingyuan0918@163.com

Yueting Yang
yangyueting@163.com

[1] College of Mathematics, Jilin University, No. 2699, Qianjin Street, Changchun 130000, China

[2] College of Mathematics and Statistics, Beihua University, Jilin 132013, China

Let $\mathbb{C}(\mathbb{R})$ be the complex (real) field. An mth-order, n-dimensional real tensor $\mathcal{A}$ is expressed as

$$\mathcal{A} = (a_{i_1 i_2 \cdots i_m}),\ a_{i_1 i_2 \cdots i_m} \in \mathbb{R}, 1 \le i_1, \cdots, i_m \le n,$$

and x is a real-valued n-vector. $\mathcal{A}x^{m-1}$ denotes an n-vector with its ith component as

$$(\mathcal{A}x^{m-1})_i = \sum_{i_2, \cdots, i_m = 1}^{n} a_{i i_2 \cdots i_m} x_{i_2} \cdots x_{i_m}, \quad \text{for } i = 1, 2, \cdots, n.$$

$\mathcal{A}x^m$ is a scalar defined by

$$\mathcal{A}x^m = \sum_{i_1, i_2, \cdots, i_m = 1}^{n} a_{i_1 i_2 \cdots i_m} x_{i_1} \cdots x_{i_m},$$

i.e., $\mathcal{A}x^m = x^{\mathrm{T}} \mathcal{A}x^{m-1}$ [25]. Tensor $\mathcal{A}$ is positive definite if $\mathcal{A}x^m > 0$ for all $x \neq 0$ and is symmetric if its entries $a_{i_1 i_2 \cdots i_m}$ are invariant under any permutation of $i_1, \cdots, i_m$. Following [2], if $\mathcal{A}$ is symmetric, $\mathcal{A}x^m$ satisfies

$$\nabla(\mathcal{A}x^m) = m\mathcal{A}x^{m-1}.$$

In this paper, only symmetric tensors are under considered.

For a tensor $\mathcal{A}$, if there exist $\lambda \in \mathbb{R}$ and $x \in \mathbb{R}^n$ satisfying

$$\begin{aligned} \mathcal{A}x^{m-1} &= \lambda x, \\ x^T x &= 1, \end{aligned} \tag{1}$$

then λ is called a Z-eigenvalue of $\mathcal{A}$ and x is called the corresponding Z-eigenvector, (λ, x) is called a Z-eigenpairs of $\mathcal{A}$ [25].

If there exist $\lambda \in \mathbb{R}$ and a nonzero vector $x \in \mathbb{R}^n$ satisfying

$$\mathcal{A}x^{m-1} = \lambda x^{[m-1]}, \tag{2}$$

where $x^{[\alpha]} = [x_1^\alpha, x_2^\alpha, \cdots, x_n^\alpha]^{\mathrm{T}}$. Then λ is called an H-eigenvalue of $\mathcal{A}$ and x is called the corresponding H-eigenvector, (λ, x) is called an H-eigenpairs [16]. Obviously, (2) is a homogeneous polynomial equation.

Qi et al. [28] gave some Z-eigenvalues methods for solving a global polynomial optimization problem. In this work, they proposed a direct Z-eigenvalues method when the dimension is two, and a direct orthogonal transformation Z-eigenvalues method in the case of third-order three-dimension. For general symmetric tensors, a shifted high-order power method (PM) was presented for computing Z-eigenpairs in [13]. Forced the objective to be (locally) convex/concave by a shift, convergency of the proposed method was proved. Using fixed point analysis, the linear convergence rate of PM is obtained. However, similarly to the case of matrix, the convergence of PM is very slow if the largest eigenvalue is close to the second dominant eigenvalue [5].

Hao et al. [9] found extreme Z-eigenvalues and the corresponding Z-eigenvectors by the sequential subspace projection method (SSPM). Under certain assumptions, global convergence and linear convergence were established for supersymmetric

tensors. Preliminary numerical results showed that the SSPM is promising. Hao et al. [10] proposed a feasible trust region method (FTR) for Z-eigenvalues of symmetric tensors. Global convergence and local quadratic convergence of the FTR method were established. Compared with PM and SSPM in their numerical experiments, FTR obtained the extreme Z-eigenvalues with a higher probability.

The power method have been successfully generalized to find the largest H-eigenvalue of a nonnegative irreducible tensor [21]. The convergence of the method was established by Chang et al. [3] under primitivity, and by Fridland et al. [8] under weak primitivity. The linear convergence rate of the method was achieved by Zhang and Qi [35]. The R-linear convergence rate of the method was obtained by Hu et al. [12] under weak primitivity. Li et al. [18] showed that the largest H-eigenvalue function of a symmetric tensor of even order is convex, continuous and semismooth on the tensor space.

Cui et al. [1] introduced $\mathcal{B}$-eigenvalues of symmetric tensors as follows

$$\begin{aligned} \mathcal{A}x^{m-1} &= \lambda \mathcal{B}x^{l-1}, \\ \mathcal{B}x^{l} &= 1, \end{aligned} \tag{3}$$

where l is positive integer, m is even, $\mathcal{B}$ is an $lth-$order, n-dimensional symmetric tensor and is positive definite. $\lambda \in \mathbb{C}$ and $x \in \mathbb{C}^n$ are called $\mathcal{B}$-eigenvalues and the corresponding $\mathcal{B}$-eigenvectors of $(\mathcal{A}, \mathcal{B})$ and $(\lambda, x) \in \mathbb{C} \times \mathbb{C}^n \setminus \{0\}$ is called a $\mathcal{B}$-eigenpairs of $(\mathcal{A}, \mathcal{B})$.

If $l = 2$ and $\mathcal{B}$ is an identity matrix, then (3) reduces to (1).

If $l = m$, $\mathcal{B}$ satisfies $\mathcal{B}x^{m-1} = [x_1^{m-1}, x_2^{m-1}, \cdots, x_n^{m-1}]^{\mathrm{T}}$ and $\lambda \in \mathbb{R}$, then (3) reduces to (2).

Therefore, the problem (3) can be regarded as a unified form of H-eigenvalues [25], Z-eigenvalues [25], and D-eigenvalues [29].

Motivated by the work of [10], we transform the $\mathcal{B}$-eigenvalues problem into a homogenous polynomial optimization on a unit hyper-sphere $\mathcal{B}x^l = 1$. Besides, we propose a new trust region algorithm with self-adaptive technique for computing extreme $\mathcal{B}$-eigenvalues in this work. A new self-adaptive rule is used to update the trust region radius and efficiently dealt with the so-called too successful iteration. This strategy could be expected to improve the numerical performance of the algorithm. Meanwhile, we show the global convergence of the proposed method.

The rest of this paper is organized as follows. In Section 2, a self-adaptive trust region algorithm (SATR) for $\mathcal{B}$-eigenvalues of (3) is proposed. Global convergence results are established in Section 3. Numerical experiments are shown in Section 4. The conclusions are drawn in Section 5.

2 Self-adaptive trust region algorithm for $\mathcal{B}$-eigenvalues

Motivated by the idea of the trust region method for calculating extreme Z-eigenvalues of symmetric tensors in [10], we propose a self-adaptive trust region algorithm to compute $\mathcal{B}$-eigenpairs (3).

First, we reformulate (3) into a constrained optimization problem by the variational principle,

$$\begin{aligned} &\max_{x\in\mathbb{R}^n} \ \mathcal{A}x^m, \\ &\text{s.t.} \quad \mathcal{B}x^l = 1, \end{aligned} \tag{4}$$

with $(\mathcal{A}x^m, x)$ being a $\mathcal{B}$-eigenpairs of $(\mathcal{A}, \mathcal{B})$, where $\mathcal{A}x^m = \sum_{i_1,i_2,\cdots,i_m=1}^{n} a_{i_1 i_2 \cdots i_m} x_{i_1} \cdots x_{i_m}$, $\mathcal{B}x^l = \sum_{i_1,i_2,\cdots,i_l=1}^{n} b_{i_1 i_2 \cdots i_l} x_{i_1} \cdots x_{i_l}$.

We propose a self-adaptive trust region algorithm (SATR) for solving (4). The problem (4) can be rewritten as follows:

$$\begin{aligned} &\max_{x\in\mathbb{R}^n} \ f(x) = \tfrac{1}{m}\mathcal{A}x^m, \\ &\text{s.t.} \quad \tfrac{1}{l}(\mathcal{B}x^l - 1) = 0. \end{aligned} \tag{5}$$

Consider the Lagrangian function of (5),

$$L(x, \lambda) = f(x) - \frac{\lambda}{l}(\mathcal{B}x^l - 1). \tag{6}$$

It is clear that the KKT point x^* and the related Lagrangian multiplier $\lambda^* = \nabla f(x^*)^{\mathrm{T}} x^*$ of (5) exactly form a $\mathcal{B}$-eigenpairs (λ^*, x^*) of $(\mathcal{A}, \mathcal{B})$. The search direction d_k of a trust region type is obtained by solving the following trust region subproblem at the current feasible point x_k,

$$\begin{aligned} &\max_{d\in\mathbb{R}^n} \ m_k(d) = f_k + g_k^{\mathrm{T}} d + \tfrac{1}{2} d^{\mathrm{T}} G_k d, \\ &\text{s.t.} \quad (\mathcal{B}x_k^{l-1})^{\mathrm{T}} d = 0, \\ &\qquad \|d\| \le \Delta_k, \end{aligned} \tag{7}$$

where

$$\begin{aligned} &f_k = f(x_k), \\ &g_k = g(x_k) = \nabla f(x_k) - \lambda_k \mathcal{B}x_k^{l-1}, \\ &G_k = G(x_k) = \nabla^2 f(x_k) - (l-1)\lambda_k \mathcal{B}x_k^{l-2}, \end{aligned}$$

are the function value, gradient, and Hessian of $L(x, \lambda)$ at (x_k, λ_k), respectively. Δ_k is the trust region radius. At each iteration, if the trial step d_k is accepted, the iteration $x_k + d_k$ is enforced to be feasible by setting $x_{k+1} = p(x_k + d_k)$, where

$$p(x) = \frac{x}{(\mathcal{B}x^l)^{1/l}}. \tag{8}$$

And for the value of λ_k, we have

$$\lambda_k = \nabla f(x_k)^{\mathrm{T}} x_k = \mathcal{A}x_k^m. \tag{9}$$

Assume that U_k is the $n \times (n-1)$ column orthogonal matrix as the basis of the null space $N_{x_k} = \{d \mid (\mathcal{B}x_k^{l-1})^{\mathrm{T}} d = 0\}$ of x_k. Let $q = U_k^{\mathrm{T}} d$, we have $d = U_k q$, $\|d\| = \|q\|$, then using the null space method, we obtain the reduced trust region model of subproblem (7)

$$\begin{aligned} &\max_{q\in\mathbb{R}^{n-1}} \quad \tilde{m}_k(q) = f_k + \tilde{g}_k^{\mathrm{T}} q + \tfrac{1}{2} q^{\mathrm{T}} \tilde{G}_k q, \\ &\qquad \text{s.t.} \ \|q\| \le \Delta_k, \end{aligned} \tag{10}$$

where $\tilde{g}_k = U_k^{\mathrm{T}} g_k$, and $\tilde{G}_k = U_k^{\mathrm{T}} G_k U_k \in \mathbb{R}^{(n-1)\times(n-1)}$. By simple derivation, we obtain $m_k(d) = \tilde{m}_k(q)$ if $q = U_k^{\mathrm{T}} d$. It contributes to show the global convergence of the method.

After obtaining a trial step d_k of (7), whether the trail point $x_k + d_k$ is accepted and how to adjust the trust region radius depend on the ratio

$$r_k = \frac{f(p(x_k + d_k)) - f(x_k)}{m_k(d_k) - m_k(0)}, \tag{11}$$

where the numerator and the denominator are called the actual increase and the predicted increase, respectively. Since $q = 0$ is in the feasible region of the problem (10), the predicted increase always be nonnegative. Thus, if r_k is negative, the new objective value $f(p(x_k + d_k))$ is smaller than the current value $f(p(x_k))$, therefore the step must be rejected. Otherwise, if r_k is close to 1, there is a good agreement between the model $m_k(d)$ in (7) and the function f in (5) over this step. At this point, it is safe to enlarge the trust region for the next iteration. If r_k is positive but significantly smaller than 1, the parameter Δ_k should increase with ratio r_k increasing. If r_k is close to zero or negative, we shrink the trust region by reducing Δ_k at the next iteration.

In this paper, we consider a particular case that r_k is significantly larger than one, the iteration is called too successful iterations [6, 11, 30, 32]. It implies that the local approximation of the objective function by the model function is not good. In order to overcome this shortcoming, Lu et al. [20] introduced the L-function $L(r_k)$ to update Δ_k. That is

$$L(r_k) = \begin{cases} 1 + e^{-\frac{r_k+\eta_1-2}{(\eta_1-2)^2}}, & \text{if } r_k \geq 2-\eta_1; \\ 2, & \text{if } \eta_1 \leq r_k < 2-\eta_1; \\ \frac{1-1/2e^{\eta_1}}{1-e^{\eta_1}} - \frac{e^{\eta_1}}{2} e^{\frac{r_k-\eta_1}{1-e^{\eta_1}}}, & \text{if } 0 < r_k < \eta_1; \\ 0.125, & \text{if } r_k \leq 0. \end{cases}$$

And then, let

$$\Delta_{k+1} = L(r_k)\Delta_k. \tag{12}$$

The authors showed that the L-function $L(r_k)$ can adjust the trust region radius adaptively and can deal with the situation of "too successful" as well.

Now we describe a self-adaptive trust region algorithm (SATR) for $\mathcal{B}$-eigenvalues (3) using the self-adaptive technique (12). Denote the set of all real symmetric mth-order n-dimensional tensors by $\mathcal{S}^{[m,n]}$. $\mathcal{S}_+^{[m,n]}$ is the set of all real symmetric positive definite mth-order n-dimensional tensors.

Algorithm 2.1 (SATR)

Step 1: Given $x_0 \in \mathbb{R}^n$, $\mathcal{A} \in S^{[m,n]}$, $\mathcal{B} \in S_+^{[l,n]}$, $0 < \eta_2 < \eta_1 < 1$, $\Delta_0 > 0$, $\varepsilon \geq 0$ and $\lambda_0 = \mathcal{A}x_0^m$. Set $k := 0$.

Step 2: Compute g_k, G_k, U_k, $\tilde{g}_k$ and $\tilde{G}_k$.

Step 3: Solve the subproblem (10) to get q_k. Let $d_k = U_k q_k$.

Step 4: If $\|d_k\| \leq \varepsilon$, stop, output (λ_k, x_k).

Step 5: Compute r_k by (11). If $r_k \geq \eta_2$, set $x_{k+1} = p(x_k + d_k)$ and $\lambda_{k+1} = \mathcal{A}x_{k+1}^m$; else $x_{k+1} = x_k$ and $\lambda_{k+1} = \lambda_k$.

Step 6: Update the trust region radius Δ_{k+1} by using (12). Set $k := k + 1$ and go to step 2.

In Step 3, the CG-Steihaug algorithm in [24] is used to solve the subproblem (10). One good property of the CG-Steihaug algorithm is that the solution computed has a sufficient increase property, which was proved by Yuan [34]. Moreover, the CG-Steihaug algorithm is suitable for solving large scale problem [33]. And the global convergence of the corresponding trust region method can be ensured.

3 Convergence of algorithm 2.1

In this section, we prove the global convergence of Algorithm 2.1.

To facilitate analyzing, let $h(x) = f(p(x)) = f(\frac{x}{(\mathcal{B}x^l)^{1/l}})$. The gradient and Hessian of $h(x)$ are

$$\begin{aligned}
\nabla h(x) &= \left(\frac{I}{(\mathcal{B}x^l)^{1/l}} - \frac{x(\mathcal{B}x^{l-1})^{\mathrm{T}}}{(\mathcal{B}x^l)^{1+1/l}}\right)\nabla f(p(x)),\\
\nabla^2 h(x) &= \frac{\nabla^2 f(p(x))\nabla p(x)}{(\mathcal{B}x^l)^{1/l}} - \frac{(\mathcal{B}x^{l-1})\nabla f(p(x))^{\mathrm{T}}}{(\mathcal{B}x^l)^{1+1/l}} - \frac{(\mathcal{B}x^{l-1})^{\mathrm{T}}\nabla f(p(x))I}{(\mathcal{B}x^l)^{1+1/l}}\\
&\quad - \frac{(l-1)(\mathcal{B}x^{l-2})\nabla f(p(x))x^{\mathrm{T}} + x(\mathcal{B}x^{l-1})^{\mathrm{T}}\nabla^2 f(p(x))\nabla p(x)}{(\mathcal{B}x^l)^{1+1/l}}\\
&\quad + \frac{(l+1)x(\mathcal{B}x^{l-1})^{\mathrm{T}}\nabla f(p(x))^{\mathrm{T}}(\mathcal{B}x^{l-1})}{(\mathcal{B}x^l)^{2(1+1/l)}},
\end{aligned}$$

respectively.

Now, some lemmas are given to prove the global convergence of Algorithm 2.1.

Lemma 3.1 (i) *For all x, y satisfying $x^{\mathrm{T}}\mathcal{B}x^{l-1} = 1$, $y^{\mathrm{T}}\mathcal{B}y^{l-1} = 1$, then*

$$\begin{aligned}
&\|G(x)\| \le M,\\
&\|g(x) - g(y)\| \le L_0\|x - y\|,\\
&\|G(x) - G(y)\| \le L_1\|x - y\|,
\end{aligned} \tag{13}$$

where M, L_0, and L_1 are positive constants.

(ii) *There exists a bounded set of Ω, for all x, y satisfying $x^{\mathrm{T}}\mathcal{B}x^{l-1} \in \Omega$, $y^{\mathrm{T}}\mathcal{B}y^{l-1} \in \Omega$, then*

$$\|\nabla^2 h(x) - \nabla^2 h(y)\| \le L_2\|x - y\|,$$

where L_2 is a positive constant.

Proof It is obvious that $g(x)$, $G(x)$, and $\nabla^2 h(x)$ are smooth and bounded, which implies that the proof is completed. □

Lemma 3.2 *Consider the error between the function $h(x_k + d_k)$ and the model $m_k(d_k)$, then*

$$|m_k(d_k) - h(x_k + d_k)| \leq \frac{1}{2} L_2 \|d_k\|^3. \tag{14}$$

Proof From Taylor theorem, we have

$$h(x_k + d_k) = h(x_k) + \nabla h(x_k)^{\mathrm{T}} d_k + \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 h(x_k + \theta_k d_k) d_k, \ \theta_k \in (0, 1).$$

By using $(\mathcal{B} x_k^{l-1})^{\mathrm{T}} d_k = 0$, we obtain $\nabla h(x_k)^{\mathrm{T}} d_k = \nabla f(x_k)^{\mathrm{T}} [I - x_k (\mathcal{B} x_k^{l-1})^{\mathrm{T}}] d_k = \nabla f(x_k)^{\mathrm{T}} d_k$, $g_k^{\mathrm{T}} d_k = [\nabla f(x_k)^{\mathrm{T}} - \lambda_k (\mathcal{B} x_k^{l-1})^{\mathrm{T}}] d_k = \nabla f(x_k)^{\mathrm{T}} d_k$. Then $\nabla h(x_k)^{\mathrm{T}} d_k = g_k^{\mathrm{T}} d_k$. It follows from (7) and Lemma 3.1 that

$$\begin{aligned} |m_k(d_k) - h(x_k + d_k)| &= |\frac{1}{2} d_k^{\mathrm{T}} G_k d_k - \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 h(x_k + \theta_k d_k) d_k| \\ &= |\frac{1}{2} d_k^{\mathrm{T}} \nabla^2 h(x_k) d_k - \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 h(x_k + \theta_k d_k) d_k| \\ &\leq \frac{1}{2} L_2 \|d_k\|^3, \end{aligned}$$

which completes the proof. □

Lemma 3.3 *The predicted increase of the problem (7) satisfies*

$$m_k(d_k) - m_k(0) \geq \sigma \|g_k\| \min\{\Delta_k, \frac{\|g_k\|}{\|G_k\|}\}. \tag{15}$$

Proof The proof of Lemma 3.3 is similar to that in [10], so, here it is omitted. □

Lemma 3.4 *Let the sequence $\{x_k\}$ be generated by Algorithm 2.1. If there is a positive number ε satisfying $\|g_k\| \geq \varepsilon$ for all k, then there exists a positive constant $\bar{\Delta}$, for all k such that*

$$\Delta_k \geq \bar{\Delta}. \tag{16}$$

Proof Suppose that the conclusion is not true, there exists an infinite set $\mathcal{K} \subset N$, such that

$$\lim_{k \in \mathcal{K}, k \to \infty} \Delta_k = 0. \tag{17}$$

Since $\|g_k\| \geq \varepsilon$ for all k, by Lemma 3.1 and Lemma 3.3, we have

$$m_k(d_k) - m_k(0) \geq \sigma \|g_k\| \min\{\Delta_k, \frac{\|g_k\|}{\|G_k\|}\} \geq \sigma \varepsilon \min\{\Delta_k, \frac{\varepsilon}{M}\}. \tag{18}$$

From (14) and (18),

$$\begin{aligned} |r_k - 1| &= |\frac{f(p(x_k + d_k)) - f(x_k)}{m_k(d_k) - m_k(0)} - 1| \\ &= |\frac{h(x_k + d_k) - m_k(d_k)}{m_k(d_k) - m_k(0)}| \\ &\leq \frac{\frac{1}{2}L_2\|d_k\|^3}{\sigma\varepsilon\min\{\Delta_k, \frac{\varepsilon}{M}\}} \\ &\leq \frac{\frac{1}{2}L_2\Delta_k^3}{\sigma\varepsilon\min\{\Delta_k, \frac{\varepsilon}{M}\}}. \end{aligned} \tag{19}$$

Taking limit from both side of (19), and from (17), we have

$$\lim_{k\in\mathcal{K}, k\to\infty} r_k = 1. \tag{20}$$

By (12), we know that Δ_{k+1} will not shrink, i.e., $\Delta_{k+1} \geq \Delta_k$ for all sufficiently large k, which contradicts with (17). Therefore, (16) is true. □

Theorem 3.1 *Let the sequence $\{x_k\}$ be generated by Algorithm 2.1. Then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{21}$$

Proof The proof is deduced by contradiction. There exists a constant $\varepsilon > 0$, such that

$$\|g_k\| \geq \varepsilon, \text{ for all } k. \tag{22}$$

From (20), for sufficiently large k, we have $r_k \geq \eta_1$. Combined with (15) and (22), then

$$\begin{aligned} h(x_{k+1}) - h(x_k) &= r_k[m_k(d_k) - m_k(0)] \\ &\geq \eta_1[m_k(d_k) - m_k(0)] \\ &\geq \sigma\eta_1\|g_k\|\min\{\Delta_k, \frac{\|g_k\|}{\|G_k\|}\} \\ &\geq \sigma\eta_1\varepsilon\min\{\Delta_k, \frac{\varepsilon}{\tau}\}, \end{aligned} \tag{23}$$

where $\tau = \max\{1 + \|G_k\|\}$. Since $\{h(x_k)\}$ is monotonic increase and bounded over the hyper-sphere, and $h(x_{k+1}) - h(x_k) \to 0$, by the inequality from above , then

$$\lim_{k\to\infty} \Delta_k = 0,$$

which contradicts to Lemma 3.4. Hence (21) holds. □

Theorem 3.2 *Let the sequence $\{x_k\}$ be generated by Algorithm 2.1. Then*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{24}$$

Moreover, for any cluster point x^* of $\{x_k\}$, the second-order necessary conditions hold, i.e., $g(x^*) = 0$ and

$$d^{\mathrm{T}} G^* d \leq 0 \tag{25}$$

for all vectors d satisfying $d^{\mathrm{T}}(\mathcal{B}x^{*(l-1)}) = 0$, where $G^* = G(x^*)$.

Proof Consider any index t satisfying $\|g_t\| \neq 0$. For every point x in the ball $B(x_t, \delta) = \{x \mid x^{\mathrm{T}}\mathcal{B}x^{l-1} = 1, \|x - x_t\| \leq \delta\}$, from (i) in Lemma 3.1 we obtain

$$\|g(x)\| \geq \|g_t\| - \|g(x) - g_t\| \geq \|g_t\| - L_0\|x - x_t\|, \ x \in B(x_t, \delta).$$

Set $\delta = \frac{\|g_t\|}{2L_0}$, we have

$$\|g(x)\| \geq \|g_t\| - L_0\delta = \frac{1}{2}\|g_t\|. \tag{26}$$

Therefore, the ball $B(x_t, \delta)$ can not contain the whole sequence $\{x_k\}$ from (21) in Theorem 3.1.

Suppose that $s \geq t$ and x_{s+1} is the first iteration after x_t outside $B(x_t, \delta)$. We have

$$\{x_t, \cdots, x_s\} \in B(x_t, \delta)$$

and

$$\begin{aligned} f(x_{s+1}) - f(x_t) &= \sum_{k=t}^{s}(f(x_{k+1}) - f(x_k)) \\ &\geq \sum_{k=t, x_k \neq x_{k+1}}^{s} \eta_2(m_k(d) - m_k(0)) \\ &\geq \sum_{k=t, x_k \neq x_{k+1}}^{s} \eta_2\sigma\|g_k\| \min\{\Delta_k, \frac{\|g_k\|}{\|G_k\|}\} \\ &\geq \sum_{k=t, x_k \neq x_{k+1}}^{s} \frac{1}{2}\eta_2\sigma\|g_t\| \min\{\Delta_k, \frac{\|g_t\|}{2M}\}, \end{aligned}$$

where the above inequalities hold by Step 5 in Algorithm 2.1, (15) and (26), respectively. If $\Delta_k > \frac{\|g_t\|}{2M}$, for all $k = t, \cdots, s$,

$$f(x_{s+1}) - f(x_t) \geq \frac{1}{4M}\eta_2\sigma\|g_t\|^2. \tag{27}$$

If $\Delta_k \leq \frac{\|g_t\|}{2M}$, for all $k = t, \cdots, s$, we have

$$\begin{aligned} f(x_{s+1}) - f(x_t) &\geq \frac{1}{2}\eta_2\sigma\|g_t\| \sum_{k=t, x_k \neq x_{k+1}}^{s} \Delta_k \geq \frac{1}{2}\eta_2\sigma\|g_t\|\delta \\ &= \frac{1}{4L_0}\eta_2\sigma\|g_t\|^2. \end{aligned} \tag{28}$$

Since $f(x)$ is smooth on the unit hyper-sphere, the value $f(x^*)$ must be finite . As the sequence $f(x_k)$ is nondecreasing, we have $f(x^*) - f(x_k) \to 0$, then

$$0 \leftarrow f(x^*) - f(x_t) \geq f(x_{s+1}) - f(x_t) \geq \eta_2 \sigma \|g_t\|^2 \min\{\frac{1}{4M}, \frac{1}{4L_0}\}.$$

Therefore, $\lim_{t\to\infty} \|g_t\| = 0$, i.e., any cluster point x^* of $\{x_k\}$ satisfies $g(x^*) = 0$.

Now, we prove (25) by contradiction. Suppose that there exists a positive eigenvalue ξ such that

$$\omega^{\mathrm{T}} G^* \omega = \xi, \text{ where } \omega^{\mathrm{T}} \mathcal{B} x^{*(l-1)} = 0, \|\omega\| = 1. \tag{29}$$

Assume that $g_k^{\mathrm{T}} \omega \geq 0$, otherwise let $\omega = -\omega$. Consider $d = t\omega$ for some $0 \leq t \leq \Delta_k$, then

$$\begin{aligned} m_k(d) - m_k(0) &= g_k^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} G_k d = t g_k^{\mathrm{T}} \omega + \frac{1}{2} t^2 \omega^{\mathrm{T}} G_k \omega \\ &\geq \frac{1}{2} \|d\|^2 \omega^{\mathrm{T}} G_k \omega. \end{aligned} \tag{30}$$

From Lemma 3.1, there exists an infinite subsequence $\mathcal{T}$, satisfying $k \in \mathcal{T}$, for a convergent subsequence $\{x_k\}$ with $G_k \to G^*$, if x_k is sufficiently close to x^*, we have $\omega^{\mathrm{T}} G_k \omega \to \xi$.

Consider the following two cases. The one case is $r_k \geq \eta_1,\ k \in \mathcal{T}$. By (30), then

$$f(x_{k+1}) - f(x_k) \geq \eta_1 (m_k(d) - m_k(0)) \geq \frac{1}{4} \eta_1 \xi \|d_k\|^2,\ k \in \mathcal{T}. \tag{31}$$

It follows from $f(x_{k+1}) - f(x_k) \to 0$ that $\|d_k\| \to 0$. Therefore, $d_k = 0$ will be the optimal solution of (7) for sufficiently large k, namely, $q_k = 0$ is the optimal solution of (10). We have that $\tilde{g}_k = 0$ and $\tilde{G}_k$ is negative semidefinite. When k is sufficiently large, then $\tilde{g}^* = 0, \tilde{G}^*$ is negative semidefinite. This is contradict to (29). For the other case, if $r_k < \eta_1,\ k \in \mathcal{T}$, we have $\lim_{k\to\infty} \Delta_k = 0$. In fact

$$|1 - r_k| = |\frac{h(x_k + d_k) - m_k(d_k)}{m_k(d_k) - m_k(0)}|,$$

using (14) and (30) indicates that $r_k \to 1$, which contradicts to $r_k < \eta_1$.

For both cases, we all get contradiction (25) is obtained. The proof is completed. □

4 Numerical experiments

This section mainly includes two aspects of numerical experiments conclusions. One part is to compare the SATR method with the FTR method if $\mathcal{B}$ is chosen as an identity matrix ($l = 2$) in the problem (3). In this case, the problem (3) reduces to (2). On the other hand, we consider the case of $l > 2$. We list the numerical results for solving $\mathcal{B}$-eigenvalues of (3) to illustrate the efficiency of the proposed method.

All experiments were done on a PC with CPU 2.40GHz and 2.00GB RAM using MATLAB R2009b. In the implementation of Algorithm 2.1, we set parameters $\Delta_0 = 2, \varepsilon = 10^{-5}, \eta_1 = 0.75, \eta_2 = 0.1$. The parameters $\eta_1 = 0.75$ and $\eta_2 = 0.1$

are chosen according to experience. One could also set them by oneself. Other suggestions were recommended in [24]. All methods share the same start points and stopping criterion.

The basis matrix U_k is generated by the Householder transformation. If $(x_k)_1$ is the first element of $x_k \in \mathbb{R}^n$, $e_1 = (1, 0, 0, \cdots, 0)^{\mathrm{T}}$, then we have the Householder transformation matrix H_k satisfies $H_k^2 = I$ and $H_k x_k = -\mathrm{sign}((x_k)_1)e_1$. Moreover,

$$H_k = I - 2\omega_k \omega_k^{\mathrm{T}}, \text{ where } \omega_k = \frac{x_k + \mathrm{sign}((x_k)_1)e_1}{\|x_k + \mathrm{sign}((x_k)_1)e_1\|}.$$

The matrix U_k can be chosen as following

$$[-\mathrm{sign}((x_k)_1)x_k, U_k] = H_k.$$

The tensor $\mathcal{A}$ of Example 1 is originally from [15] and the tensors $\mathcal{A}$ of examples 2–4 are originally from [23]. Hao et al. [10] utilized the examples 3 and 4 showing the effective of FTR algorithm for solving Z-eigenvalues of symmetric tensors.

Example 1 Let $\mathcal{A} \in S^{[4,3]}$ defined by

$$\begin{aligned} &a_{1111} = 0.2883,\ a_{1112} = -0.0031,\ a_{1113} = 0.1973,\ a_{1122} = -0.2485,\\ &a_{1223} = 0.1862,\ a_{1133} = 0.3847,\ \ a_{1222} = 0.2972,\ a_{1123} = -0.2939,\\ &a_{1233} = 0.0919,\ a_{1333} = -0.3619,\ a_{2222} = 0.1241,\ a_{2223} = -0.3420,\\ &\qquad a_{2233} = 0.2127,\ \ a_{2333} = 0.2727,\ a_{3333} = -0.3054. \end{aligned}$$

Example 2

$$\mathcal{A}_{i_1,i_2,i_3,i_4} = \frac{(-1)^{i_1}}{i_1} + \frac{(-1)^{i_2}}{i_2} + \frac{(-1)^{i_3}}{i_3} + \frac{(-1)^{i_4}}{i_4}.$$

Example 3

$$\mathcal{A}_{i_1,i_2,i_3,i_4} = \arctan((-1)^{i_1}\frac{i_1}{n}) + \cdots + \arctan((-1)^{i_4}\frac{i_4}{n}).$$

Example 4

$$\mathcal{A}_{i_1,i_2,i_3} = \frac{(-1)^{i_1}}{i_1} + \frac{(-1)^{i_2}}{i_2} + \frac{(-1)^{i_3}}{i_3}.$$

The numerical results are listed in Tables 1 and 2. $Ex.$ is the number of example. m records the order and n is the dimension. k is the number of iterations, $\lambda_{\max}$ means the largest Z-eigenvalue returned by SATR or FTR in Table 1, $\lambda'_{\max}$ is the largest $\mathcal{B}$-eigenvalue obtained by SATR in Table 2. CPU stands for the CPU time (in seconds).

In Table 1, we compare the numerical results of SATR and FTR, when we choose $\mathcal{B}$ as an identity matrix in (3).

As we can see in Table 1, the two methods reach the same Z-eigenvalues for problems with different dimensions. SATR is faster than FTR for Examples 1–4. The number of iterations of SATR is less than that of FTR. For the cases that $n = 100$, $n = 200$, and $n = 500$, the CPU time of SATR is better than that of FTR. Particularly, with the dimension n increased, the superiority of SATR is more obvious on the CPU time.

Table 1 Numerical results of examples 1-4 for Z-eigenvalues

$Ex.$	m	n	SATR ($k/\lambda_{\max}$/CPU)	FTR ($k/\lambda_{\max}$/CPU)
1	4	$n = 3$	5/0.8893/0.0100	8/0.8893/0.0313
2	4	$n = 10$	8/43.3/0.1875	11/43.3/0.2500
3	4	$n = 5$	8/13.1/0.0781	11/13.1/0.0938
3	4	$n = 10$	8/49.5/0.1563	11/49.5/0.2031
4	3	$n = 10$	7/17.8/0.0781	10/17.8/0.0781
4	3	$n = 20$	7/34.2/0.0250	10/34.2/0.0625
4	3	$n = 30$	7/50.1/0.1875	10/50.1/0.0938
4	3	$n = 40$	7/65.9/0.2581	10/65.9/0.3125
4	3	$n = 50$	7/81.6/0.4212	10/81.6/0.5772
4	3	$n = 60$	7/97.2/0.5148	10/97.2/0.7800
4	3	$n = 70$	7/113.2/0.4836	10/113.2/0.7956
4	3	$n = 80$	7/128/0.6921	10/128/0.9843
4	3	$n = 100$	7/158.2/1.3750	10/158.2/1.7344
4	3	$n = 200$	7/311.3/5.1875	10/311.3/7.4375
4	3	$n = 500$	7/765.4/77.7188	10/765.4/110.8438

In Table 2, we list the numerical results of largest $\mathcal{B}$-eigenvalue by SATR method. Here $\mathcal{B}$ is chosen as a special lth-order n-dimensional tensor and $l = m$

$$\mathcal{B} = b_{i_1 i_2 \cdots i_l} = \begin{cases} 2, & \text{if } i_1 = i_2 = \cdots = i_l, \\ 0, & \text{otherwise.} \end{cases}$$

Since m is even when we compute $\mathcal{B}$-eigenvalues in (3) , we only give the numerical results of Examples 1–3.

From Table 2, we can see that SATR method is efficient for computing $\mathcal{B}$-eigenvalues of (3). Moreover, as the dimension of problem grows larger, the CPU time cost of algorithm becomes increase, see the results of Example 2 and Example 3.

Table 2 Numerical results of Examples 1–3 for $\mathcal{B}$-eigenvalues

$Ex.$	n	k	$\lambda'_{\max}$	CPU
1	$n = 3$	5	1.3	0.0468
2	$n = 10$	7	9.0	0.0312
2	$n = 20$	7	51.9	0.1248
2	$n = 30$	7	131.3	1.1700
3	$n = 5$	6	14.7	0.0625
3	$n = 10$	6	46.8	0.0720

5 Conclusions

In this paper, we consider a $\mathcal{B}$-eigenvalues problem of symmetric tensors. And we propose the SATR method for solving it. Global convergence of the SATR method is established . Some numerical experiments illustrate that the SATR method is faster than FTR method when the problem (3) is reduce to (2). Other numerical results show that specific to $\mathcal{B}$-eigenvalues of symmetric tensors, the SATR method is effective.

Funding information This work is supported by the National Natural Science Foundation of China (11171131 and 11171003). Innovation Talent Training Program of Science and Technology of Jilin Province of China (20180519011JH).

Compliance with ethical standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Cui, C., Dai, Y., Nie, J.: All real eigenvalues of symmetric tensors. SIAM J. Matrix Anal. Appl. **4**, 1582–1601 (2014)
2. Chang, K., Pearson, K., Zhang, T.: On eigenvalue problems of real symmetric tensors. J. Math. Anal. Appl. **350**, 416–422 (2009)
3. Chang, K., Pearson, K., Zhang, T.: Primitivity, the convergence of the NQZ method, and the largest eigenvalue for nonnegative tensors. SIAM J. Matrix Anal. Appl. **3**, 806–819 (2011)
4. Chang, K., Zhang, T.: On the uniqueness and nonuniqueness of the Z-eigenvector for transition probability tensors. J. Math. Anal. Appl. **408**, 525–540 (2013)
5. Chu, M., Wu, S.: On the second dominant eigenvalue affecting the Power method for transition probability tensors. Manuscript (2014)
6. Conn, A., Gould, N., Toint, P.: Trust Region Methods MPS/SIAM Series on Optimization, vol. 1. SIAM, Philadelphia (2000)
7. Diamond, R.: A note on the rotational superposition problem. Acta Crystallographica. **A44**, 211–216 (1988)
8. Friedland, S., Gaubert, S., Han, L.: Perron-Frobenius theorem for nonnegative mulitilinear forms and extensions. Linear Algebra Appl. **2**, 738–749 (2013)
9. Hao, C., Cui, C., Dai, Y.: A sequential subspace projection method for extreme Z-eigenvalues of supersymmetric tensors. Numer. Linear Algebra Appl. **22**, 283–298 (2015)
10. Hao, C., Cui, C., Dai, Y.: A feasible trust-region method for calculating extreme Z-eigenvalues of symmetric tensors. Pac. J. Optim. **11**, 291–307 (2015)
11. Hei, L.: A self-adaptive trust region algorithm. J. Comput. Math. **21**, 229–236 (2003)
12. Hu, S., Huang, Z., Qi, L.: Strictly nonnegative tensors and nonnegative tensor partition. Sci. China Math. **1**, 181–195 (2014)
13. Kolda, T., Mayo, J.: Shifted power method for computing tensor eigenpairs. SIAM J. Matrix Anal. Appl. **32**, 1095–1124 (2011)
14. Kolda, T., Mayo, J.: An adaptive shifted power methods for computing generalized tensor eigenpairs. SIAM J. Matrix Anal. Appl. **35**, 1563–1581 (2014)
15. Kofidis, E., Regalia, P.: On the best rank-1 approximation of higher-order supersymmetric tensors. SIAM J. Matrix Anal. Appl. **23**, 863–884 (2002)
16. Lim, L.: Singular values and eigenvalues of tensors: a variational approach. In: 2005 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, vol. 1, pp. 129–132 (2005)
17. Luo, Z., Qi, L., Xiu, N.: The sparsest solutions to z-tensor complementarity problems. Optim Lett. (2016). https://doi.org/10.1007/s11590-016-1013-9

18. Li, G., Qi, L., Yu, G.: Semismoothness of the maximum eigenvalue function of a symmetric tensor and its application. Linear Algebra Appl. **2**, 813–833 (2013)
19. Li, G., Qi, L., Yu, G.: The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory. Numer. Linear Algebra Appl. **20**, 1001–1029 (2013)
20. Lu, Y., Li, W., Cao, M., Yang, Y.: A novel self-adaptive trust region algorithm for unconstrained optimization. J. Appl Math. **1**, 1–8 (2014)
21. Ng, M., Qi, L., Zhou, G.: Finding the largest eigenvalue of a nonnegative tensor. SIAM J. Matrix Anal. Appl. **3**, 1090–1099 (2009)
22. Ni, G., Qi, L., Wang, F., Wang, Y.: The degree of the E-characteristic polynomial of an even order tensor. J. Math Anal. Appl. **329**, 1218–1229 (2007)
23. Nie, J., Wang, L.: Semidefinite relaxations for best rank-1 tensor approximations. SIAM J. Matrix Anal. Appl. **35**, 1155–1179 (2013)
24. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
25. Qi, L.: Eigenvalues of a real supersymmetric tensor. J. Symb. Comput. **40**, 1302–1324 (2005)
26. Qi, L.: Rank and eigenvalues of a supersymmetric tensor, the multivariate homogeneous polynomial and the algebraic hypersurface it defines. J. Symb. Comput. **41**, 1309–1327 (2006)
27. Qi, L.: Eigenvalues invariants of tensors. J. Math. Anal. Appl. **2**, 1363–1377 (2007)
28. Qi, L., Wang, F., Wang, Y.: Z-eigenvalue methods for a global polynomial optimization problem. Math. Program. **118**, 301–316 (2009)
29. Qi, L., Wang, Y., Wu, E.: D-eigenvalues of diffusion kurtosis tensors. J. Comput. Appl Math. **221**, 150–157 (2008)
30. Shi, Z., Wang, S.: Nonmonotone adaptive trust region method. European J. Oper. Res. **208**, 28–36 (2011)
31. Wei, T., Goldbart, P.: Geometric measure of entanglement and applications to bipartite and multipartite quantum states. Phys. Rev. **68**, 042307 (2003)
32. Walmag, J., Delhez, E.: A note on trust-region radius update. SIAM J. Optim. **16**, 548–562 (2005)
33. Yang, W., Yang, Y., Zhang, C., Cao, M.: A Newton-like trust region method for large-scale unconstrained nonconvex minimization. Abst. Appl. Anal. **2013**, 1–6 (2013)
34. Yang, Y.: On the truncated conjugate gradient method. Report, ICM99003, ICMSEC, Chinese Academy of Sciences. Beijing, China (1999)
35. Zhang, L., Qi, L.: Linear convergenc of an algorithm for computing the largest eigenvalue of a nonnegative tensor. Numer. Linear Algebra Appl. **5**, 830–841 (2012)