

A TRUST REGION ALGORITHM FOR COMPUTING EXTREME EIGENVALUES OF TENSORS

YANNAN CHEN

School of Mathematical Sciences, South China Normal University
Guangzhou 510631, China

JINGYA CHANG*

School of Applied Mathematics, Guangdong University of Technology
Guangzhou 510520, China

ABSTRACT. Eigenvalues and eigenvectors of high order tensors have crucial applications in sciences and engineering. For computing H-eigenvalues and Z-eigenvalues of even order tensors, we transform the tensor eigenvalue problem to a nonlinear optimization with a spherical constraint. Then, a trust region algorithm for the spherically constrained optimization is proposed in this paper. At each iteration, an unconstrained quadratic model function is solved inexactly to produce a trial step. The Cayley transform maps the trial step onto the unit sphere. If the trial step generates a satisfactory actual decrease of the objective function, we accept the trial step as a new iterate. Otherwise, a second order line search process is performed to exploit valuable information contained in the trial step. Global convergence of the proposed trust region algorithm is analyzed. Preliminary numerical experiments illustrate that the novel trust region algorithm is efficient and promising.

1. Introduction. Spectral tensor theory has extensive applications in various areas of sciences and engineering. For example, H-eigenvalues of the adjacency tensor of a uniform hypergraph are valuable in spectral hypergraph theory [7]. The limiting probability distribution vector of a higher order Markov chain is a Z-eigenvector associated with Z-eigenvalue being one of a transition probability tensor [14]. In magnetic resonance imaging, Z-eigenvectors corresponding to major Z-eigenvalues of the higher order diffusion tensor describe orientations of crossing nerve fibers in white matter of cerebrum [3]. Z-eigenvalues of the third order octupolar tensor are employed to study tetrahedratric nematic phases in liquid crystals [9, 4]. The Fiedler vector of an even-uniform hypergraph is the Z-eigenvector associated with the second smallest Z-eigenvalue of the hypergraph Laplacian tensor [6].

2020 *Mathematics Subject Classification.* Primary: 15A18, 15A69; Secondary: 65K05, 90C30.

Key words and phrases. Tensor eigenvalue, spherical optimization, trust region, second order line search, Cayley transform.

The first author is supported by the National Natural Science Foundation of China grant 11771405 and Guangdong Basic and Applied Basic Research Foundation 2020A1515010489. The second author is supported by the National Natural Science Foundation of China grant 11901118 and Guangdong Basic and Applied Basic Research Foundation 2020B1515310001.

* Corresponding author: Jingya Chang.

As a branch of numerical [multilinear algebra](#) in computational mathematics [17], tensor eigenvalue computations attract wide attentions from many researchers. Kolda and Mayo [12, 13] proposed a shifted power method for computing eigenvalues of symmetric tensors. Unconstrained optimization approaches were studied in [11]. Cui et al. [8] designed a polynomial optimization algorithm for evaluating all real eigenvalues of a tensor. Homotopy methods for eigen-system [2] could compute all eigenvalues of a tensor. For the case of large scale tensors, Chen et al. [5] and Chang et al. [1] proposed Barzilai–Borwein gradient algorithm and a limited memory BFGS quasi-Newton algorithm for Hankel tensors and sparse tensors, respectively.

In this paper, we study a trust region method for computing extreme eigenvalues of even order symmetric tensors. This kind of tensor eigenvalue problem is equivalent to a spherically constrained optimization. Owing to the geometry of the spherical constraint, the Cayley transform is employed to preserve iterates on the unit sphere. We construct and solve a unconstrained quadratic model function in each iteration for a trial step. If the trial step produces a satisfactory actual decrease in the objective function, we accept it as a new iterate and may enlarge the trust region radius. Otherwise, motivated by the second order linear search approach, a novel backtracking curvilinear search on the unit sphere is performed to find a proper step size. The trust region radius is reduced if the curvilinear search is triggered off. Next, we prove that the proposed combining trust region and line search algorithm for spherically constrained optimization converges from an arbitrary initial point. Finally, numerical experiments on small symmetric tensors and a large scale Hilbert tensor illustrate that the new trust region algorithm is efficient and competitive when compared with some existing approaches.

The outline of this paper is drawn as follows. Preliminary on higher order tensors and tensor eigenvalues is presented in Section 2. In Section 3, we design a novel trust region algorithm for computing eigenvalues and eigenvectors of an even-order symmetric tensor. The global convergence of the proposed trust region method is analyzed in Section 4. Numerical experiments on small and large scale tensors are illustrated in Section 5. Some concluding remarks are made in Section 6.

2. Preliminary. Let $\mathbb{R}^{[m,n]}$ be the space of m th order n dimensional real tensors, where m and n are positive integers. A tensor $\mathcal{A} \in \mathbb{R}^{[m,n]}$ has n^m entries:

$$a_{i_1 i_2 \dots i_m} \quad \text{for all } i_j \in \{1, 2, \dots, n\} \text{ and } j \in \{1, 2, \dots, m\}.$$

If each entry $a_{i_1 i_2 \dots i_m}$ of a tensor \mathcal{A} is invariable under any permutation of its indices, we call \mathcal{A} a symmetric tensor.

For a vector $\mathbf{x} \in \mathbb{R}^n$, products of $\mathcal{A} \in \mathbb{R}^{[m,n]}$ and \mathbf{x} can produce a scalar $\mathcal{A}\mathbf{x}^m \in \mathbb{R}$, a vector $\mathcal{A}\mathbf{x}^{m-1} \in \mathbb{R}^n$, and a matrix $\mathcal{A}\mathbf{x}^{m-2} \in \mathbb{R}^{n \times n}$:

$$\begin{aligned} \mathcal{A}\mathbf{x}^m &= \sum_{i_1, i_2, i_3, \dots, i_m=1}^n a_{i_1 i_2 i_3 \dots i_m} x_{i_1} x_{i_2} x_{i_3} \cdots x_{i_m}, \\ (\mathcal{A}\mathbf{x}^{m-1})_p &= \sum_{i_2, i_3, \dots, i_m=1}^n a_{p i_2 i_3 \dots i_m} x_{i_2} x_{i_3} \cdots x_{i_m}, \quad \forall p \in \{1, 2, \dots, n\}, \\ (\mathcal{A}\mathbf{x}^{m-2})_{pq} &= \sum_{i_3, \dots, i_m=1}^n a_{p q i_3 \dots i_m} x_{i_3} \cdots x_{i_m}, \quad \forall p, q \in \{1, 2, \dots, n\}. \end{aligned}$$

Let $\mathcal{I} \in \mathbb{R}^{[m,n]}$ be an identity tensor with diagonal entries $[\mathcal{I}]_{ii\dots i} = 1$ for all $i \in \{1, 2, \dots, n\}$ and off-diagonal entries being zeros. Suppose m is even. Then we have $\mathcal{I}\mathbf{x}^m = \|\mathbf{x}\|_m^m$, $\mathcal{I}\mathbf{x}^{m-1} = (x_1^{m-1}, x_2^{m-1}, \dots, x_n^{m-1}) =: \mathbf{x}^{[m-1]}$, and $\mathcal{I}\mathbf{x}^{m-2} = \text{diag}(\mathbf{x}^{[m-2]})$ is a diagonal matrix, where $\|\cdot\|_m$ is the ℓ_m -norm. There is another simple and symmetric tensor $\mathcal{E} \in \mathbb{R}^{[m,n]}$ such that $\mathcal{E}\mathbf{x}^m = \|\mathbf{x}\|_m^m$, $\mathcal{E}\mathbf{x}^{m-1} = \|\mathbf{x}\|_m^{m-2}\mathbf{x}$, and $\mathcal{E}\mathbf{x}^{m-2} = \frac{1}{m-1}\|\mathbf{x}\|_m^{m-4}(\|\mathbf{x}\|^2 I + (m-2)\mathbf{x}\mathbf{x}^T)$.

In 2005, Qi [16] and Lim [15] defined eigenvalues of tensors independently. Let $\mathcal{A} \in \mathbb{R}^{[m,n]}$. If there exist a $\lambda \in \mathbb{R}$ and a nonzero $\mathbf{x} \in \mathbb{R}^n$ such that

$$\mathcal{A}\mathbf{x}^{m-1} = \lambda\mathbf{x}^{[m-1]}, \quad (1)$$

then λ is called an H-eigenvalue of \mathcal{A} and \mathbf{x} is the associated H-eigenvector. If $(\lambda, \mathbf{x}) \in \mathbb{R} \oplus \mathbb{R}^n$ satisfying the following system

$$\mathcal{A}\mathbf{x}^{m-1} = \lambda\mathbf{x} \quad \text{and} \quad \mathbf{x}^T \mathbf{x} = 1, \quad (2)$$

then λ is called a Z-eigenvalue of \mathcal{A} and \mathbf{x} is the associated Z-eigenvector. We refer the interested readers to the monograph [18].

3. A trust region method. In this section, we are going to design a trust region method for computing H-eigenvalues and Z-eigenvalues of tensors. Suppose that m is even, $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{[m,n]}$ are symmetric, and \mathcal{B} is a positive definite tensor (i.e., $\mathcal{B}\mathbf{x}^m > 0$ for all $\mathbf{x} \neq 0$).¹ We consider a spherically constrained optimization [1, 5]:

$$\min f(\mathbf{x}) := \frac{\mathcal{A}\mathbf{x}^m}{\mathcal{B}\mathbf{x}^m} \quad \text{s.t. } \mathbf{x} \in \mathbb{S}^{n-1}, \quad (3)$$

where $\mathbb{S}^{n-1} := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T \mathbf{x} = 1\}$ is a unit spherical surface. By Calculus, the gradient of f is

$$\nabla f(\mathbf{x}) = \frac{m}{\mathcal{B}\mathbf{x}^m} \left(\mathcal{A}\mathbf{x}^{m-1} - \frac{\mathcal{A}\mathbf{x}^m}{\mathcal{B}\mathbf{x}^m} \mathcal{B}\mathbf{x}^{m-1} \right). \quad (4)$$

Clearly, it holds that $\mathbf{x}^T \nabla f(\mathbf{x}) = 0$. In the case of $\nabla f(\mathbf{x}) = 0$, if $\mathcal{B} = \mathcal{I}$, it is easy to see that $f(\mathbf{x}) = \mathcal{A}\mathbf{x}^m / \|\mathbf{x}\|_m^m$ is an H-eigenvalue of \mathcal{A} . If $\mathcal{B} = \mathcal{E}$, $\nabla f(\mathbf{x}) = 0$ implies $f(\mathbf{x}) = \mathcal{A}\mathbf{x}^m / \|\mathbf{x}\|_m^m$ is a Z-eigenvalue of \mathcal{A} . According to the minimization in (3), the minimum value of (3) is an extreme H- or Z-eigenvalue of \mathcal{A} . Therefore, the proposed trust region method is devoted to solve the spherically constrained optimization (3), where the tensor \mathcal{B} could be either \mathcal{I} for extreme H-eigenvalues of \mathcal{A} or \mathcal{E} for extreme Z-eigenvalues of \mathcal{A} .

Trust region method is an iterative algorithm [20, 21, 22]. At an iterate \mathbf{x}_k , we construct a trust region subproblem

$$\begin{aligned} \min m_k(\mathbf{d}) &:= f_k + \mathbf{g}_k^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T H_k \mathbf{d} \\ \text{s.t. } \|\mathbf{d}\| &\leq \Delta_k, \end{aligned} \quad (5)$$

where Δ_k is a trust region radius, m_k is a quadratic model function, $f_k := f(\mathbf{x}_k)$, $\mathbf{g}_k := \nabla f(\mathbf{x}_k)$, $H_k := P_k \nabla^2 f(\mathbf{x}_k) P_k$ is the projected Hessian of f , $P_k := (I - \mathbf{x}_k \mathbf{x}_k^T)$ is the projection matrix onto the tangent hyperplane of \mathbb{S}^{n-1} at \mathbf{x}_k ,

$$\begin{aligned} \nabla^2 f(\mathbf{x}_k) &= \frac{m(m-1)}{\mathcal{B}\mathbf{x}^m} \mathcal{A}\mathbf{x}^{m-2} - \frac{m^2}{(\mathcal{B}\mathbf{x}^m)^2} (\mathcal{A}\mathbf{x}^{m-1} \odot \mathcal{B}\mathbf{x}^{m-1}) \\ &\quad - \frac{m(m-1)\mathcal{A}\mathbf{x}^m}{(\mathcal{B}\mathbf{x}^m)^2} \mathcal{B}\mathbf{x}^{m-2} + \frac{m^2\mathcal{A}\mathbf{x}^m}{(\mathcal{B}\mathbf{x}^m)^3} (\mathcal{B}\mathbf{x}^{m-1} \odot \mathcal{B}\mathbf{x}^{m-1}), \end{aligned} \quad (6)$$

¹If m is odd, the optimization model (3) is not applicable because its denominator may be positive, negative, or zero.

and $\mathbf{u} \odot \mathbf{v} = \mathbf{u}\mathbf{v}^T + \mathbf{v}\mathbf{u}^T$. The trust region subproblem is solved inexactly for a trial step \mathbf{d}_k satisfying

$$m_k(0) - m_k(\mathbf{d}_k) \geq \tau \|\mathbf{g}_k\| \min \left\{ \Delta_k, \frac{\|\mathbf{g}_k\|}{\|H_k\|} \right\} \quad (7)$$

and

$$\mathbf{g}_k^T \mathbf{d}_k \leq 0, \quad (8)$$

where $\tau \in (0, 1/2]$ is a constant. We note that (8) is trivial, because we may prefer to use $-\mathbf{d}_k$ if there exists a \mathbf{d}_k satisfying (7) and $\mathbf{g}_k^T \mathbf{d}_k > 0$.

To preserve iterates on the sphere \mathbb{S}^{n-1} , the Cayley transform is serviceable [10]. From a skew-symmetric matrix

$$W_k(\alpha) = \frac{\alpha}{2} (\mathbf{x}_k \mathbf{d}_k^T - \mathbf{d}_k \mathbf{x}_k^T),$$

the Cayley transform produces an orthogonal matrix

$$Q_k(\alpha) = (I + W_k(\alpha))^{-1} (I - W_k(\alpha)),$$

where $\alpha > 0$ is a step size and $I + W_k(\alpha)$ is nonsingular. Hence, once $\mathbf{x}_k \in \mathbb{S}^{n-1}$, we also have $\mathbf{x}_k^+(\alpha) := Q_k(\alpha) \mathbf{x}_k \in \mathbb{S}^{n-1}$. Indeed, matrices $W_k(\alpha)$ and $Q_k(\alpha)$ are not required to form explicitly. It holds that [1, 6]

$$\mathbf{x}_k^+(\alpha) = \frac{[(2 - \alpha \mathbf{d}_k^T \mathbf{x}_k)^2 - \alpha^2 \|\mathbf{d}_k\|^2] \mathbf{x}_k + 4\alpha \mathbf{d}_k}{4 + \alpha^2 \|\mathbf{d}_k\|^2 - \alpha^2 (\mathbf{d}_k^T \mathbf{x}_k)^2}. \quad (9)$$

Next, we perform a second order curvilinear search process to determine the step size. Let

$$q_k(\mathbf{d}) := f_k + \mathbf{g}_k^T \mathbf{d} + \frac{1}{2} \min\{0, \mathbf{d}^T H_k \mathbf{d}\} \quad (10)$$

be a surrogate (lower bound) of the model function m_k . We take $\alpha \leftarrow 1$ initially. If the ratio between the actual decrease in f and the predicted decrease in q_k is sufficiently large, i.e.,

$$\rho_k := \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k^+(\alpha))}{q_k(0) - q_k(\alpha \mathbf{d}_k)} \geq \eta_1, \quad (11)$$

where $\eta_1 \in (0, 1)$ is a constant, we accept the step size as α_k , take $\mathbf{x}_{k+1} = \mathbf{x}_k^+(\alpha)$, and may enlarge the trust region radius. Otherwise, we decrease the step size α and try again. Since q_k is used, this process is indeed a second order linear search [21]. We reduce the trust region radius when a backtracking linear search is triggered off. Next, we explain that a step size α could be found in finitely many steps. By calculations, we have

$$\left. \frac{d}{d\alpha} f(\mathbf{x}_k^+(\alpha)) \right|_{\alpha=0} = \mathbf{g}_k^T \mathbf{d}_k \quad \text{and} \quad \left. \frac{d^2}{d\alpha^2} f(\mathbf{x}_k^+(\alpha)) \right|_{\alpha=0} = \mathbf{d}_k^T H_k \mathbf{d}_k, \quad (12)$$

Because \mathbf{d}_k satisfies (7) and (8), there are two cases: (i) $\mathbf{g}_k^T \mathbf{d}_k < 0$ and (ii) $\mathbf{g}_k^T \mathbf{d}_k = 0$ and $\mathbf{d}_k^T H_k \mathbf{d}_k < 0$. In both cases, a sufficiently small α will satisfy (11) since $\eta_1 \in (0, 1)$.

In summary, the trust region method for computing an extreme eigenvalue and its associated eigenvector of an even-order symmetric tensor is presented in Algorithm 1. Note that we can set $f(\mathbf{x}) := -\frac{\mathcal{A}\mathbf{x}^m}{B\mathbf{x}^m}$ in the context of computing the maximal eigenvalue of a tensor.

Algorithm 1 A trust region algorithm for computing an eigenvalue of a tensor.

- 1: Set $\mathcal{B} = \mathcal{I}$ and $\mathcal{B} = \mathcal{E}$ if an H-eigenvalue and a Z-eigenvalue are in purpose, respectively.
- 2: Set parameters $0 < \eta_1 < \eta_2 < 1/2$, $0 < \gamma_1 < \gamma_2 < 1 < \gamma_3$, and $0 < \Delta_0 \leq \bar{\Delta}$. Choose an initial point $\mathbf{x}_0 \in \mathbb{S}^{n-1}$ and set $k \leftarrow 0$.
- 3: **while** $\nabla f(\mathbf{x}_k) \neq 0$ **do**
- 4: Calculate $\mathcal{A}\mathbf{x}^m, \mathcal{B}\mathbf{x}^m, \mathcal{A}\mathbf{x}^{m-1}, \mathcal{B}\mathbf{x}^{m-1}, \mathcal{A}\mathbf{x}^{m-2}$, and $\mathcal{B}\mathbf{x}^{m-2}$.
- 5: Solve the trust region subproblem:

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{d}^T H_k \mathbf{d} + \mathbf{g}_k^T \mathbf{d} + f_k \\ \text{s.t.} \quad & \|\mathbf{d}\| \leq \Delta_k, \end{aligned}$$

inexactly for a trial step \mathbf{d}_k satisfying (7) and (8).

- 6: Backtracking search on \mathbb{S}^{n-1} . Find a smallest nonnegative integer j such that the step size $\alpha = \gamma_2^j$ satisfies:

$$\rho_k = \frac{f_k - f(\mathbf{x}_k^+(\alpha))}{q_k(0) - q_k(\alpha \mathbf{d}_k)} \geq \eta_1,$$

where $\mathbf{x}_k^+(\alpha)$ is defined by (9).

- 7: Update an iterate. Set $\alpha_k = \gamma_2^j$ and $\mathbf{x}_{k+1} = \mathbf{x}_k^+(\alpha_k)$.
- 8: Update a trust region radius. If $\alpha_k = 1$, we choose

$$\Delta_{k+1} \in \begin{cases} [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\Delta_k, \min\{\gamma_3 \Delta_k, \bar{\Delta}\}] & \text{if } \rho_k \geq \eta_2, \end{cases}$$

else

$$\Delta_{k+1} \in [\max\{\gamma_1 \Delta_k, \alpha_k \|\mathbf{d}_k\|\}, \gamma_2 \Delta_k].$$

- 9: Set $k \leftarrow k + 1$.

10: **end while**

4. Convergence analysis. We are going to prove that, by starting from any initial point on the unit sphere, the proposed trust region method produces a subsequence of iterates that converges to a eigen-pair of the tensor. Because the feasible region \mathbb{S}^{n-1} is compact and the tensor \mathcal{B} is positive definite, the objective function f in (3) is twice continuously differentiable. Hence, the following lemma holds.

Lemma 4.1. *For all $\mathbf{x}_k \in \mathbb{S}^{n-1}$, there exists a constant $M \geq 1$ such that*

$$|f_k| \leq M, \quad \|\mathbf{g}_k\| \leq M, \quad \text{and} \quad \|H_k\| \leq M.$$

We consider the case that Algorithm 1 generates an infinity sequence of iterations. To prove the convergence of Algorithm 1, we prove by contradiction. Suppose there exists a positive constant κ_g such that

$$\|\mathbf{g}_k\| \geq \kappa_g. \tag{13}$$

The following lemma shows that the trust region radius is bounded away from zero.

Lemma 4.2. *Under the assumption of (13), there exists a positive constant Δ_{\min} satisfying*

$$\Delta_k \geq \Delta_{\min}.$$

Proof. Following the proof of [21, Lemma 6.1.7], we assume that a trust region radius satisfies

$$\Delta_k \leq \frac{4\tau\kappa_g(1-\eta_2)}{(4+\bar{\Delta})M} := c_1.$$

Then, it yields $\rho_k \geq \eta_2$ and hence the trust region can not be shrunk. Since $q_k(\mathbf{d}_k) \leq m_k(\mathbf{d}_k) \leq m_k(0) = q_k(0)$ and (7), we have

$$|q_k(0) - q_k(\mathbf{d}_k)| \geq |m_k(0) - m_k(\mathbf{d}_k)| \geq \tau\Delta_k\kappa_g.$$

It yields from Taylor's Theorem that

$$f(\mathbf{x}_k^+(1)) \leq f_k + \mathbf{g}_k^T(\mathbf{x}_k^+(1) - \mathbf{x}_k) + \frac{M}{2}\|\mathbf{x}_k^+(1) - \mathbf{x}_k\|^2.$$

By $\mathbf{x}_k^+(1), \mathbf{x}_k \in \mathbb{S}^{n-1}$ and (9), we have

$$\|\mathbf{x}_k^+(1) - \mathbf{x}_k\|^2 = 2 - 2(\mathbf{x}_k^+(1))^T \mathbf{x}_k = \frac{4(\|\mathbf{d}_k\|^2 - (\mathbf{d}_k^T \mathbf{x}_k)^2)}{4 + \|\mathbf{d}_k\|^2 - (\mathbf{d}_k^T \mathbf{x}_k)^2} \leq \|\mathbf{d}_k\|^2.$$

By $\mathbf{g}_k^T \mathbf{x}_k = 0$ and (9), we get

$$|\mathbf{g}_k^T(\mathbf{x}_k^+(1) - \mathbf{x}_k - \mathbf{d}_k)| = |\mathbf{g}_k^T \mathbf{d}_k| \frac{\|\mathbf{d}_k\|^2 - (\mathbf{d}_k^T \mathbf{x}_k)^2}{4 + \|\mathbf{d}_k\|^2 - (\mathbf{d}_k^T \mathbf{x}_k)^2} \leq \frac{M\bar{\Delta}}{4}\|\mathbf{d}_k\|^2.$$

Combining the above inequalities, we obtain

$$\begin{aligned} |\rho_k - 1| &= \frac{|f_k - f(\mathbf{x}_k^+(1)) - q_k(0) + q_k(\mathbf{d}_k)|}{|q_k(0) - q_k(\mathbf{d}_k)|} \\ &\leq \frac{|\mathbf{g}_k^T(\mathbf{x}_k^+(1) - \mathbf{x}_k - \mathbf{d}_k)| + \frac{M}{2}\|\mathbf{x}_k^+(1) - \mathbf{x}_k\|^2 + \frac{M}{2}\|\mathbf{d}_k\|^2}{|q_k(0) - q_k(\mathbf{d}_k)|} \\ &\leq \frac{(M\bar{\Delta}/4)\|\mathbf{d}_k\|^2 + M\|\mathbf{d}_k\|^2}{\tau\Delta_k\kappa_g} \\ &\leq \frac{(4+\bar{\Delta})M}{4\tau\kappa_g}\Delta_k \leq (1-\eta_2), \end{aligned}$$

which implies $\rho_k \geq \eta_2$. By the rule of updating Δ_k in Algorithm 1, we say $\Delta_{k+1} \geq \Delta_k$. For every Δ_k satisfying $\Delta_k \leq c_1$ and $\Delta_{k-1} > c_1$, it holds that $\Delta_k \geq c_1\gamma_1$. Hence, by choosing $\Delta_{\min} = c_1\gamma_1$, we prove this lemma. \square

Next, we prove that the step size is also bounded away from zero.

Lemma 4.3. Suppose (13) holds. We have

$$\liminf_{k \rightarrow \infty} \alpha_k > 0.$$

Proof. We process by contradiction. Suppose that there is a subsequence of iterations such that $\alpha_{k_i} \rightarrow 0$. By the backtracking search rule, we know

$$\frac{f_{k_i} - f(\mathbf{x}_{k_i}^+(\gamma_2^{-1}\alpha_{k_i}))}{q_{k_i}(0) - q_{k_i}(\gamma_2^{-1}\alpha_{k_i}\mathbf{d}_{k_i})} < \eta_1.$$

From Taylor's Theorem and (12), the above inequality yields

$$(1 - \eta_1)\mathbf{g}_{k_i}^T \mathbf{d}_{k_i} + \frac{\alpha_{k_i}}{2\gamma_2}(\mathbf{d}_{k_i}^T H_{k_i} \mathbf{d}_{k_i} - \eta_1 \min\{0, \mathbf{d}_{k_i}^T H_{k_i} \mathbf{d}_{k_i}\}) + o\left(\frac{\alpha_{k_i}^2}{\gamma_2^2}\right) > 0. \quad (14)$$

By (8), we also know

$$\mathbf{d}_{k_i}^T H_{k_i} \mathbf{d}_{k_i} - \eta_1 \min\{0, \mathbf{d}_{k_i}^T H_{k_i} \mathbf{d}_{k_i}\} + o\left(\frac{\alpha_{k_i}}{\gamma_2}\right) > 0. \quad (15)$$

When $i \rightarrow \infty$, a subsequence of $\{\mathbf{x}_{k_i}\}$ converges to a limit point \mathbf{x}_∞ . Then, $\mathbf{g}_{k_i} \rightarrow \mathbf{g}_\infty$, $H_{k_i} \rightarrow H_\infty$, and $\mathbf{d}_{k_i} \rightarrow \mathbf{d}_\infty$. Here, we slightly abuse of notations. Now, we consider the following two cases.

(1) Assume $\mathbf{g}_\infty^T \mathbf{d}_\infty < 0$. By taking $i \rightarrow \infty$ on (14), we obtain $\mathbf{g}_\infty^T \mathbf{d}_\infty \geq 0$, which generates a contradiction.

(2) Then $\mathbf{g}_\infty^T \mathbf{d}_\infty = 0$. For sufficiently large i , by (7) and Lemma 4.2, we know

$$\mathbf{d}_{k_i}^T H_{k_i} \mathbf{d}_{k_i} \leq -\tau \kappa_g \Delta_{\min}/2 < 0. \quad (16)$$

By taking $i \rightarrow \infty$ on (15), we obtain $(1 - \eta_1) \mathbf{d}_\infty^T H_\infty \mathbf{d}_\infty \geq 0$, which contradicts (16).

Therefore, we can not make the assumption $\alpha_{k_i} \rightarrow 0$. The proof is complete. \square

Finally, we get the following theorem.

Theorem 4.4. *Suppose that Algorithm 1 generates an infinity sequence of iterations. Then, we have*

$$\liminf_{k \rightarrow \infty} \|\mathbf{g}_k\| = 0.$$

There exists a subsequence of iterates $\{\mathbf{x}_{k_j}\}$ that converges to a stationary point \mathbf{x}_ . Hence, $f(\mathbf{x}_*)$ is an eigenvalue of \mathcal{A} and \mathbf{x}_* is the associated eigenvector.*

Proof. Because $q_k(\alpha_k \mathbf{d}_k) \leq m_k(\alpha_k \mathbf{d}_k) < m_k(0) = q_k(0)$, $\alpha_k \in (0, 1]$, and (8), we know

$$q_k(0) - q_k(\alpha_k \mathbf{d}_k) \geq m_k(0) - m_k(\alpha_k \mathbf{d}_k) \geq \alpha_k^2 (m_k(0) - m_k(\mathbf{d}_k)).$$

Therefore, by (11), (7), Lemmas 4.1 and 4.2, we obtain

$$\begin{aligned} f_0 + M &\geq \sum_{k=0}^{\infty} (f_k - f_{k+1}) \\ &\geq \eta_1 \sum_{k=0}^{\infty} (q_k(0) - q_k(\alpha_k \mathbf{d}_k)) \\ &\geq \eta_1 \sum_{k=0}^{\infty} \alpha_k^2 (m_k(0) - m_k(\mathbf{d}_k)) \\ &\geq \eta_1 \sum_{k=1}^{\infty} \alpha_k^2 \tau \kappa_g \min\{\Delta_{\min}, \kappa_g/M\}, \end{aligned}$$

which implies $\alpha_k \rightarrow 0$ as $k \rightarrow \infty$. This contradicts Lemma 4.3. Therefore, we can not make the assumption (13). This theorem is proved. \square

We note that [1, Theorem 4.9], when we choose an initial point \mathbf{x}_0 on the unit sphere \mathbb{S}^{n-1} uniformly, the optimization algorithm produces an extreme eigenvalue of \mathcal{A} with a positive probability, by using the semialgebraic property of (3). Moreover, if we run the optimization algorithm many times from different initial points sampled uniformly on \mathbb{S}^{n-1} and pick up the best solution, the probability of obtaining the extreme eigenvalue of \mathcal{A} tends to 1.

5. Numerical experiments. To evaluate performance of the proposed trust region algorithm (TRA), we compare it with three other algorithms.

- An adaptive power (PM) method [12, 13]. The codes of PM is available from the Tensor Toolbox.
- A Barzilai–Borwein gradient optimization algorithm (BBGA) [5].
- Limited memory BFGS quasi-Newton algorithm (QNA) [1].

The proposed trust region method is written in MATLAB with the following settings

$$\eta_1 = 0.01, \eta_2 = 0.25, \gamma_1 = 0.25, \gamma_2 = 0.5, \gamma_3 = 2, \text{ and } \bar{\Delta} = 10.$$

We solve the trust region subproblem (5) by a conjugate gradient algorithm, where a subroutine for computing the product of the Hessian matrix and vectors is provided.

Example 1. We consider a symmetric tensor $\mathcal{A} \in \mathbb{R}^{[4,3]}$ [12] with independent entries

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

The largest Z-eigenvalue of \mathcal{A} is $\lambda_{\max}^Z = 0.8893$ and the smallest Z-eigenvalue of \mathcal{A} is $\lambda_{\min}^Z = -1.0954$.

TABLE 1. Numerical results on Example 1.

Solvers	PM	BBGA	QNA	TRA
λ_{\max}^Z	0.8893	0.8893	0.8893	0.8893
#Iter'n	3699	1697	1142	450
CPU time	12.55	0.56	0.46	0.37
λ_{\min}^Z	-1.0953	-1.0953	-1.0953	-1.0953
#Iter'n	1725	1121	808	333
CPU time	5.65	0.23	0.30	0.24

We run four algorithms PM, BBGA, QNA, and TRA from 100 initial points sampled randomly on the unit sphere. By taking the best one, we report the largest and smallest Z-eigenvalues of \mathcal{A} estimated by four algorithms in rows 2 and 5 of Table 1 respectively. Obviously, they all find the true eigenvalues. The total number of iterations and the total CPU times (in seconds) of 100 runs are also illustrated in Table 1. Clearly, the optimization algorithms BBGA, QNA, and TRA are faster than PM. Particularly, since second order information from the Hessian of the objective function is exploited, the proposed TRA works better than other algorithms.

Example 2. Let $\mathcal{A}(\alpha) \in \mathbb{R}^{[4,3]}$ be a symmetric tensor with nonzero independent entries [8]

$$a_{1111} = 2, a_{2222} = 3, a_{3333} = 5, a_{1123} = \alpha/3.$$

The largest H-eigenvalue of $\mathcal{A}(1)$ and $\mathcal{A}(3)$ are 5.1812 and 7.4505 respectively. The smallest H-eigenvalue of $\mathcal{A}(1)$ and $\mathcal{A}(3)$ are 1.2268 and -1.3952 respectively.

By running PM, BBGA, QNA, and TRA algorithms from 100 randomly generated initial points, the smallest and largest H-eigenvalues of $\mathcal{A}(1)$ and $\mathcal{A}(3)$ are found and reported in Table 2. The total number of iterations and CPU times

TABLE 2. Numerical results on Example 2.

Solvers	PM	BBGA	QNA	TRA
λ_{\min}^H of $\mathcal{A}(1)$	1.2268	1.2268	1.2268	1.2268
#Iter'n	16713	1423	1159	482
CPU time	38.12	0.31	0.43	0.29
λ_{\max}^H of $\mathcal{A}(1)$	5.1812	5.1812	5.1812	5.1812
#Iter'n	23632	1336	1159	753
CPU time	53.06	0.30	0.43	0.37
λ_{\min}^H of $\mathcal{A}(3)$	-1.3952	-1.3952	-1.3952	-1.3952
#Iter'n	21214	1127	986	429
CPU time	48.92	0.29	0.40	0.28
λ_{\max}^H of $\mathcal{A}(3)$	7.4505	7.4505	7.4505	7.4505
#Iter'n	21711	1263	1152	711
CPU time	50.09	0.30	0.45	0.36

used for these algorithms are also reported. There algorithms: BBGA, QNA, and TRA performs significantly better than PM. Compared with BBGA and QNA, the proposed TRA approach saves about a half iterations.

Example 3. A Hilbert tensor $\mathcal{H} \in \mathbb{R}^{[m,n]}$ has entries [19, 5]

$$h_{i_1 i_2 \dots i_m} = \frac{1}{i_1 + i_2 + \dots + i_m - m + 1} \quad \forall i_1, i_2, \dots, i_m.$$

This is a special Hankel tensor with a generating vector $\mathbf{v} := (1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{m(n-1)+1})$. Hilbert tensor is positive definite when its order is even. The largest Z-eigenvalue of a Hilbert tensor is bounded by $n^{m/2} \sin \frac{\pi}{n}$.

There is a fast algorithm for computing products of a Hankel tensor and vectors. For a vector $\mathbf{x} \in \mathbb{R}^n$, we define

$$\tilde{\mathbf{x}} := \begin{pmatrix} \mathbf{x} \\ 0_{m(n-1)+1-n} \end{pmatrix}.$$

By the fast (inverse) Fourier transform, we obtain

$$\mathcal{H}\mathbf{x}^m = \text{ifft}(\mathbf{v})^T (\text{fft}(\tilde{\mathbf{x}})^{\circ m}),$$

where “ \circ ” stands for the Hadamard product and $\mathbf{y}^{\circ m} = \mathbf{y} \circ \mathbf{y} \circ \dots \circ \mathbf{y}$ of m vectors \mathbf{y} . Furthermore, $\mathcal{H}\mathbf{x}^{m-1}$ and $(\mathcal{H}\mathbf{x}^{m-2})\mathbf{d}$ are constituted by the leading n entries of vectors

$$\text{fft} \left(\text{ifft}(\mathbf{v}) \circ \left(\text{fft}(\tilde{\mathbf{x}})^{\circ(m-1)} \right) \right) \quad \text{and} \quad \text{fft} \left(\text{ifft}(\mathbf{v}) \circ \text{fft}(\tilde{\mathbf{d}}) \circ \left(\text{fft}(\tilde{\mathbf{x}})^{\circ(m-2)} \right) \right),$$

respectively.

We compute the largest Z-eigenvalue of Hilbert tensors with fourth and sixth orders and dimensions ranging from ten to one million. In each case, we run BBGA, QNA, and TRA from 10 different initial points and record the best Z-eigenvalue of a Hilbert tensor. Because these algorithms always find the same Z-eigenvalue, we only write once in Table 3. The computed Z-eigenvalues and total CPU times are reported in Table 3. As the increase of the dimension, the largest Z-eigenvalue of the Hilbert tensor increases too. Since the curvature of the objective function is used, the proposed TRA is faster than BBGA and QNA from the viewpoint of CPU times.

TABLE 3. Numerical results on Hilbert tensors.

Order	Dimension	λ_{\max}^Z	BBGA	QNA	TRA
4	10	6.5289	0.04	0.06	0.11
	100	6.0499×10^1	0.09	0.09	0.11
	1,000	6.0050×10^2	0.32	0.31	0.29
	10,000	6.0006×10^3	3.99	3.50	2.63
	100,000	6.0001×10^4	31.23	30.23	23.72
	1,000,000	6.0001×10^5	425.05	452.17	371.71
6	10	4.0427×10^1	0.14	0.29	0.09
	100	3.7308×10^3	0.14	0.13	0.13
	1,000	3.7023×10^5	0.73	0.58	0.55
	10,000	3.6994×10^7	7.36	6.84	7.12
	100,000	3.6991×10^9	113.75	112.62	75.49
	1,000,000	3.6991×10^{11}	3091.54	3186.61	1439.50

6. Conclusion. We employed the classical trust region method from nonlinear programming for computing extreme H-eigenvalues and Z-eigenvalues of tensors. Under the assumption that the tensor is of even order and symmetric, the tensor eigenvalue problem is equivalent to a smooth optimization constrained by a unit sphere. By exploit the nice geometry of the spherical constraint, the proposed trust region algorithm seems like an unconstrained one. Preliminary numerical experiments on small and large scale tensors illustrated that the proposed trust region algorithm is efficient and promising.

Acknowledgments. We thank two referees for their valuable comments.

REFERENCES

- [1] J. Chang, Y. Chen and L. Qi, [Computing eigenvalues of large scale sparse tensors arising from a hypergraph](#), *SIAM Journal on Scientific Computing*, **38** (2016), A3618–A3643.
- [2] L. Chen, L. Han and L. Zhou, [Computing tensor eigenvalues via homotopy methods](#), *SIAM Journal on Matrix Analysis and Applications*, **37** (2016), 290–319.
- [3] Y. Chen, Y. Dai, D. Han and W. Sun, [Positive semidefinite generalized diffusion tensor imaging via quadratic semidefinite programming](#), *SIAM Journal on Imaging Sciences*, **6** (2013), 1531–1552.
- [4] Y. Chen, L. Qi and E. G. Virga, [Octupolar tensors for liquid crystals](#), *Journal of Physics A: Mathematical and Theoretical*, **51** (2018), 025206.
- [5] Y. Chen, Liqun Qi and Qun Wang, [Computing extreme eigenvalues of large scale Hankel tensors](#), *Journal of Scientific Computing*, **68** (2016), 716–738.
- [6] Y. Chen, L. Qi and X. Zhang, [The Fiedler vector of a Laplacian tensor for hypergraph partitioning](#), *SIAM Journal on Scientific Computing*, **39** (2017), A2508–A2537.
- [7] J. Cooper and A. Dutle, [Spectra of uniform hypergraphs](#), *Linear Algebra and Its Applications*, **436** (2012), 3268–3292.
- [8] C. Cui, Y. Dai and J. Nie, [All real eigenvalues of symmetric tensors](#), *SIAM Journal on Matrix Analysis and Applications*, **35** (2014), 1582–1601.
- [9] G. Gaeta and E. G. Virga, Octupolar order in three dimensions, *The European Physical Journal E*, **39** (2016), 113.
- [10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th edition, The Johns Hopkins University Press, Baltimore, Maryland, 2013.
- [11] L. Han, [An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors](#), *Numerical Algebra, Control and Optimization*, **3** (2013), 583–599.
- [12] T. G. Kolda and J. R. Mayo, [Shifted power method for computing tensor eigenpairs](#), *SIAM Journal on Matrix Analysis and Applications*, **32** (2011), 1095–1124.

- [13] T. G. Kolda and J. R. Mayo, [An adaptive shifted power method for computing generalized tensor eigenpairs](#), *SIAM Journal on Matrix Analysis and Applications*, **35** (2014), 1563–1581.
- [14] W. Li and M. K. Ng, [On the limiting probability distribution of a transition probability tensor](#), *Linear and Multilinear Algebra*, **62** (2014), 362–385.
- [15] L. Lim, Singular values and eigenvalues of tensors: a variational approach, in *1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Piscataway, NJ, (2005), 129–132.
- [16] L. Qi, [Eigenvalues of a real supersymmetric tensor](#), *Journal of Symbolic Computation*, **40** (2005), 1302–1324.
- [17] L. Qi, W. Sun and Y. Wang, [Numerical multilinear algebra and its applications](#), *Frontiers of Mathematics in China*, **2** (2007), 501–526.
- [18] L. Qi and Z. Luo, *Tensor Analysis: Spectral Theory and Special Tensors*, SIAM, Philadelphia, 2017.
- [19] Y. Song and L. Qi, [Infinite and finite dimensional Hilbert tensors](#), *Linear Algebra and its Applications*, **451** (2014), 1–14.
- [20] W. Sun, [Nonmonotone trust region method for solving optimization problems](#), *Applied Mathematics and Computation*, **156** (2004), 159–174.
- [21] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, New York, 2006.
- [22] W. Sun, L. Hou and C. Dang, [A modified trust region method with Beale’s PCG technique for optimization](#), *Computational Optimization and Applications*, **40** (2008), 59–72.

Received April 2020; 1st revision August 2020; Final revision September 2020.

E-mail address: ynchen@scnu.edu.cn

E-mail address: jychang@gdut.edu.cn