Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 11, No. 2, pp. 291–307, April 2015

A FEASIBLE TRUST-REGION METHOD FOR CALCULATING EXTREME Z-EIGENVALUES OF SYMMETRIC TENSORS*

Chun-Lin Hao†, Chun-Feng Cui and Yu-Hong Dai

Abstract: It is known that computing the largest (smallest) Z-eigenvalue of a symmetric tensor is equivalent to maximizing (minimizing) a homogenous polynomial over the unit sphere. Based on such a reformulation, we shall propose a feasible trust-region method for calculating extreme Z-eigenvalues of symmetric tensors. One basic feature of the method is that the true Hessian, which is ready for polynomials, is utilized in the trust-region subproblem so that any cluster point of the iterations can be shown to satisfy the second-order necessary conditions. The other feature is that after a trial step d_k is provided by solving the trust-region subproblem at the current point x_k, the projection of $x_k + d_k$ to the unit sphere, instead of the point $x_k + d_k$ itself, is judged and if successful, is used for the next point. Global convergence and local quadratic convergence of the feasible trust-region method are established for the tensor Z-eigenvalue problem. The preliminary numerical results over several testing problems show that the feasible trust-region method is quite promising.

Key words: *symmetric tensor, extreme Z-eigenvalue, feasible trust-region, global convergence, local quadratic convergence*

Mathematics Subject Classification: *15A18, 15A69, 90C55*

1 Introduction

Eigenvalues and eigenvectors of symmetric tensors have received wide applications in diffusion tensor imaging [1, 19], signal processing [17], quantum physics [21] and independent component analysis [9] since the pioneer works by Qi [16] and Lim [11]. Among them, Z-eigenvalues, especially those extreme ones, have many applications. The largest-magnitude Z-eigenvalue λ and its related Z-eigenvector x give rise to the best rank-1 approximation λx^m of $\mathcal{A}$ [16]. The smallest Z-eigenvalue can be applied to determine the positive definiteness of an even order symmetric tensor, which plays an important part in the diffusion tensor imaging [1, 19] and the stability study of nonlinear autonomous system via Lyapunov's direct method in automatic control [13].

Consider an m-order n-dimensional symmetric tensor $\mathcal{A}$

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

*The first author was supported by the Chinese NSF grant (no. 11301016), the Research Fund for the Doctoral Program of Higher Education of China (no. 20131103120029) and Ri-Xin Talents Project of Beijing University of Technology, and the third author was supported by the National Key Basic Research Program of China (Grant No. 2015CB856000), the China National Funds for Distinguished Young Scientists (no. 11125107) and the Chinese NSF grants (nos. 11331012 and 81173633).

†Corresponding Author.

which corresponds to an m degree homogenous polynomial

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

where $x = (x_1, x_2, \ldots, x_n)^{\mathrm{T}} \in \mathbb{R}^n$. Meanwhile, the n-dimensional column vector $\mathcal{A}x^{m-1}$ can be defined in the component form

$$\left(\mathcal{A}x^{m-1}\right)_i := \sum_{i_2,\ldots,i_m=1}^{n} a_{i,i_2,\ldots,i_m} x_{i_2} \cdots x_{i_m}.$$

Moreover, the tensor $\mathcal{A}$ is said to be symmetric if its entries $a_{i_1,\ldots,i_m}$ are invariant under any permutation of their indices $\{i_1, \ldots, i_m\}$ [7, 10, 16]. In this paper, we assume that all tensors involved are symmetric.

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

$$\begin{aligned} \mathcal{A}x^{m-1} &= \lambda x, \\ x^{\mathrm{T}}x &= 1, \end{aligned} \tag{1.1}$$

then λ is a Z-eigenvalue of $\mathcal{A}$ and x is the corresponding Z-eigenvector [16].

By the variational principle, any vector x satisfying (1.1) is a KKT point of the polynomial optimization problem

$$\begin{aligned} \max_{x\in\mathbb{R}^n} \quad & \mathcal{A}x^m \\ \text{s.t.} \quad & x^{\mathrm{T}}x = 1 \end{aligned} \tag{1.2}$$

with $(\mathcal{A}x^m, x)$ being a Z-eigenpair of $\mathcal{A}$.

There have been several algorithms for finding the rank-1 tensor approximation, including the higher order power method (PM) [7, 10], the alternating least square (ALS) [25], the Newton iteration [25] and the modified power method [26]. Kolda and Mayo [8] designed the shifted power method that is guaranteed to converge to a KKT point. It should be noted that these algorithms aim to find the largest-magnitude Z-eigenvalue. Nie and Wang [14] studied semidefinite relaxations of this problem that can provide both the largest and smallest Z-eigenvalues.

There have also been several algorithms to compute eigenvalues directly. Qi *et al.* [18] put forth a direct method for the cases that $n = 2$ and that $m = n = 3$, in which all Z-eigenvalues can analytically be given. Han [3] provided an unconstrained optimization approach for finding generalized eigenpairs of even order symmetric tensors. Hao *et al.* [4] presented a sequential subspace projection method (SSPM) for solving extreme Z-eigenvalues of large scale symmetric tensors and established the linear convergence of the SSPM method. Besides, a random phase global strategy was introduced to help find extreme Z-eigenvalues. Cui *et al.* [2] came up with a new approach to compute all real eigenvalues sequentially by using Jacobian SDP relaxations of some special polynomial optimization problems. Jiang *et al.* [6] discussed approaches for calculating the extreme Z-eigenvalue of super-symmetric tensor via convex Optimization.

In this paper, we shall propose a feasible trust-region method for calculating extreme Z-eigenvalues of symmetric tensors based on the reformulation (1.2). The proposed method has two basic features. The first is that the true Hessian, which is ready for polynomials, is utilized in the trust-region subproblem so that any cluster point of the iterations can be shown to satisfy the second-order necessary conditions. Since saddle points and local minimizers will be excluded, the method seems to have a stronger ability in finding extreme

Z-eigenvalues compared with those algorithms that are only guaranteed to find a KKT point. This is important since the tensor Z-eigenvalue problem (1.2) is NP hard [5] and the largest Z-eigenvalue of a symmetric tensor is corresponding to the global maximizer of the problem (1.2). The second is that after a trial step d_k is provided by solving the trust-region subproblem at the current point x_k, the projection of $x_k + d_k$ to the unit sphere, instead of the point $x_k + d_k$ itself, is judged and if successful, is used for the next point x_{k+1}. Consequently, x_{k+1} is feasible. This is why we call our method by feasible trust-region method or simply FTR method.

The rest of this paper is organized as follows. In the next section, we describe the FTR method for the largest Z-eigenvalue of symmetric tensors. Global convergence and local quadratic convergence of the FTR method for the extreme Z-eigenvalue problem are established in Section 3. Numerical results with the FTR method are reported in Section 4, which show that the FTR is superior to both SSPM and PM. The conclusions are drawn in the last section.

2 The Feasible Trust-Region Method

In this section, we propose the feasible trust-region method (FTR) for solving (1.2). The method is different from the traditional trust-region method for general constrained optimization, in which some specific techniques [23, 24] are required for the trust-region subproblems in order to minimize the linear constraints violation. Instead, we deal with the constraints in another way for the problem (1.2). Specifically, at each iteration, if the trial step d_k is accepted, the iterate x_k+d_k is enforced to be feasible by setting $x_{k+1} = r(x_k+d_k)$, where

$$r(x) = \frac{x}{\|x\|} \tag{2.1}$$

is a projection operator to the unit sphere. Throughout this paper, $\|\cdot\|$ stands for the 2-norm. We also use I to denote the identity matrix.

The problem (1.2) can be rewritten as follows

$$\begin{aligned} \max_{x\in\mathbb{R}^n} \quad & f(x) = \frac{1}{m}\mathcal{A}x^m \\ \text{s.t.} \quad & \frac{1}{2}\left(x^{\mathrm{T}}x - 1\right) = 0. \end{aligned} \tag{2.2}$$

The Lagrangian function is

$$L(x,\lambda) = f(x) - \frac{\lambda}{2}(x^{\mathrm{T}}x - 1).$$

By this reformulation, the KKT point x^* and the related Lagrange multiplier $\lambda^* = \nabla f(x^*)^{\mathrm{T}}x^*$ of (2.2) exactly form a Z-eigenpair (λ^*, x^*) of $\mathcal{A}$.

We consider the following trust-region subproblem of (2.2) at the current feasible point x_k,

$$\begin{aligned} \max_{d\in\mathbb{R}^n} \quad & m_k(d) = f_k + g_k^{\mathrm{T}}d + \frac{1}{2}d^{\mathrm{T}}W_k\, d, \\ \text{s.t.} \quad & x_k^{\mathrm{T}}d = 0, \\ & \|d\| \le \Delta_k, \end{aligned} \tag{2.3}$$

where

$$f_k = f(x_k),$$
$$g_k = g(x_k) = \nabla f(x_k) - \lambda_k x_k, \tag{2.4}$$
$$W_k = W(x_k) = \nabla^2 f(x_k) - \lambda_k I \tag{2.5}$$

are the function value, gradient and Hessian of $L(x, \lambda)$ at (x_k, λ_k), respectively. Δ_k is the trust-region radius. For the value of λ_k, we prescribe

$$\lambda_k = \nabla f(x_k)^{\mathrm{T}} x_k = \mathcal{A} x_k^m. \tag{2.6}$$

Consider the null space subproblem of (2.3). We define the n by $n-1$ column orthogonal matrix U_k as the basis of the null space $N_{x_k} = \{d \,|\, x_k^{\mathrm{T}} d = 0\}$ of x_k. By writing

$$d = U_k q \tag{2.7}$$

and using $\|d\| = \|q\|$, we obtain the reduced trust-region model

$$\begin{aligned} \max_{q \in \mathbb{R}^{n-1}} \quad & \tilde{m}_k(q) = f_k + \tilde{g}_k^{\mathrm{T}} q + \frac{1}{2} q^{\mathrm{T}} \tilde{W}_k q \\ \text{s.t.} \quad & \|q\| \le \Delta_k, \end{aligned} \tag{2.8}$$

where $\tilde{g}_k = U_k^{\mathrm{T}} g_k \in \mathbb{R}^{n-1}$, $\tilde{W}_k = U_k^{\mathrm{T}} W_k U_k \in \mathbb{R}^{(n-1)\times(n-1)}$. One may refer to the relation (18.21) in [15] for the details of the null space method.

We now move to the outline of the feasible trust-region method. Two key ingredients in a trust-region algorithm are the strategies for choosing the trust-region radius and judging whether the trial step is accepted. Given a trial step $d_k = U_k q_k$, we define the ratio

$$\rho_k = \frac{f\left(r(x_k + d_k)\right) - f(x_k)}{m_k(d_k) - m_k(0)}. \tag{2.9}$$

The numerator in (2.9) is the actual increase of $f(x)$ and the denominator is the predicted increase. If ρ_k is close to 1, there is a good agreement between the model $m_k(d)$ and the function $f(x)$ over this step, it is safe to expand the trust-region radius at the next iteration. Considering $\|x_{k+1}\| = 1$, it is reasonable to restrict $\Delta_{k+1} \le 2$. If ρ_k is positive but significantly smaller than 1, we do not alter the trust-region, but if it is close to zero or negative, we shrink the trust-region radius by reducing Δ_k at the next iteration [15]. Specifically, Δ_{k+1} is updated as follows

$$\Delta_{k+1} = \begin{cases} \frac{1}{4}\Delta_k, & \text{if } \rho_k \le \sigma_1; \\ \min\left(2, 2\Delta_k\right), & \text{if } \rho_k > \sigma_2; \\ \Delta_k, & \text{else,} \end{cases} \tag{2.10}$$

where σ_1, σ_2 are constants with $0 < \sigma_1 < \sigma_2$ and $\sigma_1 < 1$.

Since $d_k = 0$ lies in the region $\|q\| \le \Delta_k$, the predicted increase is always nonnegative. Hence, if ρ_k is negative, the new objective value $f\left(r(x_k + d_k)\right)$ is smaller than $f(x_k)$, the step must be rejected. Specifically, the next iteration x_{k+1} is defined as

$$x_{k+1} = \begin{cases} r(x_k + d_k), & \text{if } \rho_k \ge \sigma_0; \\ x_k, & \text{else,} \end{cases} \tag{2.11}$$

where $\sigma_0 \in [0, \sigma_1)$ is a constant. Since r is the projection operator to the unit sphere, we see that the next point x_{k+1} must keep the feasibility.

The following is a description of the FTR method for computing the largest Z-eigenvalues of symmetric tensors.

Algorithm 1 The feasible trust-region method for the problem (2.2)

Step 0. Given an initial point x_0, set the parameters $\sigma_0, \sigma_1, \sigma_2, \epsilon, \Delta_0$, $\lambda_0 = \mathcal{A}x_0^m$ and $k := 0$.

Step 1. Compute g_k, W_k, U_k,$\tilde{g}_k$ and $\tilde{W}_k$, then solve the problem (2.8) to get q_k and set $d_k = U_k q_k$.

Step 2. If $\|d_k\| \leq \epsilon$, stop and output (λ_k, x_k).

Step 3. Calculate ρ_k by (2.9).

Step 4. Update the trust-region radius Δ_k by (2.10).

Step 5. If $\rho_k \geq \sigma_0$, set $x_{k+1} = r(x_k+d_k)$ and $\lambda_{k+1} = \mathcal{A}x_{k+1}^m$; else $x_{k+1} = x_k$ and $\lambda_{k+1} = \lambda_k$. Set $k := k + 1$ and go to Step 1.

3 Convergence Analysis

In this section, we establish the global convergence and local quadratic convergence of Algorithm 1 for the problem (2.2). We shall employ the techniques in traditional trust-region methods to derive the results. However, we will face some difficulties because x_{k+1} is updated by $r(x_k + d_k)$ instead of $x_k + d_k$ in order to keep the feasibility.

To simplify our analysis, define

$$h(x) = f(r(x)).$$

The gradient and Hessian of $h(x)$ are

$$\nabla h(x) = \left(\frac{I}{\|x\|} - \frac{xx^{\mathrm{T}}}{\|x\|^3}\right)\nabla f(r(x)), \tag{3.1}$$

$$\begin{aligned}\nabla^2 h(x) = {} & \frac{\nabla r(x)\nabla^2 f(r(x))}{\|x\|} - \frac{x\nabla f(r(x))^{\mathrm{T}}}{\|x\|^3} - \frac{3\nabla f(r(x))^{\mathrm{T}} xxx^{\mathrm{T}}}{\|x\|^5} \\ & - \frac{x^{\mathrm{T}}\nabla f(r(x))I + \nabla r(x)\nabla^2 f(r(x))xx^{\mathrm{T}} + \nabla f(r(x))x^{\mathrm{T}}}{\|x\|^3},\end{aligned}$$

respectively.

3.1 Global convergence

Lemma 3.1. *(i) For all x, y satisfying $x^T x = 1$ and $y^T y = 1$, we have*

$$\|W(x)\| \leq M, \tag{3.2}$$

$$\|g(x) - g(y)\| \leq L_0\|x - y\|, \tag{3.3}$$

$$\|W(x) - W(y)\| \leq L_1\|x - y\|, \tag{3.4}$$

where M, L_0 and L_1 are positive constants.

(ii) For all x, y satisfying $1 \le \|x\| \le 3$ and $1 \le \|y\| \le 3$, we have

$$\|\nabla^2 h(x) - \nabla^2 h(y)\| \le L_2 \|x - y\|, \tag{3.5}$$

where L_2 is a positive constant.

Proof. They are obvious since $g(x)$, $W(x)$ and $\nabla^2 h(x)$ are smooth and bounded on the closed sets $\{x \,|\, x^{\mathrm{T}} x = 1\}$ and $\{x \,|\, 1 \le \|x\| \le 3\}$. □

Lemma 3.2. *For the error between the model $m_k(d_k)$ and $h(x_k + d_k)$, we have*

$$|m_k(d_k) - h(x_k + d_k)| \le \beta \|d_k\|^3, \tag{3.6}$$

where β is some positive constant.

Proof. By the mean value theorem for integration, we have

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

for some $\theta_k \in (0, 1)$. Since $x_k^{\mathrm{T}} d_k = 0$, we can see by (3.1) that

$$\nabla h(x_k)^{\mathrm{T}} d_k = \nabla f(x_k)^{\mathrm{T}} (I - x_k x_k^{\mathrm{T}}) d_k = \nabla f(x_k)^{\mathrm{T}} d_k.$$

By the expression of g_k in (2.4), we also have that $g_k^T d_k = \nabla f(x_k)^T d_k$. Thus $\nabla h(x_k)^{\mathrm{T}} d_k = g_k^T d_k$. Further, noticing that $x_k^{\mathrm{T}} d_k = 0$, $x_k^{\mathrm{T}} x_k = 1$ and $\theta_k \in (0, 1)$, we can get that

$$1 \le \|x_k + \theta_k d_k\|^2 = 1 + \theta_k^2 \|d_k\|^2 \le 5.$$

Thus the point $x_k + \theta_k d_k$ still belongs to the set $\{x \,|\, 1 \le \|x\| \le 3\}$. It follows from the definition (2.3) of $m_k(d_k)$ and (3.5) that

$$\begin{aligned} |m_k(d_k) - h(x_k + d_k)| &= \left| \frac{1}{2} d_k^{\mathrm{T}} W_k d_k - \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 h(x_k + \theta_k d_k) d_k \right| \\ &= \left| \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 h(x_k) d_k - \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 h(x_k + \theta_k d_k) d_k \right| \\ &\le \frac{1}{2} L_2 \|d_k\|^3. \end{aligned}$$

Hence (3.6) holds with $\beta = \frac{1}{2} L_2$. □

Lemma 3.3. *The predicted increase of the problem (2.3) satisfies*

$$m_k(d_k) - m_k(0) \ge \frac{1}{2} \|g_k\| \min\left(\Delta_k, \frac{\|g_k\|}{\|W_k\|} \right). \tag{3.7}$$

Proof. For the problem (2.8), we know by Lemma 4.4 in [15] that

$$\tilde{m}_k(q_k) - \tilde{m}_k(0) \ge \frac{1}{2} \|\tilde{g}_k\| \min\left(\Delta_k, \frac{\|\tilde{g}_k\|}{\|\tilde{W}_k\|} \right).$$

By the Poincaré interlacing theorem (for example, see [20]), we have that $\|\tilde{W}_k\| = \|U_k^{\mathrm{T}} W_k U_k\| \le \|W_k\|$. This with $\|\tilde{g}_k\| = \|U_k^{\mathrm{T}} g_k\| = \|g_k\|$ gives

$$\begin{aligned} m_k(d_k) - m_k(0) &= \tilde{m}_k(q_k) - \tilde{m}_k(0) \\ &\ge \frac{1}{2} \|g_k\| \min\left(\Delta_k, \frac{\|g_k\|}{\|W_k\|} \right), \end{aligned}$$

which completes the proof. □

The next lemma indicates that at least one of the cluster points of $\{x_k\}$ is a KKT points of the problem (2.2).

Lemma 3.4. *Consider the sequence $\{x_k\}$ generated by Algorithm 1. Then the sequence $\{f(x_k)\}$ is nondecreasing. Furthermore, we have*

$$\liminf_{k\to 0} \|g_k\| = 0. \tag{3.8}$$

Proof. Suppose there is a constant $\epsilon > 0$ and a positive index K such that

$$\|g_k\| \geq \epsilon, \ \forall\ k \geq K. \tag{3.9}$$

Firstly, we claim that there is a constant $\bar{\Delta} > 0$ such that

$$\Delta_k \geq \bar{\Delta}, \ \forall\ k \geq K. \tag{3.10}$$

Actually,

$$\begin{aligned} \rho_k &\geq 1 - |1 - \rho_k| \\ &= 1 - \frac{|m_k(d_k) - m_k(0) - h(x_k + d_k) + h(x_k)|}{m_k(d_k) - m_k(0)} \\ &= 1 - \frac{|m_k(d_k) - h(x_k + d_k)|}{m_k(d_k) - m_k(0)}. \end{aligned} \tag{3.11}$$

It follows from Lemmas 3.2 and 3.3, (2.9) and (3.9) that

$$\begin{aligned} \rho_k &\geq 1 - \frac{2\beta\|d_k\|^3}{\|g_k\| \min\left(\Delta_k, \frac{\|g_k\|}{\|W_k\|}\right)} \\ &\geq 1 - \frac{2\beta\Delta_k^3}{\epsilon \min\left(\Delta_k, \frac{\epsilon}{M}\right)}. \end{aligned} \tag{3.12}$$

Therefore, $\rho_k > \sigma_1$ holds if $\Delta_k < \min\left\{\frac{\epsilon}{M}, \sqrt{\frac{(1-\sigma_1)\epsilon}{2\beta}}\right\}$. By (2.10), Δ_{k+1} will not shrink. As a result, (3.10) holds with

$$\bar{\Delta} = \min\left\{\Delta_K, \frac{1}{4}\min\left\{\frac{\epsilon}{M}, \sqrt{\frac{(1-\sigma_1)\epsilon}{2\beta}}\right\}\right\}.$$

Secondly, consider the following two cases for ρ_k. For the first case, assume that there is an infinite subsequence $\mathcal{K}$ such that

$$\rho_k > \sigma_1, \quad \forall\ k \in \mathcal{K}. \tag{3.13}$$

In this case,

$$\begin{aligned} h(x_{k+1}) - h(x_k) &\geq \sigma_1 \left[m_k(d_k) - m_k(0)\right] \\ &\geq \frac{1}{2}\sigma_1\epsilon \min\left(\Delta_k, \frac{\epsilon}{M}\right), \end{aligned}$$

where the last equality uses Lemma 3.3. Noting that $\{h(x_k)\}$ is nondecreasing and is bounded on the unit sphere $x^{\mathrm{T}}x = 1$, we have

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

For the second case, if (3.13) fails, there must exist an index K_0 such that

$$\rho_k \leq \sigma_1, \quad \forall\ k \geq K_0. \tag{3.14}$$

Then we know from the updating strategy (2.10) that Δ_k is multiplied by $\frac{1}{4}$ for all sufficiently large k and hence $\lim_{k\to\infty} \Delta_k = 0$. This contradicts (3.10), giving $\liminf_{k\to\infty} \|g_k\| = 0$. □

The next theorem shows that any cluster point of $\{x_k\}$ is a KKT point and satisfies the second-order necessary conditions.

Theorem 3.5. *Consider the sequence $\{x_k\}$ generated by Algorithm 1. Then*

$$\lim_{k\to 0} \|g_k\| = 0. \tag{3.15}$$

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

$$d^T W^* d \leq 0 \tag{3.16}$$

for all vector d satisfying $d^T x^* = 0$, where $W^* = W(x^*)$.

Proof. Consider any index s such that $\|g_s\| \neq 0$. For every point x in the ball

$$B(x_s, \delta) = \{x \,|\, x^{\mathrm{T}} x = 1,\ \|x - x_s\| \leq \delta\},$$

we obtain from (3.3) that

$$\|g(x)\| \geq \|g_s\| - \|g(x) - g_s\| \geq \|g_s\| - L_0 \|x - x_s\|.$$

Denoting $\delta = \frac{\|g_s\|}{2L_0}$, we have

$$\|g(x)\| \geq \|g_s\| - L_0 \delta = \frac{1}{2}\|g_s\|. \tag{3.17}$$

It follows from $\liminf_{k\to 0} \|g_k\| = 0$ in Lemma 3.4 that the ball $B(x_s, \delta)$ cannot contain the whole sequence $\{x_k\}$.

Suppose $t \geq s$ such that x_{t+1} is the first iterate after x_s outside $B(x_s, \delta)$. Then all $\{x_s, \cdots, x_t\} \in B(x_s, \delta)$ and

$$\begin{aligned} f(x_{t+1}) - f(x_s) &= \sum_{k=s}^{t} (f(x_{k+1}) - f(x_k)) \\ &\geq \sum_{k=s, x_k \neq x_{k+1}}^{t} \sigma_0 \left(m_k(d) - m_k(0)\right) \\ &\geq \sum_{k=s, x_k \neq x_{k+1}}^{t} \frac{1}{2}\sigma_0 \|g_k\| \min\left(\Delta_k, \frac{\|g_k\|}{\|W_k\|}\right) \\ &\geq \sum_{k=s, x_k \neq x_{k+1}}^{t} \frac{1}{4}\sigma_0 \|g_s\| \min\left(\Delta_k, \frac{\|g_s\|}{2M}\right), \end{aligned}$$

where the first inequality uses (2.11), the second inequality follows from Lemma 3.3 and the last one uses (3.2).

Consider the following two cases for Δ_k. On one hand, if $\Delta_k \leq \frac{\|g_s\|}{2M}$ for all $k = s, \cdots, t$, we have

$$f(x_{t+1}) - f(x_s) \geq \frac{1}{4}\sigma_0\|g_s\| \sum_{k=s, x_k \neq x_{k+1}}^{t} \Delta_k \geq \frac{1}{4}\sigma_0\|g_s\|\delta = \frac{1}{8L_0}\sigma_0\|g_s\|^2. \tag{3.18}$$

On the other hand, if $\Delta_k > \frac{\|g_s\|}{2M}$ for some $k = s, \cdots, t$,

$$f(x_{t+1}) - f(x_s) \geq \frac{1}{8M}\sigma_0\|g_s\|^2. \tag{3.19}$$

As $f(x)$ is smooth on the unit sphere, the value $f(x^*)$ of any limit point x^* must be finite. Noting that the sequence $f(x_k)$ is nondecreasing, we have

$$f(x^*) - f(x_k) \to 0,$$

which with (3.18) and (3.19) derives

$$\begin{aligned} 0 \leftarrow f(x^*) - f(x_s) &\geq f(x_{t+1}) - f(x_s) \\ &\geq \sigma_0\|g_s\|^2 \min\left(\frac{1}{8L_0}, \frac{1}{8M}\right) \end{aligned}$$

Thus we can conclude that $\lim_{s\to\infty} \|g_s\| = 0$. Therefore any cluster point x^* of $\{x_k\}$ satisfies $g(x^*) = 0$.

Now we show (3.16) by contradiction. Suppose that there exists a positive eigenvalue η_0 satisfying

$$v^{\mathrm{T}} W^* v = \eta_0 > 0, \quad \text{where } v^{\mathrm{T}} x^* = 0,\ \|v\| = 1. \tag{3.20}$$

Without loss of generality assume that $g_k^{\mathrm{T}} v \geq 0$, otherwise let $v = -v$. Consider the direction $d = \tau_k v$ for some constant $0 \leq \tau_k \leq \Delta_k$, we have

$$\begin{aligned} m_k(d) - m_k(0) &= g_k^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} W_k d \\ &= \tau_k g_k^{\mathrm{T}} v + \frac{1}{2}\tau_k^2 v^{\mathrm{T}} W_k v \\ &\geq \frac{1}{2}\|d\|^2 v^{\mathrm{T}} W_k v. \end{aligned} \tag{3.21}$$

Since $|v^{\mathrm{T}} W_k v - v^{\mathrm{T}} W^* v| \leq \|W_k - W^*\|\|v\|^2$, $\|v\| = 1$, without loss of generality we consider a convergent subsequence $\{x_k\}$ for which $W_k \to W^*$, we can see that $v^{\mathrm{T}} W_k v \to \eta_0$ when x_k is sufficiently close to x^*.

Consider the following two cases (similarly to the proof of Lemma 3.4). For the first case, assume that there is an infinite subsequence $\mathcal{K}$ such that

$$\rho_k > \sigma_1, \quad \forall\ k \in \mathcal{K}.$$

By (3.21), we have

$$f(x_{k+1}) - f(x_k) \geq \sigma_1(m_k(d_k) - m_k(0)) \geq \frac{1}{4}\sigma_1\eta_0\|d_k\|^2, \quad \forall\ k \in \mathcal{K}. \tag{3.22}$$

It follows from $f(x_{k+1}) - f(x_k) \to 0$ that $\|d_k\| \to 0$. Thus $d_k = 0$ will be the optimal solution of (2.3) for k sufficiently large; i.e., $q_k = 0$ is the optimal solution of the subproblem (2.8).

Therefore, we have that $\tilde{g}_k = 0$ and $\tilde{W}_k$ is negative semidefinite. The latter contradicts (3.20). For the second case, if there exists an index K_0 such that

$$\rho_k \leq \sigma_1, \quad \forall\ k \geq K_0,$$

we have $\lim_{k\to\infty} \Delta_k = 0$. This with (3.6), (3.11) and (3.21) indicates that $\rho_k \to 1$ and hence we obtain a contradiction with $\rho_k \leq \sigma_1$. In both cases, we can obtain some contradiction and hence the statement (3.16) is correct. The proof is completed. □

3.2 Local quadratic convergence

In this subsection, we show the local quadratic convergence of Algorithm 1.

Assumption 3.6. Suppose that for some feasible point $x^* \in \mathbb{R}^n$ and the Lagrange multiplier $\lambda^* = \nabla f(x^*)^{\mathrm{T}} x^*$ that $g(x^*) = 0$. Suppose also that

$$d^{\mathrm{T}} W^* d < 0, \tag{3.23}$$

for all $d \neq 0$ satisfying $(x^*)^{\mathrm{T}} d = 0$, where $W^* = W(x^*)$.

Lemma 3.7. *Suppose Assumption 3.6 holds for the problem (2.2). If the iterates x_k and λ_k generated by Algorithm 1 are sufficiently close to x^* and λ^*, respectively, we have*

$$\lim_{k\to\infty} \rho_k = 1 \quad \textit{and} \quad \|d_k\| < \Delta_k, \quad \forall\ k > K_0, \tag{3.24}$$

for some positive index K_0.

Proof. Denote $\tilde{W}^* = U^{*\mathrm{T}} W^* U^*$, where U^* is the basis of $N_{x^*} = \{d \mid d^{\mathrm{T}} x^* = 0\}$. By Assumption 3.6, we know that $\tilde{W}_k$ is negative definite for sufficiently large k; i.e., the largest eigenvalue is smaller than $-\eta_1$ for some constant $\eta_1 > 0$. Consider the direction

$$q = -\tau_k \tilde{W}_k^{-1} \tilde{g}_k, \tag{3.25}$$

where $\tau_k = \min\left(1, \frac{\Delta_k}{\|\tilde{W}_k^{-1}\tilde{g}_k\|}\right)$ is the stepsize. Then q is a feasible solution of the problem (2.8). We can claim that exact solution d_k of the problem (2.3) satisfies $\|d_k\| = \|q\|$. To verify this, consider the two choices of τ_k. If $\frac{\Delta_k}{\|\tilde{W}_k^{-1}\tilde{g}_k\|} < 1$, both $\|d_k\|$ and $\|q\|$ equal to Δ_k. Otherwise, if $\frac{\Delta_k}{\|\tilde{W}_k^{-1}\tilde{g}_k\|} \geq 1$, q will be the optimal solution of the problem (2.8), and $\|q\| = \|U_k q\| = \|d_k\|$. Then for the exact solution d_k,

$$\begin{aligned} m_k(d_k) - m_k(0) &\geq \tilde{m}_k(q) - \tilde{m}_k(0) \\ &= q^{\mathrm{T}} \tilde{g}_k + \frac{1}{2} q^{\mathrm{T}} \tilde{W}_k q \\ &= -\tau_k^{-1} q^{\mathrm{T}} \tilde{W}_k q + \frac{1}{2} q^{\mathrm{T}} \tilde{W}_k q \\ &\geq -\frac{1}{2} q^{\mathrm{T}} \tilde{W}_k q \geq \frac{1}{2} \eta_1 \|q\|^2 \\ &= \frac{1}{2} \eta_1 \|d_k\|^2. \end{aligned} \tag{3.26}$$

Therefore, it follows from (3.6), (3.11) and (3.26) that

$$\rho_k \to 1.$$

That is to say, there exists an index K_0 such that

$$\rho_k > \sigma_1, \quad \forall\ k \geq K_0.$$

Noting that Δ_k does not shrink and $\|\tilde{g}_k\| \to 0$, we have $\frac{\Delta_k}{\|\tilde{W}_k^{-1}\tilde{g}_k\|} > 1$. Therefore, the τ_k in (3.25) will be equal to 1 and $\Delta_k > \|\tilde{W}_k^{-1}\tilde{g}_k\| = \|d_k\|$. This completes the proof of (3.24). □

By the logics of the proof of Theorem 3.7 in [22], we can derive the following local convergence result.

Lemma 3.8. *Suppose Assumption 3.6 holds and the sequence $\{x_k\}$ generated by Algorithm 1 is sufficiently close to x^*. Then x_k and d_k satisfy*

$$\|x_k + d_k - x^*\| = O(\|x_k - x^*\|^2). \tag{3.27}$$

Proof. The trust-region constraint can be omitted since by Lemma 3.7, $\|d_k\| < \Delta_k$ always holds for sufficiently large k. The problem (2.3) is reduced to

$$\begin{aligned} \max_d \quad & m_k(d) = f_k + g_k^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} W_k\, d \\ \text{s.t.} \quad & x_k^{\mathrm{T}} d = 0. \end{aligned} \tag{3.28}$$

Consider the KKT system of (3.28). There exist $d \in \mathbb{R}^n$ and $\mu \in \mathbb{R}$ solving the system

$$\begin{pmatrix} W_k & -x_k \\ -x_k^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} d \\ \mu \end{pmatrix} = \begin{pmatrix} -g_k \\ 0 \end{pmatrix}. \tag{3.29}$$

Let $P_k = I - x_k x_k^{\mathrm{T}}$ be the projection matrix from $\mathbb{R}^n$ to N_{x_k}. By (3.29), $P_k x_k = 0$ and $\nabla f(x^*) - \lambda^* x^* = 0$, we have

$$\begin{aligned} P_k W_k d_k &= -P_k g_k = -P_k(\nabla f(x_k) - \lambda^* x_k) \\ &= -P_k(\nabla^2 f(x^*) - \lambda^* I)(x_k - x^*) + O(\|x_k - x^*\|^2) \\ &= -P_k W^*(x_k - x^*) + O(\|x_k - x^*\|^2). \end{aligned}$$

Here, $W^* = \nabla^2 f(x^*) - \lambda^* I$. Hence, we can get

$$P_k(W_k - W^*)d_k = -P_k W^*(x_k + d_k - x^*) + O(\|x_k - x^*\|^2). \tag{3.30}$$

By $x_k^{\mathrm{T}} d_k = 0$ and $2x_k^{\mathrm{T}}(x_k - x^*) = 2 - 2x_k^{\mathrm{T}} x^* = \|x_k - x^*\|^2$, we have

$$x_k^{\mathrm{T}}(x_k + d_k - x^*) = O(\|x_k - x^*\|^2). \tag{3.31}$$

Combing equations (3.30) and (3.31), we can obtain

$$\begin{pmatrix} P_k W^* \\ x_k^{\mathrm{T}} \end{pmatrix} (x_k + d_k - x^*) = \begin{pmatrix} -P_k(W_k - W^*)d_k \\ 0 \end{pmatrix} + O(\|x_k - x^*\|^2). \tag{3.32}$$

It follows from $\|P_k\| = 1$ and (3.4) that

$$\|P_k(W_k - W^*)d_k\| \leq \|W_k - W^*\|\|d_k\| \leq L_1\|x_k - x^*\|\|d_k\|. \tag{3.33}$$

Next, we claim from (3.32) and (3.33) that

$$\|x_k + d_k - x^*\| = O(\|x_k - x^*\|\|d_k\|) + O(\|x_k - x^*\|^2). \tag{3.34}$$

Denote the matrix G_k and G^* as

$$G_k = \begin{pmatrix} P_k W^* \\ x_k^{\mathrm{T}} \end{pmatrix}, \quad G^* = \begin{bmatrix} P^* W^* \\ x^{*\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{(n+1)\times n},$$

where $P^* = I - x^*(x^*)^{\mathrm{T}}$. For any $d \in \mathbb{R}^n$ satisfying $G^* d = 0$, we have $x^{*\mathrm{T}} d = 0$ and $P^* W^* d = 0$, which means that $d^{\mathrm{T}} P^* W^* d = d^{\mathrm{T}} W^* d = 0$. By (3.23) we can get that $d = 0$. Therefore, G^* has full column rank. Further, as x_k is close to x^*, the matrix G_k has also full column rank. Taking norms on the both sides of (3.32) and using (3.33), we can obtain (3.34).

Finally, we can derive from (3.34) that

$$\begin{aligned} \|d_k\| &\le \|x_k + d_k - x^*\| + \|x_k - x^*\| \\ &= O(\|x_k - x^*\|\|d_k\|) + O(\|x_k - x^*\|^2) + \|x_k - x^*\| \\ &= O(\|x_k - x^*\|), \end{aligned} \tag{3.35}$$

which with (3.34) gives (3.27). This completes the proof. □

Now we are ready to prove the local quadratic convergence of the FTR method for the tensor Z-eigenvalue problem.

Theorem 3.9. *Under the assumptions of Lemma 3.8, the sequence $\{x_k\}$ generated by Algorithm 1 satisfies*

$$\|x_{k+1} - x^*\| = O(\|x_k - x^*\|^2). \tag{3.36}$$

Proof. By the facts that $\|x_k + d_k\|^2 = 1 + d_k^{\mathrm{T}} d_k \ge 1$ and $\|x^*\| = 1$, it follows from Lemma 3.8 that

$$\begin{aligned} \|x_{k+1} - x^*\| &= \left\| \frac{x_k + d_k}{\|x_k + d_k\|} - x^* \right\| \\ &= \frac{1}{\|x_k + d_k\|} \|x_k + d_k - \|x_k + d_k\| x^*\| \\ &\le \|x_k + d_k - x^* + x^* - \|x_k + d_k\| x^*\| \\ &\le \|x_k + d_k - x^*\| + (\|x_k + d_k\| - \|x^*\|) \\ &\le \|x_k + d_k - x^*\| + \|x_k + d_k - x^*\| \\ &= O(\|x_k - x^*\|^2). \end{aligned} \tag{3.37}$$

This completes the proof. □

4 Numerical Experiments

We tested the proposed feasible trust-region method (FTR), which was compared with the sequential subspace projection method (SSPM) [4] and the high order power method (PM) [7, 10]. Some testing examples were also calculated in [14], where the exact extreme Z-eigenvalues can be computed theoretically. Therefore we are able to verify whether the algorithm returns the correct extreme Z-eigenvalues.

Our codes are implemented in MATLAB (R2013a). All the experiments are preformed on a Dell desktop with Intel dual core E6750 CPU at 2.66 GHz and 2GB of memory running Windows 7. The three models share the same normalized random initial points which obey the Gaussian distribution. The parameters for FTR are set to be

$$\sigma_0 = 0.1, \quad \sigma_1 = 0.25, \quad \sigma_2 = 0.75, \quad \epsilon = 1.0^{-5}, \quad \Delta_0 = 2.$$

The stopping criterion for both SSPM and PM is

$$\left| 1 - \frac{\nabla f(x)^{\mathrm{T}} x}{\|\nabla f(x)\| \cdot \|x\|} \right| \leq 10^{-10},$$

which naturally holds if $\nabla f(x)$ is parallel to x.

The matrix U_k in (2.7) can be generated in the following way. Consider the Householder transformation matrix H_k (that satisfies $H_k^2 = I$) such that

$$H_k x_k = -\mathrm{sign}((x_k)_1) e_1,$$

where $(x_k)_1$ is the first element of $x_k \in \mathbb{R}^n$ and $e_1 = (1, 0, \cdots, 0)^{\mathrm{T}}$. More exactly, we have

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

The matrix U_k can be chosen such that

$$[-\mathrm{sign}((x_k)_1) x_k, U_k] = H_k. \tag{4.1}$$

Therefore the computational cost to formulate the reduced problem (2.8) is $O(n^2)$.

If we change max into min in the problem (2.2), both FTR and SSPM can return the smallest Z-eigenvalue. However, the PM method looks only for the largest-magnitude Z-eigenvalue.

The following five examples in [14] are used in our tests.

Example 4.1. Calculate the largest Z-eigenvalue of the 3th-order n-dimensional symmetric tensor

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

Example 4.2. Calculate the smallest Z-eigenvalue of the 4th-order n-dimensional symmetric tensor

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

Example 4.3. Calculate the smallest Z-eigenvalue of the 4th-order n-dimensional symmetric tensor

$$\mathcal{A}_{i_1,i_2,i_3,i_4} = \tan(i_1) + \tan(i_2) + \tan(i_3) + \tan(i_4).$$

Example 4.4. Calculate the smallest Z-eigenvalue of the 4th-order n-dimensional symmetric tensor

$$\mathcal{A}_{i_1,i_2,i_3,i_4} = \sin(i_1 + i_2 + i_3 + i_4).$$

Example 4.5. Calculate the smallest Z-eigenvalue of the 5th-order n-dimensional symmetric tensor

$$\mathcal{A}_{i_1,i_2,i_3,i_4,i_5} = (-1)^{i_1} \log(i_1) + \cdots + (-1)^{i_5} \log(i_5).$$

Table 1: The numerical results of the five examples

			SSPM			FTR			PM		
Ex.	m	n	λ	iter	time	λ	iter	time	λ	iter	time
4.1	3	10	1.78e+1	5	3.00e-2	1.78e+1	3	2.80e-2	1.78e+1	34	4.4e-3
		20	3.42e+1	4	6.90e-3	3.42e+1	4	2.18e-2	3.42e+1	45	1.65e-2
		30	5.01e+1	3	1.19e-2	5.01e+1	4	2.81e-2	5.01e+1	54	4.74e-2
		40	6.59e+1	3	2.40e-2	6.59e+1	4	3.68e-2	6.59e+1	62	1.70e-1
		50	8.16e+1	2	3.67e-2	8.16e+1	4	6.44e-2	8.16e+1	58	2.06e-1
		60	9.72e+1	3	6.27e-2	9.72e+1	4	7.52e-2	9.72e+1	72	3.87e-1
		70	1.13e+2	2	6.46e-2	1.13e+2	4	1.02e-1	1.13e+2	64	5.04e-1
		80	1.28e+2	3	1.35e-1	1.28e+2	4	1.38e-1	1.28e+2	85	9.20e-1
4.2	4	10	-7.71e+1	5	1.01e-1	-7.71e+1	4	2.99e-1	-7.71e+1	68	1.52e-2
		20	-2.83e+2	5	5.60e-2	-2.83e+2	5	5.51e-2	-2.83e+2	131	2.34e-1
		30	-6.18e+2	7	3.32e-1	-6.18e+2	5	1.54e-1	-6.18e+2	208	1.54e+0
		40	-1.08e+3	3	4.42e-1	-1.08e+3	6	4.58e-1	-1.08e+3	332	7.18e+0
		50	-1.67e+3	4	1.33e+0	-1.67e+3	5	8.82e-1	-1.67e+3	315	1.60e+1
		60	-2.39e+3	7	4.95e+0	-2.39e+3	5	2.02e+0	-2.39e+3	443	4.79e+1
		70	-3.24e+3	6	8.17e+0	-3.24e+3	5	3.71e+0	-3.24e+3	496	1.09e+2
		80	-4.22e+3	4	1.19e+1	-4.22e+3	5	1.10e+1	-4.22e+3	512	2.14e+2
4.3	4	10	-5.59e+2	3	5.4e-3	-5.59e+2	4	1.77e-2	-5.59e+2	12	3.7e-3
		20	-3.69e+4	5	5.70e-2	-3.69e+4	4	4.25e-2	-3.69e+4	24	7.85e-2
		30	-6.49e+4	5	2.51e-1	-6.49e+4	4	1.26e-1	-6.49e+4	28	2.62e-1
		40	-1.07e+5	5	7.54e-1	-1.07e+5	4	3.31e-1	-1.07e+5	27	7.02e-1
		50	-1.45e+5	5	1.77e+0	-1.45e+5	4	7.88e-1	-1.45e+5	29	1.68e+0
		60	-1.96e+5	5	3.80e+0	-1.96e+5	4	1.65e+0	-1.96e+5	29	3.43e+0
		70	-2.41e+5	4	5.86e+0	-2.41e+5	4	3.23e+0	-2.41e+5	31	7.53e+0
		80	-2.98e+5	5	1.40e+1	-2.98e+5	4	6.24e+0	-2.98e+5	31	1.41e+1
4.4	4	10	-2.73e+1	21	3.47e-2	-2.27e+1	5	2.16e-2	2.56e+1	F	2.18e-1
		20	-8.96e+1	40	4.43e-1	-1.11e+2	5	5.51e-2	7.97e+1	F	1.94e+0
		30	-2.42e+2	3	1.79e-1	-2.09e+2	7	2.35e-1	-7.08e+1	F	7.64e+0
		40	-4.10e+2	24	3.29e+0	-3.90e+2	5	4.27e-1	1.01e+2	F	2.27e+1
		50	-6.25e+2	4	1.46e+0	-6.25e+2	8	1.44e+0	-1.10e+2	F	5.60e+1
		60	-9.05e+2	4	3.21e+0	-9.05e+2	8	3.25e+0	9.73e+1	F	1.22e+2
		70	-1.25e+3	4	6.14e+0	-1.20e+3	8	5.83e+0	1.23e+3	F	2.41e+2
		80	-1.65e+3	4	1.26e+1	-1.65e+3	7	1.00e+1	8.31e+2	F	4.24e+2
4.5	5	10	8.83e+2	5	1.61e+0	8.83e+2	5	7.80e-1	8.83e+2	50	2.12e+0
		20	6.24e+3	9	5.31e+1	6.24e+3	6	1.77e+1	6.24e+3	111	9.57e+1
		30	1.94e+4	9	3.48e+2	1.94e+4	6	1.15e+2	1.94e+4	170	9.62e+2
		40	4.05e+4	7	1.06e+3	4.05e+4	8	5.66e+2	4.05e+4	175	3.81e+3

The numerical results are summarized in Table 1. 'Ex.' is the number of the example, 'm' is the order, 'n' records the dimension, 'iter' stands for the iteration number, 'time' means the CPU time in seconds, and 'λ' is the Z-eigenvalue output by FTR, SSPM and PM, respectively. The algorithm is terminated once the iteration number exceeds 1000, in which case we mark with 'F'.

We shall give some comments on the numerical results.

Firstly, the FTR method has a good theoretical property; namely, it can guarantee each cluster point of the iterations satisfies the second-order necessary conditions. We see from Table 1 that both SSPM and PM can obtain the same extreme Z-eigenvalue except for Example 4.4. For this example, the PM method is divergent and fails to provide the same extreme Z-eigenvalue, SSPM terminates at a local KKT point when $n = 20$ and FTR finds

some local minimizers when $n = 30, 40, 70$. Hence, it should be noted that both FTR and SSPM are essentially looking for better local solutions. Therefore they may fail to find extreme Z-eigenvalus sometimes although they work well for most of the experiments.

Secondly, the FTR method takes only fewer than 10 iterations. FTR and SSPM are about the same except for Example 4.4.

Thirdly, the FTR method and SSPM perform much faster than PM obviously. FTR takes less time compared with SSPM for most of the experiments. We can also notice that for all the three algorithms, the cases that $m = 4$ are slower than the cases that $m = 3$, but faster than the cases that $m = 5$. The five examples are also computed in [14], where the exact largest and smallest Z-eigenvalues can be obtained simultaneously by the semidefinite relaxation method (SDPR). It is known that tensor eigenvalue problems can be reformulated into high dimensional positive semidefinite problems, which require much more time to solve. Consider Example 4.2, SDPR takes less than one minute when $5 \leq n \leq 30$; less than one hour when $35 \leq n \leq 55$, but about one hour when $n = 60$. In contrast, FTR takes less than 0.5 second when $10 \leq n \leq 40$; less than four seconds for $50 \leq n \leq 70$; and only about 11 seconds when $n = 80$. The performance of SSPM is similar to FTR. Similar observations can be obtained from some other examples.

The computational complexity of the three methods are summarized in Table 2, where '-' means this computation is not involved. It should be mentioned that in FTR, with $\mathcal{A}x^{m-2}$ available, $O(n^2)$ and $O(n^2)$ operations are required for $\mathcal{A}x^{m-1}$ and $\tilde{W}_k$ respectively. Similarly, in the three methods, with $\mathcal{A}x^{m-1}$ available, $O(n)$ operations are required for $\mathcal{A}x^m$. The subproblem of SSPM involves computing an m-order 2-dimensional tensor $\mathcal{A}_k$ by $O(m^2n^m)$ operations and solving an m-degree polynomial equation. The trust-region subproblem (2.8) is solved by the exact method [12], which costs $O(n^3)$ operations.

Table 2: The complexity of the three methods

	SSPM	FTR	PM
$\mathcal{A}x^{m-2}$	-	$O(mn^m)$	-
$\mathcal{A}x^{m-1}$	$O(mn^m)$	$O(n^2)$	$O(mn^m)$
$\mathcal{A}x^m$	$O(n)$	$O(n)$	$O(n)$
$\mathcal{A}_k$	$O(m^2n^m)$	-	-
$\tilde{W}_k$	-	$O(n^2)$	-
subprob	root of polynomial	$O(n^3)$	-

5 Conclusions

We have proposed a feasible trust-region method (FTR) for calculating Z-eigenvalues of symmetric tensors. It can deal with nonconvex/nonconcave problems because of the trust-region constraint, thus it takes more chance to escape from saddle points and local minimizers/maximizers. Therefore, the extreme Z-eigenvalues can be found with higher probability. Moreover, it keeps all the iteratives feasible. Global convergence and local quadratic convergence are established for the FTR method for the tensor Z-eigenvalue problem. In our numerical experiments, FTR is compared with the sequential subspace projection method (SSPM) and the high order power method (PM). The testing examples include symmetric tensors for $n \leq 100$ with $m = 3$, $n \leq 80$ with $m = 4$, and $n \leq 40$ with $m = 5$. FTR takes no more than 10 iterations and can terminate within about 11 seconds for all experiments except the cases that $m = 5$.

References

[1] Y. Chen, Y. Dai, D. Han and W. Sun, Positive semidefinite generalized diffusion tensor imaging via quadratic semidefinite programming, *SIAM J. Imaging Sci.* 6 (2013) 1531–1552.

[2] C. Cui, Y. Dai and J. Nie, All real eigenvalues of symmetric tensors, *SIAM J. Matrix Anal. Appl.* 35 (2014) 1582–1601.

[3] L. Han, An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors, *Numer. Algebra Control Optim.* 3 (2013) 583–599.

[4] C. Hao, C. Cui and Y. Dai, A sequential subspace projection method for solving extreme Z-eigenvalues of symmetric tensors, *Numer. Linear Algebra Appl.* doi: 10.1002/nla.1949.

[5] C. Hillar and L. Lim, Most tensor problems are NP-hard, *J. ACM* 60 (2013).

[6] B. Jiang, S. Ma an S. Zhang. *Tensor principal component analysis via convex optimization*, *Math. Program.* Ser. A. to appear, 2014.

[7] E. Kofidis and P. Regalia, On the best rank-1 approximation of higher-order supersymmetric tensors, *SIAM J. Matrix Anal. Appl.* 23 (2002) 863–884.

[8] T. Kolda, J. Mayo, Shifted power method for computing tensor eigenpairs, *SIAM J. Matrix Anal. Appl.* 32 (2011) 1095–1124.

[9] L. Lathauwer, P. Comon, B. Moor and J. Vandewalle, High-order power-application in independent component analysis, in Procedding of the International Symposium on Nonlinear Theory and its Applications (NOLTA'95) 1995, pp. 91–96.

[10] L. Lathauwer, B. Moor and J. Vandewalle, On the best rank-1 and rank-$(R_1, R_2, ..., R_N)$ approximation of higher-order tensors, *SIAM J. Matrix Anal. Appl.* 21 (2000) 1324–1342.

[11] L. Lim, Singular values and eigenvalues of tensors: a variational approach, in Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Addaptive Processing (CAMSAP'05), Vol.1, IEEE Computer Society Press: Piscataway, New Jersey, 2005, pp. 129–132.

[12] J. Moré, D. Sorensen, Computing a trust region step, *SIAM J. Sci. Comput.* 4 (1983) 553–572.

[13] Q. Ni, L. Qi and F. Wang, An eigenvalue method for testing positive definiteness of a multivariate form, *IEEE Trans. Automat. Control.* 53 (2008) 1096–1107.

[14] J. Nie and L. Wang, Semidefinite relaxations for best rank-1 tensor approximations, *SIAM J. Matrix Anal. Appl.* 35 (2014) 1155–1179.

[15] J. Nocedal and S. Wright, *Numerical Optimization*, Springerverlang, USA, 1999.

[16] L. Qi, Eigenvalues of a real supersymmetric tensor, *J. Symbolic Comput.* 40 (2005) 1302–1324.

[17] L. Qi and K. Teo, Multivariate polynomial minimization and its application in signal processing, *J. Global Optim.* 26 (2003) 419–433.

[18] L. Qi, F. Wang and Y. Wang, Z-eigenvalue methods for a global polynomial optimization problem, *Math. Program.* 118 (2009) 301–316.

[19] L. Qi, G. Yu and E. Wu, Higer order positive semidefinite diffusion tensor imaging, *SIAM J. Imaging Sci.* 3 (2010) 416–433.

[20] S. Wang, M. Wu and Z. Jia, *Matrix Inequality*, Science Press, 2ed., 2006 (in Chinese).

[21] T. Wei and P. Goldbart, Geometric measure of entanglement and applications to bipartite and multipartite quantum states, *Phys. Rev.* A 68 (2003) 042307.

[22] S. Xi, *Nonlinear Optimization Methods*, High education press, China, 1992 (in Chinese).

[23] Y. Yuan, A review of trust region algorithms for optimization in *Proceedings of the 4th International Con-gress on Industrial & Applied Mathematics (ICIAM 99)*, Edinburgh, pp. 271–282.

[24] Y. Yuan and W. Sun, *The Optimization Theory and Methods*, Science press, China, 1997 (in Chinese).

[25] T. Zhang and G. Golub, Rank-1 approximation to higher-order tensors, *SIAM J. Matrix Anal. Appl.* 23 (2001) 534-550.

[26] X. Zhang, C. Ling and L. Qi, The best rank-1 approximation of a symmetric tensor and related spherical optimization problems, *SIAM J. Matrix Anal. Appl.* 33 (2012) 806-821.

Manuscript received 16 May 2014
revised 2 October 2014, 2 Decembe 2014
accepted for publication 3 December 2014

Chun-Lin Hao
School of Applied Mathmatics, Beijing University of Technology
Pingleyuan 100, Chaoyang district, Beijing, China
E-mail address: haochl@bjut.edu.cn

Chun-Feng Cui
Academy of Mathematics and Systems Science
Haidian District, Beijing 100190, P. R. China
E-mail address: Email: cuichf@lsec.cc.ac.cn

Yu-Hong Dai
Academy of Mathematics and Systems Science
Haidian District, Beijing 100190, P. R. China
E-mail address: dyh@lsec.cc.ac.cn