J Sci Comput (2016) 68:716–738
DOI 10.1007/s10915-015-0155-8

# Computing Extreme Eigenvalues of Large Scale Hankel Tensors

**Yannan Chen**[1] · **Liqun Qi**[2] · **Qun Wang**[2]

Received: 10 May 2015 / Revised: 8 November 2015 / Accepted: 22 December 2015 /
Published online: 4 January 2016


**Abstract** Large scale tensors, including large scale Hankel tensors, have many applications in science and engineering. In this paper, we propose an inexact curvilinear search optimization method to compute Z- and H-eigenvalues of $m$th order $n$ dimensional Hankel tensors, where $n$ is large. Owing to the fast Fourier transform, the computational cost of each iteration of the new method is about $\mathcal{O}(mn\log(mn))$. Using the Cayley transform, we obtain an effective curvilinear search scheme. Then, we show that every limiting point of iterates generated by the new algorithm is an eigen-pair of Hankel tensors. Without the assumption of a second-order sufficient condition, we analyze the linear convergence rate of iterate sequence by the Kurdyka–Łojasiewicz property. Finally, numerical experiments for Hankel tensors, whose dimension may up to one million, are reported to show the efficiency of the proposed curvilinear search method.

**Keywords** Cayley transform · Curvilinear search · Extreme eigenvalue · Fast Fourier transform · Hankel tensor · Kurdyka–Łojasiewicz property · Large scale tensor

**Mathematics Subject Classification** 15A18 · 15A69 · 65F15 · 65K05 · 90C52

Yannan Chen: This author's work was supported by the National Natural Science Foundation of China (Grant No. 11401539) and the Development Foundation for Excellent Youth Scholars of Zhengzhou University (Grant No. 1421315070). Liqun Qi: This author's work was partially supported by the Hong Kong Research Grant Council (Grant No. PolyU 502111, 501212, 501913 and 15302114).

✉ Liqun Qi
maqilq@polyu.edu.hk

Yannan Chen
ynchen@zzu.edu.cn

Qun Wang
wangqun876@gmail.com

[1] School of Mathematics and Statistics, Zhengzhou University, Zhengzhou, China

[2] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

## 1 Introduction

With the coming era of massive data, large scale tensors have important applications in science and engineering. How to store and analyze these tensors? This is a pressing and challenging problem. In the literature, there are two strategies for manipulating large scale tensors. The first one is to exploit their structures such as sparsity [3]. For example, we consider an online store (e.g. Amazon.com) where users may review various products [35]. Then, a third order tensor with modes: users, items, and words could be formed naturally and it is sparse. The other one is to use distributed and parallel computation [12,16]. This technique could deal with large scale dense tensors, but it depends on a supercomputer. Recently, researchers applied these two strategies simultaneously for large scale tensors [11,28].

In this paper, we consider a class of large scale dense tensors with a special Hankel structure. Hankel tensors appear in many engineering problems such as signal processing [6,18], automatic control [48], and geophysics [39,50]. For instance, in nuclear magnetic resonance spectroscopy [52], a Hankel matrix was formed to analyze the time-domain signals, which is important for brain tumour detection. Papy et al. [40,41] improved this method by using a high order Hankel tensor to replace the Hankel matrix. Ding et al. [18] proposed a fast computational framework for products of a Hankel tensor and vectors. On the mathematical properties, Luque and Thibon [34] explored the Hankel hyperdeterminants. Qi [43] and Xu [54] studied the spectra of Hankel tensors and gave some upper bounds and lower bounds for the smallest and the largest eigenvalues. In [43], Qi raised a question: Can we construct some efficient algorithms for the largest and the smallest H- and Z-eigenvalues of a Hankel tensor?

Numerous applications of the eigenvalues of higher order tensors have been found in science and engineering, such as automatic control [37], medical imaging [9,45,47], quantum information [36], and spectral graph theory [13]. For example, in magnetic resonance imaging [45], the principal Z-eigenvalues of an even order tensor associated to the fiber orientation distribution of a voxel in white matter of human brain denote volume factions of several nerve fibers in this voxel, and the corresponding Z-eigenvectors express the orientations of these nerve fibers. The smallest eigenvalue of tensors reflects the stability of a nonlinear multivariate autonomous system in automatic control [37]. For a given even order symmetric tensor, it is positive semidefinite if and only if its smallest H- or Z-eigenvalue is nonnegative [42].

The conception of eigenvalues of higher order tensors was defined independently by Qi [42] and Lim [32] in 2005. Unfortunately, it is an NP-hard problem to compute eigenvalues of a tensor even though the involved tensor is symmetric [26]. For two and three dimensional symmetric tensors, Qi et al. [44] proposed a direct method to compute all of its Z-eigenvalues. It was pointed out in [30,31] that the polynomial system solver, `NSolve` in *Mathematica*, could be used to compute all of the eigenvalues of lower order and low dimensional tensors. We note that the mathematical software *Maple* has a similar command `solve` which is also applicable for the polynomial systems of eigenvalues of tensors.

For general symmetric tensors, Kolda and Mayo [30] proposed a shifted symmetric higher order power method to compute its Z-eigenpairs. Recently, they [31] extended the shifted power method to generalized eigenpairs of tensors and gave an adaptive shift. Based on the nonlinear optimization model with a compact unit spherical constraint, the power methods [17] project the gradient of the objective at the current iterate onto the unit sphere at each iteration. Its computation is very simple but may not converge [29]. Kolda and Mayo [30,31] introduced a shift to force the objective to be (locally) concave/convex. Then the power method produces increasing/decreasing steps for computing maximal/minimal eigenvalues. The sequence of objectives converges to eigenvalues since the feasible region is compact. The convergence of the sequence of iterates to eigenvectors is established under the assumption that the tensor has finitely many real eigenvectors. The linear convergence rate is estimated by a fixed-point analysis.

Inspired by the power method, various optimization methods have been established. Han [23] proposed an unconstrained optimization model, which is indeed a quadratic penalty function of the constrained optimization for generalized eigenvalues of symmetric tensors. Hao et al. [24] employed a subspace projection method for Z-eigenvalues of symmetric tensors. Restricted by a unit spherical constraint, this method minimizes the objective in a big circle of $n$ dimensional unit sphere at each iteration. Since the objective is a homogeneous polynomial, the minimization of the subproblem has a closed-form solution. Additionally, Hao et al. [25] gave a trust region method to calculate Z-eigenvalues of symmetric tensors. The sequence of iterates generated by this method converges to a second order critical point and enjoys a locally quadratic convergence rate.

Since nonlinear optimization methods may produce a local minimizer, some convex optimization models have been studied. Hu et al. [27] addressed a sequential semi-definite programming method to compute the extremal Z-eigenvalues of tensors. A sophisticated Jacobian semi-definite relaxation method was explored by Cui et al. [14]. A remarkable feature of this method is the ability to compute all of the real eigenvalues of symmetric tensors. Recently, Chen et al. [8] proposed homotopy continuation methods to compute all of the complex eigenvalues of tensors. When the order or the dimension of a tensor grows larger, the CPU times of these methods become longer and longer.

In some applications [39,52], the scales of Hankel tensors can be quite huge. This highly restricted the applications of the above mentioned methods in this case. How to compute the smallest and the largest eigenvalues of a Hankel tensor? Can we propose a method to compute the smallest and the largest eigenvalues of a relatively large Hankel tensor, say $1,000,000$ dimension? This is one of the motivations of this paper.

Owing to the multi-linearity of tensors, we model the problem of eigenvalues of Hankel tensors as a nonlinear optimization problem with a unit spherical constraint. Our algorithm is an inexact steepest descent method on the unit sphere. To preserve iterates on the unit sphere, we employ the Cayley transform to generate an orthogonal matrix such that the new iterate is this orthogonal matrix times the current iterate. By the Sherman-Morrison-Woodbury formula, the product of the orthogonal matrix and a vector has a closed-form solution. So the subproblem is straightforward. A curvilinear search is employed to guarantee the convergence. Then, we prove that every accumulation point of the sequence of iterates is an eigenvector of the involved Hankel tensor, and its objective is the corresponding eigenvalue. Furthermore, using the Kurdyka–Łojasiewicz property of the eigen-problem of tensors, we prove that the sequence of iterates converges without an assumption of second order sufficient condition. Under mild conditions, we show that the sequence of iterates has a linear or a sublinear convergence rate. Numerical experiments show that this strategy is successful.

The outline of this paper is drawn as follows. We introduce a fast computational framework for products of a well-structured Hankel tensor and vectors in Sect. 2. The computational cost is cheap. In Sect. 3, we show the technique of using the Cayley transform to construct an effective curvilinear search algorithm. The convergence of objective and iterates are analyzed in Sect. 4. The Kurdyka–Łojasiewicz property is applied to analyze an inexact line search method. Numerical experiments in Sect. 5 address that the new method is efficient and promising. Finally, we conclude the paper with Sect. 6.

## 2 Hankel Tensors

Suppose $\mathcal{A}$ is an $m$th order $n$ dimensional real symmetric tensor

$$\mathcal{A} = (a_{i_1,i_2,\dots,i_m}), \qquad \text{for } i_j = 1, \dots, n,\ j = 1, \dots, m,$$

where all of the entries are real and invariant under any index permutation. Two products of the tensor $\mathcal{A}$ and a column vector $\mathbf{x} \in \mathbb{R}^n$ used in this paper are defined as follows.

- $\mathcal{A}\mathbf{x}^m$ is a scalar

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

- $\mathcal{A}\mathbf{x}^{m-1}$ is a column vector

$$\left(\mathcal{A}\mathbf{x}^{m-1}\right)_i = \sum_{i_2,\dots,i_m=1}^{n} a_{i,i_2,\dots,i_m} x_{i_2} \cdots x_{i_m}, \qquad \text{for } i = 1, \dots, n.$$

When the tensor $\mathcal{A}$ is dense, the computations of products $\mathcal{A}\mathbf{x}^m$ and $\mathcal{A}\mathbf{x}^{m-1}$ require $\mathcal{O}(n^m)$ operations, since the tensor $\mathcal{A}$ has $n^m$ entries and we must visit all of them in the process of calculation. When the tensor is symmetric, the computational cost for these products is about $\mathcal{O}(n^m/m!)$ [46]. Obviously, they are expensive. In this section, we will study a special tensor, the Hankel tensor, whose elements are completely determined by a generating vector. So there exists a fast algorithm to compute products of a Hankel tensor and vectors. Let us give the definitions of two structured tensors.

**Definition 1** An $m$th order $n$ dimensional tensor $\mathcal{H}$ is called a Hankel tensor if its entries satisfy

$$h_{i_1,i_2,\dots,i_m} = v_{i_1+i_2+\cdots+i_m-m}, \qquad \text{for } i_j = 1, \dots, n,\ j = 1, \dots, m.$$

The vector $\mathbf{v} = (v_0, v_1, \dots, v_{m(n-1)})^\top$ with length $\ell \equiv m(n-1)+1$ is called the generating vector of the Hankel tensor $\mathcal{H}$.

An $m$th order $\ell$ dimensional tensor $\mathcal{C}$ is called an anti-circulant tensor if its entries satisfy

$$c_{i_1,i_2,\dots,i_m} = v_{(i_1+i_2+\cdots+i_m-m \bmod \ell)}, \qquad \text{for } i_j = 1, \dots, \ell,\ j = 1, \dots, m.$$

It is easy to see that $\mathcal{H}$ is a sub-tensor of $\mathcal{C}$. Since for the same generating vector $\mathbf{v}$ we have

$$c_{i_1,i_2,\dots,i_m} = h_{i_1,i_2,\dots,i_m}, \qquad \text{for } i_j = 1, \dots, n,\ j = 1, \dots, m.$$

For example, a third order two dimensional Hankel tensor with a generating vector $\mathbf{v} = (v_0, v_1, v_2, v_3)^\top$ is

$$\mathcal{H} = \left[\begin{array}{cc|cc} v_0 & v_1 & v_1 & v_2 \\ v_1 & v_2 & v_2 & v_3 \end{array}\right].$$

It is a sub-tensor of an anti-circulant tensor with the same order and a larger dimension

$$\mathcal{C} = \left[\begin{array}{cccc|cccc|cccc|cccc} v_0 & v_1 & v_2 & v_3 & v_1 & v_2 & v_3 & v_0 & v_2 & v_3 & v_0 & v_1 & v_3 & v_0 & v_1 & v_2 \\ v_1 & v_2 & v_3 & v_0 & v_2 & v_3 & v_0 & v_1 & v_3 & v_0 & v_1 & v_2 & v_0 & v_1 & v_2 & v_3 \\ v_2 & v_3 & v_0 & v_1 & v_3 & v_0 & v_1 & v_2 & v_0 & v_1 & v_2 & v_3 & v_1 & v_2 & v_3 & v_0 \\ v_3 & v_0 & v_1 & v_2 & v_0 & v_1 & v_2 & v_3 & v_1 & v_2 & v_3 & v_0 & v_2 & v_3 & v_0 & v_1 \end{array}\right].$$

As discovered in [18, Theorem 3.1], the $m$th order $\ell$ dimensional anti-circulant tensor $\mathcal{C}$ could be diagonalized by the $\ell$-by-$\ell$ Fourier matrix $F_\ell$, i.e., $\mathcal{C} = \mathcal{D}F_\ell^m$, where $\mathcal{D}$ is a diagonal tensor whose diagonal entries are $\mathrm{diag}(\mathcal{D}) = F_\ell^{-1}\mathbf{v}$. It is well-known that the computations involving the Fourier matrix and its inverse times a vector are indeed the fast (inverse) Fourier transform `fft` and `ifft`, respectively. The computational cost is about $\mathcal{O}(\ell \log \ell)$ multiplications, which is significantly smaller than $\mathcal{O}(\ell^2)$ for a dense matrix times a vector when the dimension $\ell$ is large.

Now, we are ready to show how to compute the products introduced in the beginning of this section, when the involved tensor has a Hankel structure. For any $\mathbf{x} \in \mathbb{R}^n$, we define another vector $\mathbf{y} \in \mathbb{R}^\ell$ such that

$$\mathbf{y} \equiv \begin{bmatrix} \mathbf{x} \\ \mathbf{0}_{\ell-n} \end{bmatrix},$$

where $\ell = m(n-1)+1$ and $\mathbf{0}_{\ell-n}$ is a zero vector with length $\ell - n$. Then, we have

$$\mathcal{H}\mathbf{x}^m = \mathcal{C}\mathbf{y}^m = \mathcal{D}(F_\ell \mathbf{y})^m = \ \mathtt{ifft}(\mathbf{v})^\top \left(\mathtt{fft}(\mathbf{y})^{\circ m}\right).$$

To obtain $\mathcal{H}\mathbf{x}^{m-1}$, we first compute

$$\mathcal{C}\mathbf{y}^{m-1} = F_\ell \left(\mathcal{D}(F_\ell \mathbf{y})^{m-1}\right) = \ \mathtt{fft}\Big( \ \mathtt{ifft}(\mathbf{v}) \circ \Big( \ \mathtt{fft}(\mathbf{y})^{\circ(m-1)}\Big)\Big).$$

Then, the entries of vector $\mathcal{H}\mathbf{x}^{m-1}$ is the leading $n$ entries of $\mathcal{C}\mathbf{y}^{m-1}$. Here, $\circ$ denotes the Hadamard product such that $(A \circ B)_{i,j} = A_{i,j}B_{i,j}$. Three matrices $A$, $B$ and $A \circ B$ have the same size. Furthermore, we define $A^{\circ k} = A \circ \cdots \circ A$ as the Hadamard product of $k$ copies of $A$.

Since the computations of $\mathcal{H}\mathbf{x}^m$ and $\mathcal{H}\mathbf{x}^{m-1}$ require 2 and 3 `fft/iffts`, the computational cost is about $\mathcal{O}(mn\log(mn))$ and obviously cheap. Another advantage of this approach is that we do not need to store and deal with the tremendous Hankel tensor explicitly. It is sufficient to keep and work with the compact generating vector of that Hankel tensor.

## 3 A Curvilinear Search Algorithm

We consider the generalized eigenvalue [7,19] of an $m$th order $n$ dimensional Hankel tensor $\mathcal{H}$

$$\mathcal{H}\mathbf{x}^{m-1} = \lambda \mathcal{B}\mathbf{x}^{m-1},$$

where $m$ is even, $\mathcal{B}$ is an $m$th order $n$ dimensional symmetric tensor and it is positive definite. If there is a scalar $\lambda$ and a real vector $\mathbf{x}$ satisfying this system, we call $\lambda$ a generalized eigenvalue and $\mathbf{x}$ its associated generalized eigenvector. Particularly, we find the following definitions from the literature, where the computation on the tensor $\mathcal{B}$ is straightforward.

- Qi [42] called a real scalar $\lambda$ a Z-eigenvalue of a tensor $\mathcal{H}$ and a real vector $\mathbf{x}$ its associated Z-eigenvector if they satisfy

$$\mathcal{H}\mathbf{x}^{m-1} = \lambda \mathbf{x} \quad \text{and} \quad \mathbf{x}^\top \mathbf{x} = 1.$$

  This definition means that the tensor $\mathcal{B}$ is an identity tensor $\mathcal{E}$ such that $\mathcal{E}\mathbf{x}^{m-1} = \|\mathbf{x}\|^{m-2}\mathbf{x}$.
- If $\mathcal{B} = \mathcal{I}$, where

$$(\mathcal{I})_{i_1,\ldots,i_m} = \begin{cases} 1 & \text{if } i_1 = \cdots = i_m, \\ 0 & \text{otherwise}, \end{cases}$$

  the real scalar $\lambda$ is called an H-eigenvalue and the real vector $\mathbf{x}$ is its associated H-eigenvector [42]. Obviously, we have $(\mathcal{I}\mathbf{x}^{m-1})_i = x_i^{m-1}$ for $i = 1, \ldots, n$.

To compute a generalized eigenvalue and its associated eigenvector, we consider the following optimization model with a spherical constraint

$$\min \quad f(\mathbf{x}) \equiv \frac{\mathcal{H}\mathbf{x}^m}{\mathcal{B}\mathbf{x}^m} \qquad \text{s.t.} \ \ \|\mathbf{x}\| = 1, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm or its induced matrix norm. The denominator of the objective is positive since the tensor $\mathcal{B}$ is positive definite. By some calculations, we get its gradient and Hessian, which are formally presented in the following lemma.

**Lemma 1** *Suppose that the objective is defined as in (1). Then, its gradient is*

$$\mathbf{g}(\mathbf{x}) = \frac{m}{\mathcal{B}\mathbf{x}^m}\left(\mathcal{H}\mathbf{x}^{m-1} - \frac{\mathcal{H}\mathbf{x}^m}{\mathcal{B}\mathbf{x}^m}\mathcal{B}\mathbf{x}^{m-1}\right). \tag{2}$$

*And its Hessian is*

$$\begin{aligned} H(\mathbf{x}) = &\frac{m(m-1)\mathcal{H}\mathbf{x}^{m-2}}{\mathcal{B}\mathbf{x}^m} - \frac{m(m-1)\mathcal{H}\mathbf{x}^m\mathcal{B}\mathbf{x}^{m-2} + m^2(\mathcal{H}\mathbf{x}^{m-1}\circledcirc\mathcal{B}\mathbf{x}^{m-1})}{(\mathcal{B}\mathbf{x}^m)^2} \\ &+ \frac{m^2\mathcal{H}\mathbf{x}^m(\mathcal{B}\mathbf{x}^{m-1}\circledcirc\mathcal{B}\mathbf{x}^{m-1})}{(\mathcal{B}\mathbf{x}^m)^3}, \end{aligned} \tag{3}$$

*where* $\mathbf{x}\circledcirc\mathbf{y} \equiv \mathbf{x}\mathbf{y}^\top + \mathbf{y}\mathbf{x}^\top$.

Let $\mathbb{S}_{n-1} \equiv \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^\top\mathbf{x} = 1\}$ be the spherical feasible region. Suppose the current iterate is $\mathbf{x} \in \mathbb{S}_{n-1}$ and the gradient at $\mathbf{x}$ is $\mathbf{g}(\mathbf{x})$. Because

$$\mathbf{x}^\top\mathbf{g}(\mathbf{x}) = \frac{m}{\mathcal{B}\mathbf{x}^m}\left(\mathbf{x}^\top\mathcal{H}\mathbf{x}^{m-1} - \frac{\mathcal{H}\mathbf{x}^m}{\mathcal{B}\mathbf{x}^m}\mathbf{x}^\top\mathcal{B}\mathbf{x}^{m-1}\right) = 0, \tag{4}$$

the gradient $\mathbf{g}(\mathbf{x})$ of $\mathbf{x} \in \mathbb{S}_{n-1}$ is located in the tangent plane of $\mathbb{S}_{n-1}$ at $\mathbf{x}$.

**Lemma 2** *Suppose* $\|\mathbf{g}(\mathbf{x})\| = \epsilon$, *where* $\mathbf{x} \in \mathbb{S}_{n-1}$ *and* $\epsilon$ *is a small number. Denote* $\lambda = \frac{\mathcal{H}\mathbf{x}^m}{\mathcal{B}\mathbf{x}^m}$. *Then, we have*

$$\|\mathcal{H}\mathbf{x}^{m-1} - \lambda\mathcal{B}\mathbf{x}^{m-1}\| = \mathcal{O}(\epsilon).$$

*Moreover, if the gradient* $\mathbf{g}(\mathbf{x})$ *at* $\mathbf{x}$ *vanishes, then* $\lambda = f(\mathbf{x})$ *is a generalized eigenvalue and* $\mathbf{x}$ *is its associated generalized eigenvector.*

*Proof* Recalling the definition of gradient (2), we have

$$\|\mathcal{H}\mathbf{x}^{m-1} - \lambda\mathcal{B}\mathbf{x}^{m-1}\| = \frac{\mathcal{B}\mathbf{x}^m}{m}\epsilon.$$

Since the tensor $\mathcal{B}$ is positive definite and the vector $\mathbf{x}$ belongs to a compact set $\mathbb{S}_{n-1}$, $\mathcal{B}\mathbf{x}^m$ has a finite upper bound. Thus, the first assertion is valid.

If $\epsilon = 0$, we immediately know that $\lambda = f(\mathbf{x})$ is a generalized eigenvalue and $\mathbf{x}$ is its associated generalized eigenvector. □

Next, we construct the curvilinear search path using the Cayley transform [22]. Cayley transform is an effective method which could preserve the orthogonal constraints. It has various applications in the inverse eigenvalue problem [20], $p$-harmonic flow [21], and matrix optimization [53].

Suppose the current iterate is $\mathbf{x}_k \in \mathbb{S}_{n-1}$ and the next iterate is $\mathbf{x}_{k+1}$. To preserve the spherical constraint $\mathbf{x}_{k+1}^\top\mathbf{x}_{k+1} = \mathbf{x}_k^\top\mathbf{x}_k = 1$, we choose the next iterate $\mathbf{x}_{k+1}$ such that

$$\mathbf{x}_{k+1} = Q\mathbf{x}_k, \tag{5}$$

where $Q \in \mathbb{R}^{n\times n}$ is an orthogonal matrix, whose eigenvalues do not contain $-1$. Using the Cayley transform, the matrix

$$Q = (I + W)^{-1}(I - W) \tag{6}$$

is orthogonal if and only if the matrix $W \in \mathbb{R}^{n\times n}$ is skew-symmetric.[1] Now, our task is to select a suitable skew-symmetric matrix $W$ such that $\mathbf{g}(\mathbf{x}_k)^\top(\mathbf{x}_{k+1} - \mathbf{x}_k) < 0$. For simplicity, we take the matrix $W$ as

$$W = \mathbf{a}\mathbf{b}^\top - \mathbf{b}\mathbf{a}^\top, \tag{7}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ are two undetermined vectors. From (5) and (6), we have

$$\mathbf{x}_{k+1} - \mathbf{x}_k = -W(\mathbf{x}_k + \mathbf{x}_{k+1}).$$

Then, by (7), it yields that

$$\mathbf{g}(\mathbf{x}_k)^\top(\mathbf{x}_{k+1} - \mathbf{x}_k) = -[(\mathbf{g}(\mathbf{x}_k)^\top\mathbf{a})\mathbf{b}^\top - (\mathbf{g}(\mathbf{x}_k)^\top\mathbf{b})\mathbf{a}^\top](\mathbf{x}_k + \mathbf{x}_{k+1}).$$

For convenience, we choose

$$\mathbf{a} = \mathbf{x}_k \quad \text{and} \quad \mathbf{b} = -\alpha\mathbf{g}(\mathbf{x}_k). \tag{8}$$

Here, $\alpha$ is a positive parameter, which serves as a step size, so that we have some freedom to choose the next iterate. According to this selection and (4), we obtain

$$\begin{aligned}\mathbf{g}(\mathbf{x}_k)^\top(\mathbf{x}_{k+1} - \mathbf{x}_k) &= -\alpha\|\mathbf{g}(\mathbf{x}_k)\|^2\mathbf{x}_k^\top(\mathbf{x}_k + \mathbf{x}_{k+1})\\ &= -\alpha\|\mathbf{g}(\mathbf{x}_k)\|^2(1 + \mathbf{x}_k^\top Q\mathbf{x}_k).\end{aligned}$$

Since $-1$ is not an eigenvalue of the orthogonal matrix $Q$, we have $1 + \mathbf{x}_k^\top Q\mathbf{x}_k > 0$ for $\mathbf{x}_k^\top\mathbf{x}_k = 1$. Therefore, the conclusion $\mathbf{g}(\mathbf{x}_k)^\top(\mathbf{x}_{k+1} - \mathbf{x}_k) < 0$ holds for any positive step size $\alpha$.

We summarize the iterative process in the following Theorem.

**Theorem 1** *Suppose that the new iterate $\mathbf{x}_{k+1}$ is generated by (5), (6), (7), and (8). Then, the following assertions hold.*

- *The iterative scheme is*

$$\mathbf{x}_{k+1}(\alpha) = \frac{1 - \alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}{1 + \alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}\mathbf{x}_k - \frac{2\alpha}{1 + \alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}\mathbf{g}(\mathbf{x}_k). \tag{9}$$

- *The progress made by $\mathbf{x}_{k+1}$ is*

$$\mathbf{g}(\mathbf{x}_k)^\top(\mathbf{x}_{k+1}(\alpha) - \mathbf{x}_k) = -\frac{2\alpha\|\mathbf{g}(\mathbf{x}_k)\|^2}{1 + \alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}. \tag{10}$$

[1] See "http://en.wikipedia.org/wiki/Cayley_transform".

*Proof* From the equality (4) and the Sherman-Morrison-Woodbury formula, we have

$$\begin{aligned}
\mathbf{x}_{k+1}(\alpha) &= (I - \alpha \mathbf{x}_k \mathbf{g}(\mathbf{x}_k)^\top + \alpha \mathbf{g}(\mathbf{x}_k)\mathbf{x}_k^\top)^{-1}(I + \alpha \mathbf{x}_k \mathbf{g}(\mathbf{x}_k)^\top - \alpha \mathbf{g}(\mathbf{x}_k)\mathbf{x}_k^\top)\mathbf{x}_k \\
&= (I + \alpha \mathbf{g}(\mathbf{x}_k)\mathbf{x}_k^\top - \alpha \mathbf{x}_k \mathbf{g}(\mathbf{x}_k)^\top)^{-1}(\mathbf{x}_k - \alpha \mathbf{g}(\mathbf{x}_k)) \\
&= \left(I - \begin{bmatrix} \alpha \mathbf{g}(\mathbf{x}_k) & -\mathbf{x}_k \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \mathbf{x}_k^\top \\ \alpha \mathbf{g}(\mathbf{x}_k)^\top \end{bmatrix} I \begin{bmatrix} \alpha \mathbf{g}(\mathbf{x}_k) & -\mathbf{x}_k \end{bmatrix}\right)^{-1} \cdot \right. \\
&\qquad \left. \begin{bmatrix} \mathbf{x}_k^\top \\ \alpha \mathbf{g}(\mathbf{x}_k)^\top \end{bmatrix}\right)(\mathbf{x}_k - \alpha \mathbf{g}(\mathbf{x}_k)) \\
&= \mathbf{x}_k - \alpha \mathbf{g}(\mathbf{x}_k) - \begin{bmatrix} \alpha \mathbf{g}(\mathbf{x}_k) & -\mathbf{x}_k \end{bmatrix} \begin{bmatrix} 1 & -1 \\ \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ -\alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2 \end{bmatrix} \\
&= \frac{1 - \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2}{1 + \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2} \mathbf{x}_k - \frac{2\alpha}{1 + \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2} \mathbf{g}(\mathbf{x}_k).
\end{aligned}$$

The proof of (10) is straightforward. □

Whereafter, we devote to choose a suitable step size $\alpha$ by an inexact curvilinear search. At the beginning, we give a useful theorem.

**Theorem 2** *Suppose that the new iterate $\mathbf{x}_{k+1}(\alpha)$ is generated by (9). Then, we have*

$$\left.\frac{\mathrm{d} f(\mathbf{x}_{k+1}(\alpha))}{\mathrm{d}\alpha}\right|_{\alpha=0} = -2\|\mathbf{g}(\mathbf{x}_k)\|^2.$$

*Proof* By some calculations, we get

$$\mathbf{x}'_{k+1}(\alpha) = \frac{-2}{1 + \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2} \mathbf{g}(\mathbf{x}_k) + \frac{-4\alpha \|\mathbf{g}(\mathbf{x}_k)\|^2}{(1 + \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2)^2} (\mathbf{x}_k - \alpha \mathbf{g}(\mathbf{x}_k)).$$

Hence, $\mathbf{x}'_{k+1}(0) = -2\mathbf{g}(\mathbf{x}_k)$. Furthermore, $\mathbf{x}_{k+1}(0) = \mathbf{x}_k$. Therefore, we obtain

$$\left.\frac{\mathrm{d} f(\mathbf{x}_{k+1}(\alpha))}{\mathrm{d}\alpha}\right|_{\alpha=0} = \mathbf{g}(\mathbf{x}_{k+1}(0))^\top \mathbf{x}'_{k+1}(0) = \mathbf{g}(\mathbf{x}_k)^\top(-2\mathbf{g}(\mathbf{x}_k)) = -2\|\mathbf{g}(\mathbf{x}_k)\|^2.$$

The proof is completed. □

According to Theorem 2, for any constant $\eta \in (0, 2)$, there exists a positive scalar $\tilde{\alpha}$ such that for all $\alpha \in (0, \tilde{\alpha}]$,

$$f(\mathbf{x}_{k+1}(\alpha)) - f(\mathbf{x}_k) \le -\eta\alpha \|\mathbf{g}(\mathbf{x}_k)\|^2.$$

Hence, the curvilinear search process is well-defined.

Now, we present a curvilinear search algorithm (ACSA) formally in Algorithm 1 for the smallest generalized eigenvalue and its associated eigenvector of a Hankel tensor. If our aim is to compute the largest generalized eigenvalue and its associated eigenvector of a Hankel tensor, we only need to change respectively (9) and (11) used in Steps 5 and 6 of the ACSA algorithm to

$$\mathbf{x}_{k+1}(\alpha) = \frac{1 - \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2}{1 + \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2} \mathbf{x}_k + \frac{2\alpha}{1 + \alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2} \mathbf{g}(\mathbf{x}_k),$$

and

$$f(\mathbf{x}_{k+1}(\alpha_k)) \ge f(\mathbf{x}_k) + \eta\alpha_k \|\mathbf{g}(\mathbf{x}_k)\|^2.$$

When the Z-eigenvalue of a Hankel tensor is considered, we have $\mathcal{E}\mathbf{x}^m = \|\mathbf{x}\|^m = 1$ and the objective $f(\mathbf{x})$ is a polynomial. Then, we could compute the global minimizer of the step

**Algorithm 1** A curvilinear search algorithm (ACSA).

Give the generating vector $\mathbf{v}$ of a Hankel tensor $\mathcal{H}$, the symmetric tensor $\mathcal{B}$, an initial unit iterate $\mathbf{x}_1$, parameters $\eta \in (0, \frac{1}{2}]$, $\beta \in (0, 1)$, $\bar{\alpha}_1 = 1 \le \alpha_{\max}$, and $k \leftarrow 1$.
**while** the sequence of iterates does not converge **do**
  Compute $\mathcal{H}\mathbf{x}_k^m$ and $\mathcal{H}\mathbf{x}_k^{m-1}$ by the fast computational framework introduces in Sect. 2.
  Calculate $\mathcal{B}\mathbf{x}_k^m$, $\mathcal{B}\mathbf{x}_k^{m-1}$, $\lambda_k = f(\mathbf{x}_k) = \frac{\mathcal{H}\mathbf{x}_k^m}{\mathcal{B}\mathbf{x}_k^m}$ and $\mathbf{g}(\mathbf{x}_k)$ by (2).
  Choose the smallest nonnegative integer $\ell$ and determine $\alpha_k = \beta^\ell \bar{\alpha}_k$ such that

$$f(\mathbf{x}_{k+1}(\alpha_k)) \le f(\mathbf{x}_k) - \eta\alpha_k \|\mathbf{g}(\mathbf{x}_k)\|^2, \tag{11}$$

  where $\mathbf{x}_{k+1}(\alpha)$ is calculated by (9).
  Update the iterate $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}(\alpha_k)$.
  Choose an initial step size $\bar{\alpha}_{k+1} \in (0, \alpha_{\max}]$ for the next iteration.
  $k \leftarrow k + 1$.
**end while**

size $\alpha_k$ (the exact line search) in each iteration as [24]. However, we use a cheaper inexact line search here. The initial step size of the next iteration follows Dai's strategy [15]

$$\bar{\alpha}_{k+1} = \frac{\|\Delta \mathbf{x}_k\|}{\|\Delta \mathbf{g}_k\|}, \tag{12}$$

which is the geometric mean of Barzilai-Borwein step sizes [4].

## 4 Convergence Analysis

Since the optimization model (1) has a nice algebraic nature, we will use the Kurdyka–Łojasiewicz property [5,33] to analyze the convergence of the proposed ACSA algorithm. Before we start, we give some basic convergence results.

### 4.1 Basic Convergence Results

If the ACSA algorithm terminates finitely, there exists a positive integer $k$ such that $\mathbf{g}(\mathbf{x}_k) = 0$. According to Lemma 2, $f(\mathbf{x}_k)$ is a generalized eigenvalue and $\mathbf{x}_k$ is its associated generalized eigenvector.

Next, we assume that ACSA generates an infinite sequence of iterates.

**Lemma 3** *Suppose that the even order symmetric tensor $\mathcal{B}$ is positive definite. Then, all the functions, gradients, and Hessians of the objective (1) at feasible points are bounded. That is to say, there is a positive constant $M$ such that for all $\mathbf{x} \in \mathbb{S}_{n-1}$*

$$|f(\mathbf{x})| \le M, \quad \|\mathbf{g}(\mathbf{x})\| \le M, \quad \textit{and} \quad \|H(\mathbf{x})\| \le M. \tag{13}$$

*Proof* Since the spherical feasible region $\mathbb{S}_{n-1}$ is compact, the denominator $\mathcal{B}\mathbf{x}^m$ of the objective is positive and bounds away from zero. Recalling Lemma 1, we get this theorem immediately. □

**Theorem 3** *Suppose that the infinite sequence $\{\lambda_k\}$ is generated by ACSA. Then, the sequence $\{\lambda_k\}$ is monotonously decreasing. And there exists a $\lambda_*$ such that*

$$\lim_{k\to\infty} \lambda_k = \lambda_*.$$

*Proof* Since $\lambda_k = f(\mathbf{x}_k)$ which is bounded and monotonously decreasing, the infinite sequence $\{\lambda_k\}$ must converge to a unique $\lambda_*$. □

This theorem means that the sequence of generalized eigenvalues converges. To show the convergence of iterates, we first prove that the step sizes bound away from zero.

**Lemma 4** *Suppose that the step size $\alpha_k$ is generated by ACSA. Then, for all iterations k, we get*

$$\alpha_k \geq \frac{(2-\eta)\beta}{5M} \equiv \alpha_{\min} > 0. \tag{14}$$

*Proof* Let $\underline{\alpha} \equiv \frac{2-\eta}{5M}$. According to the curvilinear search process of ACSA, it is sufficient to prove that the inequality (11) holds if $\alpha_k \in (0, \underline{\alpha}]$.

From the iterative formula (9) and the equality (4), we get

$$\begin{aligned}\|\mathbf{x}_{k+1}(\alpha) - \mathbf{x}_k\|^2 &= \left\| \frac{-2\alpha^2 \|\mathbf{g}(\mathbf{x}_k)\|^2}{1+\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}\mathbf{x}_k - \frac{2\alpha}{1+\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}\mathbf{g}(\mathbf{x}_k)\right\|^2 \\ &= \frac{4\alpha^4\|\mathbf{g}(\mathbf{x}_k)\|^4\|\mathbf{x}_k\|^2 + 4\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}{(1+\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2)^2} \\ &= \frac{4\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}{1+\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}.\end{aligned}$$

Hence,

$$\|\mathbf{x}_{k+1}(\alpha) - \mathbf{x}_k\| = \frac{2\alpha\|\mathbf{g}(\mathbf{x}_k)\|}{\sqrt{1+\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}}. \tag{15}$$

From the mean value theorem, (9), (4), and (15), we have

$$\begin{aligned}& f(\mathbf{x}_{k+1}(\alpha)) - f(\mathbf{x}_k) \leq \mathbf{g}(\mathbf{x}_k)^\top(\mathbf{x}_{k+1}(\alpha) - \mathbf{x}_k) + \frac{1}{2}M\|\mathbf{x}_{k+1}(\alpha) - \mathbf{x}_k\|^2 \\ &= \frac{1}{1+\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}\left(-2\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2\mathbf{g}(\mathbf{x}_k)^\top\mathbf{x}_k - 2\alpha\|\mathbf{g}(\mathbf{x}_k)\|^2 + \frac{M}{2}4\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2\right) \\ &\leq \frac{\alpha\|\mathbf{g}(\mathbf{x}_k)\|^2}{1+\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}(4\alpha M - 2).\end{aligned}$$

It is easy to show that for all $\alpha \in (0, \underline{\alpha}]$

$$4\alpha M - 2 \leq -\eta(1+\alpha^2 M^2).$$

Therefore, we have

$$f(\mathbf{x}_{k+1}(\alpha)) - f(\mathbf{x}_k) \leq \frac{-\eta(1+\alpha^2M^2)}{1+\alpha^2\|\mathbf{g}(\mathbf{x}_k)\|^2}\alpha\|\mathbf{g}(\mathbf{x}_k)\|^2 \leq -\eta\alpha\|\mathbf{g}(\mathbf{x}_k)\|^2.$$

The proof is completed. □

**Theorem 4** *Suppose that the infinite sequence $\{\mathbf{x}_k\}$ is generated by ACSA. Then, the sequence $\{\mathbf{x}_k\}$ has an accumulation point at least. And we have*

$$\lim_{k\to\infty} \|\mathbf{g}(\mathbf{x}_k)\| = 0. \tag{16}$$

*That is to say, every accumulation point of $\{\mathbf{x}_k\}$ is a generalized eigenvector whose associated generalized eigenvalue is $\lambda_*$.*

*Proof* Since the sequence of objectives $\{f(\mathbf{x}_k)\}$ is monotonously decreasing and bounded, by (11) and (14), we have

$$2M \geq f(\mathbf{x}_1) - \lambda_* = \sum_{k=1}^{\infty} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \sum_{k=1}^{\infty} \eta\alpha_k \|\mathbf{g}(\mathbf{x}_k)\|^2 \geq \eta\alpha_{\min} \sum_{k=1}^{\infty} \|\mathbf{g}(\mathbf{x}_k)\|^2.$$

It yields that

$$\sum_k \|\mathbf{g}(\mathbf{x}_k)\|^2 \leq \frac{2M}{\eta\alpha_{\min}} < +\infty. \tag{17}$$

Thus, the limit (16) holds.

Let $\mathbf{x}_\infty$ be an accumulation point of $\{\mathbf{x}_k\}$. Then $\mathbf{x}_\infty$ belongs to the compact set $\mathbb{S}_{n-1}$ and $\|\mathbf{g}(\mathbf{x}_\infty)\| = 0$. According to Lemma 2, $\mathbf{x}_\infty$ is a generalized eigenvector whose associated eigenvalue is $f(\mathbf{x}_\infty) = \lambda_*$. □

### 4.2 Further Results Based on the Kurdyka–Łojasiewicz Property

In this subsection, we will prove that the iterates $\{\mathbf{x}_k\}$ generated by ACSA converge without an assumption of the second-order sufficient condition. The key tool of our analysis is the Kurdyka–Łojasiewicz property. This property was first discovered by S. Łojasiewicz [33] in 1963 for real-analytic functions. Bolte et al. [5] extended this property to nonsmooth subanalytic functions. Whereafter, the Kurdyka–Łojasiewicz property was widely applied to analyze regularized algorithms for nonconvex optimization [1,2]. Significantly, it seems to be new to use the Kurdyka–Łojasiewicz property to analyze an inexact line search algorithm, e.g., ACSA proposed in Sect. 3.

We now write down the Kurdyka–Łojasiewicz property [5, Theorem 3.1] for completeness.

**Theorem 5** (Kurdyka–Łojasiewicz (KL) property) *Suppose that $\mathbf{x}_*$ is a critical point of $f(\mathbf{x})$. Then there is a neighborhood $\mathscr{U}$ of $\mathbf{x}_*$, an exponent $\theta \in [0, 1)$, and a constant $C_1$ such that for all $\mathbf{x} \in \mathscr{U}$, the following inequality holds*

$$\frac{|f(\mathbf{x}) - f(\mathbf{x}_*)|^\theta}{\|\mathbf{g}(\mathbf{x})\|} \leq C_1. \tag{18}$$

*Here, we define $0^0 \equiv 1$.*

**Lemma 5** *Suppose that $\mathbf{x}_*$ is one of the accumulation points of $\{\mathbf{x}_k\}$. For the convenience of using the Kurdyka–Łojasiewicz property, we assume that the initial iterate $\mathbf{x}_1$ satisfies $\mathbf{x}_1 \in \mathscr{B}(\mathbf{x}_*, \rho) \equiv \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{x}_*\| < \rho\} \subseteq \mathscr{U}$ where*

$$\rho > \frac{2C_1}{\eta(1-\theta)} |f(\mathbf{x}_1) - f(\mathbf{x}_*)|^{1-\theta} + \|\mathbf{x}_1 - \mathbf{x}_*\|.$$

*Then, we have the following two assertions:*

$$\mathbf{x}_k \in \mathscr{B}(\mathbf{x}_*, \rho), \qquad \forall\, k = 1, 2, \ldots, \tag{19}$$

*and*

$$\sum_k \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \frac{2C_1}{\eta(1-\theta)} |f(\mathbf{x}_1) - f(\mathbf{x}_*)|^{1-\theta}. \tag{20}$$

*Proof* We prove (19) by the induction. First, it is easy to see that $\mathbf{x}_1 \in \mathscr{B}(\mathbf{x}_*, \rho)$. Next, we assume that there is an integer $K$ such that

$$\mathbf{x}_k \in \mathscr{B}(\mathbf{x}_*, \rho), \qquad \forall\, 1 \leq k \leq K.$$

Hence, the KL property (18) holds in these iterates. Finally, we prove that $\mathbf{x}_{K+1} \in \mathscr{B}(\mathbf{x}_*, \rho)$.

For the convenience of presentation, we define a scalar function

$$\varphi(s) \equiv \frac{C_1}{1-\theta}|s - f(\mathbf{x}_*)|^{1-\theta}.$$

Obviously, $\varphi(s)$ is a concave function and its derivative is $\varphi'(s) = \frac{C_1}{|s-f(\mathbf{x}_*)|^{\theta}}$ if $s > f(\mathbf{x}_*)$. Then, for any $1 \leq k \leq K$, we have

$$\begin{aligned}
\varphi(f(\mathbf{x}_k)) - \varphi(f(\mathbf{x}_{k+1})) &\geq \varphi'(f(\mathbf{x}_k))(f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})) \\
&= \frac{C_1}{|f(\mathbf{x}_k) - f(\mathbf{x}_*)|^{\theta}}(f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})) \\
\text{[by KL property]} \qquad &\geq \frac{1}{\|\mathbf{g}(\mathbf{x}_k)\|}(f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})) \\
\text{[since (11)]} \qquad &\geq \frac{1}{\|\mathbf{g}(\mathbf{x}_k)\|}\eta\alpha_k\|\mathbf{g}(\mathbf{x}_k)\|^2 \\
&\geq \frac{\eta\alpha_k\|\mathbf{g}(\mathbf{x}_k)\|}{\sqrt{1+\alpha_k^2\|\mathbf{g}(\mathbf{x}_k)\|^2}} \\
\text{[because of (15)]} \qquad &\geq \frac{\eta}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|.
\end{aligned}$$

It yields that

$$\begin{aligned}
\sum_{k=1}^{K}\|\mathbf{x}_{k+1} - \mathbf{x}_k\| &\leq \frac{2}{\eta}\sum_{k=1}^{K}\varphi(f(\mathbf{x}_k)) - \varphi(f(\mathbf{x}_{k+1})) \\
&= \frac{2}{\eta}(\varphi(f(\mathbf{x}_1)) - \varphi(f(\mathbf{x}_{K+1}))) \\
&\leq \frac{2}{\eta}\varphi(f(\mathbf{x}_1)). \qquad (21)
\end{aligned}$$

So, we get

$$\begin{aligned}
\|\mathbf{x}_{K+1} - \mathbf{x}_*\| &\leq \sum_{k=1}^{K}\|\mathbf{x}_{k+1} - \mathbf{x}_k\| + \|\mathbf{x}_1 - \mathbf{x}_*\| \\
&\leq \frac{2}{\eta}\varphi(f(\mathbf{x}_1)) + \|\mathbf{x}_1 - \mathbf{x}_*\| \\
&< \rho.
\end{aligned}$$

Thus, $\mathbf{x}_{K+1} \in \mathscr{B}(\mathbf{x}_*, \rho)$ and (19) holds.

Moreover, let $K \to \infty$ in (21). We obtain (20). □

**Theorem 6** *Suppose that the infinite sequence of iterates* $\{\mathbf{x}_k\}$ *is generated by ACSA. Then, the total sequence* $\{\mathbf{x}_k\}$ *has a finite length, i.e.,*

$$\sum_{k}\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < +\infty,$$

*and hence the total sequence* $\{\mathbf{x}_k\}$ *converges to a unique critical point.*

*Proof* Since the domain of $f(\mathbf{x})$ is compact, the infinite sequence $\{\mathbf{x}_k\}$ generated by ACSA must have an accumulation point $\mathbf{x}_*$. According to Theorem 4, $\mathbf{x}_*$ is a critical point. Hence, there exists an index $k_0$, which could be viewed as an initial iteration when we use Lemma 5, such that $\mathbf{x}_{k_0} \in \mathscr{B}(\mathbf{x}_*, \rho)$. From Lemma 5, we have $\sum_{k=k_0}^{\infty} \|\mathbf{x}_{k+1} - \mathbf{x}_k\| < +\infty$. Therefore, the total sequence $\{\mathbf{x}_k\}$ has a finite length and converges to a unique critical point. □

Next, we give an estimation for the convergence rate of ACSA, which is a specialization of Theorem 2 of Attouch and Bolte [1]. The proof here is clearer since we have a new bound in Lemma 6.

**Lemma 6** *There exists a positive constant $C_2$ such that*

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \geq C_2 \|\mathbf{g}(\mathbf{x}_k)\|. \tag{22}$$

*Proof* Since $\alpha_{\max} \geq \alpha_k \geq \alpha_{\min} > 0$ and (15), we have

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| = \frac{2\alpha_k \|\mathbf{g}(\mathbf{x}_k)\|}{\sqrt{1 + \alpha_k^2 \|\mathbf{g}(\mathbf{x}_k)\|^2}} \geq \frac{2\alpha_{\min}}{1 + \alpha_{\max} M} \|\mathbf{g}(\mathbf{x}_k)\|.$$

Let $C_2 \equiv \frac{2\alpha_{\min}}{1+\alpha_{\max} M}$. We get this lemma. □

The following theorem is almost the same as in the one in [1] on convergence rates.

**Theorem 7** *Suppose that $\mathbf{x}_*$ is the critical point of the infinite sequence of iterates $\{\mathbf{x}_k\}$ generated by ACSA. Then, we have the following estimations.*

- *If $\theta \in (0, \frac{1}{2}]$, there exists a $\gamma > 0$ and $\varrho \in (0, 1)$ such that*

$$\|\mathbf{x}_k - \mathbf{x}_*\| \leq \gamma \varrho^k.$$

- *If $\theta \in (\frac{1}{2}, 1)$, there exists a $\gamma > 0$ such that*

$$\|\mathbf{x}_k - \mathbf{x}_*\| \leq \gamma k^{-\frac{1-\theta}{2\theta-1}}.$$

*Proof* Without loss of generality, we assume that $\mathbf{x}_1 \in \mathscr{B}(\mathbf{x}_*, \rho)$. For convenience of following analysis, we define

$$\Delta_k \equiv \sum_{i=k}^{\infty} \|\mathbf{x}_i - \mathbf{x}_{i+1}\| \geq \|\mathbf{x}_k - \mathbf{x}_*\|.$$

Then, we have

$$\begin{aligned}
\Delta_k &= \sum_{i=k}^{\infty} \|\mathbf{x}_i - \mathbf{x}_{i+1}\| \\
\text{[since (20)]} \quad &\leq \frac{2C_1}{\eta(1-\theta)} |f(\mathbf{x}_k) - f(\mathbf{x}_*)|^{1-\theta} \\
&= \frac{2C_1}{\eta(1-\theta)} \left(|f(\mathbf{x}_k) - f(\mathbf{x}_*)|^{\theta}\right)^{\frac{1-\theta}{\theta}} \\
\text{[KL property]} \quad &\leq \frac{2C_1}{\eta(1-\theta)} \left(C_1 \|\mathbf{g}(\mathbf{x}_k)\|\right)^{\frac{1-\theta}{\theta}} \\
\text{[for (22)]} \quad &\leq \frac{2C_1}{\eta(1-\theta)} \left(C_1 C_2^{-1} \|\mathbf{x}_k - \mathbf{x}_{k+1}\|\right)^{\frac{1-\theta}{\theta}}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{2C_1^{\frac{1}{\theta}} C_2^{-\frac{1-\theta}{\theta}}}{\eta(1-\theta)} \left(\Delta_k - \Delta_{k+1}\right)^{\frac{1-\theta}{\theta}} \\
&\equiv C_3 \left(\Delta_k - \Delta_{k+1}\right)^{\frac{1-\theta}{\theta}}, \qquad (23)
\end{aligned}
$$

where $C_3$ is a positive constant.

If $\theta \in (0, \frac{1}{2})$, we have $\frac{1-\theta}{\theta} \geq 1$. When the iteration $k$ is large enough, the inequality (23) implies that

$$\Delta_k \leq C_3(\Delta_k - \Delta_{k+1}).$$

That is

$$\Delta_{k+1} \leq \frac{C_3 - 1}{C_3} \Delta_k.$$

Hence, recalling $\|\mathbf{x}_k - \mathbf{x}_*\| \leq \Delta_k$, we obtain the estimation if we take $\varrho \equiv \frac{C_3-1}{C_3}$.

Otherwise, we consider the case $\theta \in (\frac{1}{2}, 1)$. Let $h(s) = s^{-\frac{\theta}{1-\theta}}$. Obviously, $h(s)$ is monotonously decreasing. Then, the inequality (23) could be rewritten as

$$
\begin{aligned}
C_3^{-\frac{\theta}{1-\theta}} &\leq h(\Delta_k)(\Delta_k - \Delta_{k+1}) \\
&= \int_{\Delta_{k+1}}^{\Delta_k} h(\Delta_k) \,\mathrm{d}s \\
&\leq \int_{\Delta_{k+1}}^{\Delta_k} h(s) \,\mathrm{d}s \\
&= -\frac{1-\theta}{2\theta - 1} \left( \Delta_k^{-\frac{2\theta-1}{1-\theta}} - \Delta_{k+1}^{-\frac{2\theta-1}{1-\theta}} \right).
\end{aligned}
$$

Denote $\nu \equiv -\frac{2\theta-1}{1-\theta} < 0$ since $\theta \in (\frac{1}{2}, 1)$. Then, we get

$$\Delta_{k+1}^{\nu} - \Delta_k^{\nu} \geq -\nu C_3^{-\frac{\theta}{1-\theta}} \equiv C_4 > 0.$$

It yields that for all $k > K$,

$$\Delta_k \leq [\Delta_K^{\nu} + C_4(k - K)]^{\frac{1}{\nu}} \leq \gamma k^{\frac{1}{\nu}},$$

where the last inequality holds when the iteration $k$ is sufficiently large. □

We remark that if the Hessian $H(\mathbf{x}_*)$ at the critical point $\mathbf{x}_*$ is positive definite, the key parameter $\theta$ in the Kurdyka–Łojasiewicz property is $\theta = \frac{1}{2}$. Under Theorem 7, the sequence of iterates generated by ACSA has a linear convergence rate. In this viewpoint, the Kurdyka–Łojasiewicz property is weaker than the second order sufficient condition of $\mathbf{x}_*$ being a minimizer.

## 5 Numerical Experiments

To show the efficiency of the proposed ACSA algorithm, we perform some numerical experiments. The parameters used in ACSA are

$$\eta = .001, \qquad \beta = .5, \qquad \alpha_{\max} = 10000.$$

**Table 1** Computed Z-eigenvalues of the Hankel tensor in Example 1

| Algorithms | Power M. | Han's UOA | ACSA-general | ACSA-Hankel |
|---|---|---|---|---|
| −8.846335 | 54 % | 58 % | 72 % | 72 % |
| −3.920428 | 46 % | 42 % | 28 % | 28 % |
| CPU t. (sec) | 23.09 | 9.34 | 8.39 | 0.67 |

We terminate the algorithm if the objectives satisfy

$$\frac{|\lambda_{k+1} - \lambda_k|}{\max(1, |\lambda_k|)} < 10^{-12}\sqrt{n}$$

or the number of iterations exceeds 1000. The codes are written in MATLAB R2012a and run in a desktop computer with Intel Core E8500 CPU at 3.17GHz and 4GB memory running Windows 7.

We will compare the following four algorithms in this section.

- An adaptive shifted power method [30,31] (Power M.) is implemented as `eig_sshopm` and `eig_geap` in Tensor Toolbox 2.6 for Z- and H-eigenvalues of even order symmetric tensors.
- An unconstrained optimization approach [23] (Han's UOA) is solved by `fminunc` in MATLAB with settings: `GradObj:on, LargeScale:off, TolX:1.e-10, TolFun:1.e-8, MaxIter:10000, Display:off.`
- For general symmetric tensors without considering a Hankel structure, we implement ACSA as ACSA-general.
- The ACSA algorithm (ACSA-Hankel) is proposed in Sect. 3 for Hankel tensors.

### 5.1 Small Hankel Tensors

First, we examine some small tensors, whose Z- and H-eigenvalues could be computed exactly.

*Example 1* ([38]) A Hankel tensor $\mathcal{A}$ whose entries are defined as

$$a_{i_1 i_2 \cdots i_m} = \sin(i_1 + i_2 + \cdots + i_m), \qquad i_j = 1, 2, \ldots, n, \ j = 1, 2, \ldots, m.$$

Its generating vector is $\mathbf{v} = (\sin(m), \sin(m+1), \ldots, \sin(mn))^\top$.

If $m = 4$ and $n = 5$, there are five Z-eigenvalues which are listed as follows [8,14]

$$\lambda_1 = 7.2595, \quad \lambda_2 = 4.6408, \quad \lambda_3 = 0.0000, \quad \lambda_4 = -3.9204, \quad \lambda_5 = -8.8463.$$

We test four kinds of algorithms: power method, Han's UOA, ACSA-general and ACSA-Hankel. For the purpose of obtaining the smallest Z-eigenvalue of the Hankel tensor, we select 100 random initial points on the unit sphere. The entries of each initial point is first chosen to have a Gaussian distribution, then we normalize it to a unit vector. The resulting Z-eigenvalues and CPU times are reported in Table 1. All of the four methods find the smallest Z-eigenvalue −8.846335. But the occurrences for each method finding the smallest Z-eigenvalue are different. We say that the ACSA algorithm proposed in Sect. 3 could find the extreme eigenvalues with a higher probability.

Form the viewpoint of totally computational times, ACSA-general, and ACSA-Hankel are faster than the power method and Han's UOA. When the Hankel structure of a fourth

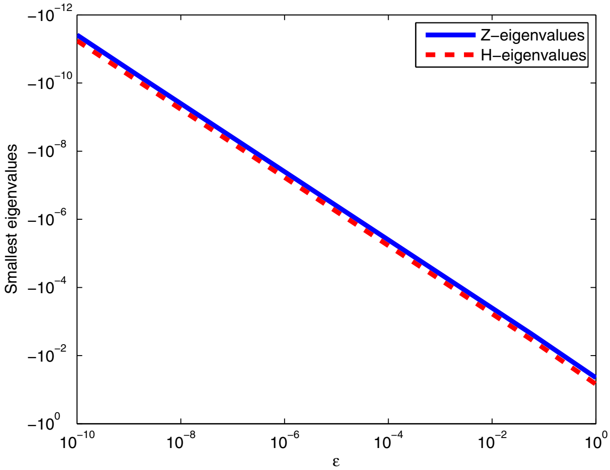


**Fig. 1** The smallest Z- and H-eigenvalues of the parameterized fourth order four dimensional Hankel tensors

order five dimensional symmetric tensor $\mathcal{A}$ is exploited, it is unexpected that the new method is about 30 times faster than the power method.

*Example 2* We study a parameterized fourth order four dimensional Hankel tensor $\mathcal{H}_\epsilon$ whose generating vector has the following form

$$\mathbf{v}_\epsilon = (8-\epsilon, 0, 2, 0, 1, 0, 1, 0, 1, 0, 2, 0, 8-\epsilon)^\top.$$

If $\epsilon = 0$, $\mathcal{H}_0$ is positive semidefinite but not positive definite [10]. When the parameter $\epsilon$ is positive and trends to zero, the smallest Z- and H-eigenvalues are negative and trends to zero. In this example, we will illustrate this phenomenon by a numerical approach.

Again, we compare the power method, Han's UOA, ACSA-general, and ACSA-Hankel for computing the smallest Z- and H-eigenvalues of the parameterized Hankel tensors in Example 2. For the purpose of accuracy, we slightly modify the setting `TolX:1.e-12`, `TolFun:1.e-12` for Han's UOA. In each case, thirty random initial points on a unit sphere are selected to obtain the smallest Z- or H-eigenvalues. When the parameter $\epsilon$ decreases from 1 to $10^{-10}$, the smallest Z- and H-eigenvalues returned by these four algorithm are congruent. We show this results in Fig. 1. When $\epsilon$ trends to zero, the smallest Z- and H-eigenvalues are negative and going to zero too.

The detailed CPU times for these four algorithms computing the smallest Z- and H-eigenvalues of the parameterized fourth order four dimensional Hankel tensors are drawn in Table 2. Obviously, even without exploiting the Hankel structure, ACSA-general is two times faster than the power method and Han's UOA. Furthermore, when the fast computational framework for the products of a Hankel tensor time vectors is explored, ACSA-Hankel saves about 90 % CPU times.

## 5.2 Large Scale Problems

When the Hankel structure of higher order tensors is explored, we could compute eigenvalues and associated eigenvectors of large scale Hankel tensors.

**Table 2** CPU times (second) for computing Z- and H-eigenvalues of the parameterized Hankel tensors shown in Example 2

| Algorithms | Power M. | Han's UOA | ACSA-general | ACSA-Hankel |
|---|---|---|---|---|
| Z-eigenvalues | 41.980 | 46.629 | 17.878 | 1.498 |
| H-eigenvalues | 29.562 | 45.833 | 16.973 | 1.544 |
| Total CPU times | 71.542 | 92.462 | 34.851 | 3.042 |

*Example 3* A Vandermonde tensor [43,54] is a special Hankel tensor. Let

$$\alpha = \frac{n}{n-1} \qquad \text{and} \qquad \beta = \frac{1-n}{n}.$$

Then, $\mathbf{u}_1 = (1, \alpha, \alpha^2, \ldots, \alpha^{n-1})^\top$ and $\mathbf{u}_2 = (1, \beta, \beta^2, \ldots, \beta^{n-1})^\top$ are two Vandermonde vectors. The following $m$th order $n$ dimensional symmetric tensor

$$\mathcal{H}_V = \underbrace{\mathbf{u}_1 \otimes \mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_1}_{m \text{ times}} + \underbrace{\mathbf{u}_2 \otimes \mathbf{u}_2 \otimes \cdots \otimes \mathbf{u}_2}_{m \text{ times}}$$

is a Vandermonde tensor which satisfies the Hankel structure. Here $\otimes$ is the outer product. Obviously, the generating vector of $\mathcal{H}_V$ is $\mathbf{v} = (2, \alpha + \beta, \ldots, \alpha^{m(n-1)} + \beta^{m(n-1)})^\top$.

**Proposition 1** *Suppose the mth order n dimensional Hankel tensor $\mathcal{H}_V$ is defined as in Example 3. Then, when n is even, the largest Z-eigenvalue of $\mathcal{H}_V$ is $\|\mathbf{u}_1\|^m$ and its associated eigenvector is $\frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}$.*

*Proof* Since $\alpha\beta = -1$ and $n$ is even, $\mathbf{u}_1$ and $\mathbf{u}_2$ are orthogonal. We consider the optimization problem

$$\begin{aligned} \max \quad & \mathcal{H}_V \mathbf{x}^m = (\mathbf{u}_1^\top \mathbf{x})^m + (\mathbf{u}_2^\top \mathbf{x})^m, \\ \text{s.t.} \quad & \mathbf{x}^\top \mathbf{x} = 1. \end{aligned}$$

Since $\|\mathbf{u}_1\| > \|\mathbf{u}_2\|$, when $\mathbf{x} = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}$, the above optimization problem obtains its maximal value $\|\mathbf{u}_1\|^m$. We write down its KKT condition, and it is easy to see that $(\|\mathbf{u}_1\|^m, \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|})$ is a Z-eigenpair of $\mathcal{H}_V$. □

Now, we employ the proposed ACSA algorithm which works with the generating vector of a Hankel tensor to compute the largest Z-eigenvalue of the Vandermonde tensor defined in Example 3. We consider different orders $m = 4, 6, 8$ and various dimension $n = 10, \ldots, 10^6$. For each case, we choose ten random initial points, which has a Gaussian distribution on a unit sphere. Table 3 shows the computed largest Z-eigenvalues and the associated CPU times. For all case, the resulting largest Z-eigenvalue is agree with Proposition 1. When the dimension of the tensor is one million, the computational times for fourth order and sixth order Vandermonde tensors are about 35 and 55 minutes respectively.

*Example 4* An $m$th order $n$ dimensional Hilbert tensor [49] is defined as

$$\mathcal{H}_H = \frac{1}{i_1 + i_2 + \cdots + i_m - m + 1} \qquad i_j = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m.$$

Its generating vector is $\mathbf{v} = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{m(n-1)+1})^\top$. When the order $m$ is even, the Hilbert tensors are positive definite. Its largest Z-eigenvalue and largest H-eigenvalues are bounded by $n^{\frac{m}{2}} \sin \frac{\pi}{n}$ and $n^{m-1} \sin \frac{\pi}{n}$ respectively.

**Table 3** The largest Z-eigenvalues of Vandermonde tensor in Example 3

| *m* | *n* | Largest Z-eigenvalues | Occurrences | CPU times (sec.) |
|---|---|---|---|---|
| 4 | 10 | 9.487902e02 | 8 | 0.062 |
| 4 | 100 | 1.013475e05 | 8 | 0.140 |
| 4 | 1,000 | 1.019800e07 | 7 | 0.889 |
| 4 | 10,000 | 1.020431e09 | 8 | 9.048 |
| 4 | 100,000 | 1.020494e11 | 10 | 150.245 |
| 4 | 1,000,000 | 1.020500e13 | 5 | 2066.592 |
| 6 | 10 | 2.922505e04 | 5 | 0.140 |
| 6 | 100 | 3.226409e07 | 5 | 0.234 |
| 6 | 1,000 | 3.256659e10 | 7 | 1.919 |
| 6 | 10,000 | 3.259683e13 | 7 | 17.753 |
| 6 | 100,000 | 3.259985e16 | 9 | 211.537 |
| 6 | 1,000,000 | 3.260016e19 | 4 | 3190.439 |
| 8 | 10 | 9.002029e05 | 5 | 0.359 |
| 8 | 100 | 1.027131e10 | 5 | 0.437 |
| 8 | 1,000 | 1.039992e14 | 7 | 2.917 |
| 8 | 10,000 | 1.041279e18 | 7 | 30.561 |
| 8 | 100,000 | 1.041408e22 | 8 | 1058.248 |

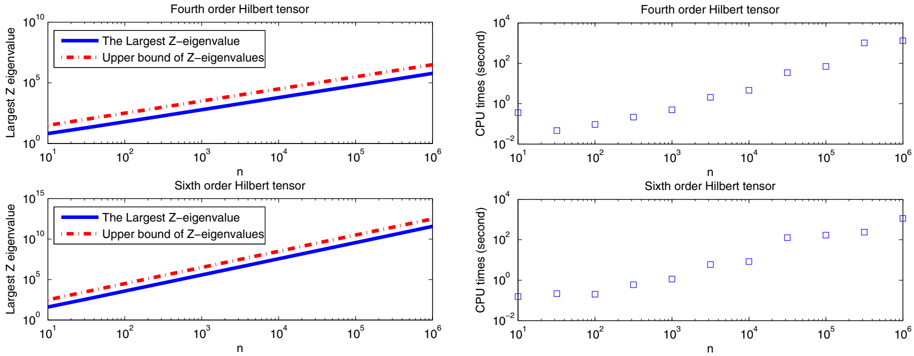


**Fig. 2** The largest Z-eigenvalue and its upper bound for Hilbert tensors

We illustrate by numerical experiments to show whether these bounds are tight? First, for the dimension varying from ten to one million, we calculate the theoretical upper bounds of the largest Z-eigenvalues of corresponding fourth order and sixth order Hilbert tensors. Then, for each Hilbert tensor, we choose ten initial points and employ the ACSA algorithm equipped with a fast computational framework for products of a Hankel tensor and vectors to compute the largest Z-eigenvalues. These results are shown in the left sub-figure of Fig. 2. The right sub-figure of Fig. 2 shows the corresponding CPU times for ACSA-Hankel. We can see that the theoretical upper bounds for the largest Z-eigenvalues of the Hilbert tensors are almost tight up to a constant multiple.

Similar results for the largest H-eigenvalues and their theoretical upper bounds of Hilbert tensors are illustrated in Fig. 3.

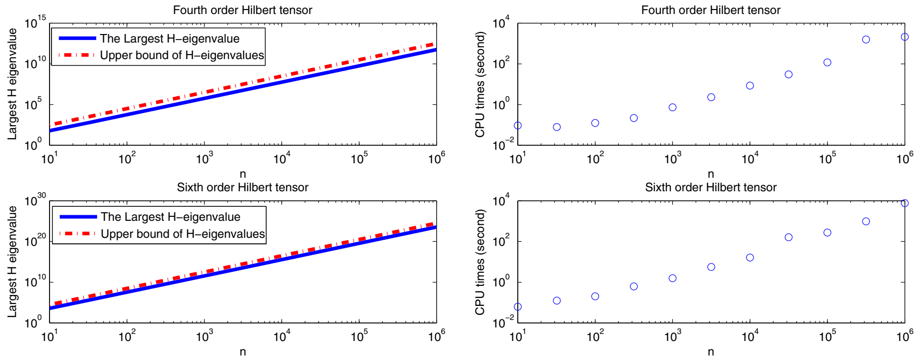


**Fig. 3** The computed largest H-eigenvalue and its upper bound for Hilbert tensors

### 5.3 An Application in Exponential Data Fitting

Exponential data fitting has important applications in signal processing and nuclear magnetic resonance [51,52]. A Hankel tensor based method was first proposed by Papy et al. [40] to process exponential data fitting. Whereafter, several approaches based on Hankel-type tensors are established and studied [6,18,41]. In this subsection, we consider a single-channel data with two peaks [40,51]

$$z(k) = \exp[(-0.01 + 2\pi\iota 0.2)k] + \exp[(-0.02 + 2\pi\iota 0.22)k], \quad k = 0, 1, \ldots, N,$$

where $\iota = \sqrt{-1}$ is an imaginary unit. An original signal $\mathbf{z} = (z(k))$ is corrupted by complex Gaussian white noise $\mathbf{e} \in \mathbb{C}^{N+1}$. The signal-noise ratio (SNR) is defined as

$$\text{SNR} = 10 \log_{10} \left( \frac{\|\mathbf{z}\|^2}{\|\mathbf{e}\|^2} \right).$$

Hence, the observed signal $\mathbf{y} = (y(k))$ is

$$y(k) = z(k) + e_{k-1}, \qquad k = 0, 1, \ldots, N.$$

In this way, we could obtain an $m$th order $n$ dimensional Hankel tensor $\mathcal{H}_E$ whose generating vector is $\mathbf{v}_E = |\mathbf{y}|$ if $N = m(n-1)$.

Now, we study the largest H- and Z-eigenvalues of $\mathcal{H}_E$ versus SNR. For instance, we consider fourth order 1000 dimensional Hankel tensors with SNR $= 10, 15, 20, 25, 30$. In each SNR, one hundred noise-corrupted Hankel tensors are generated. For each Hankel tensor, we start ACSA from ten random initial points chosen uniformly on a unit sphere. The largest one of resulting ten eigenvalues is regarded as the largest eigenvalue of this Hankel tensor. In Fig. 4, we illustrated the mean value and the standard error of the largest H- and Z-eigenvalues of one hundred noise-corrupted Hankel tensors for each SNR. A red bar stands for a mean value and a blue segment is two standard errors. Obviously, as SNR decreases, mean values and standard errors of the largest H- and Z-eigenvalues of noise-corrupted Hankel tensors increase.

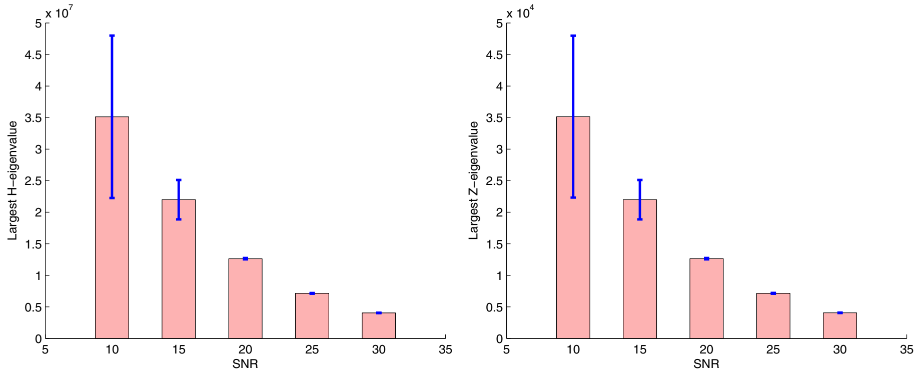


**Fig. 4** The largest H- and Z-eigenvalues of Hankel tensors arising from exponential data fitting

### 5.4 Initial Step Sizes

In the process of the curvilinear search, how to determine a suitable step size is a critical problem. Barzilai and Borwein [4] provided two candidates

$$\bar{\alpha}_{k+1}^{\mathrm{BB-I}} := \frac{\Delta \mathbf{x}_k^\top \Delta \mathbf{g}_k}{\|\Delta \mathbf{g}_k\|^2} \qquad \text{and} \qquad \bar{\alpha}_{k+1}^{\mathrm{BB-II}} := \frac{\|\Delta \mathbf{x}_k\|^2}{\Delta \mathbf{x}_k^\top \Delta \mathbf{g}_k},$$

which satisfy the quasi-Newton condition approximately. However, when the optimization problem is nonconvex, the inner product $\Delta \mathbf{x}_k^\top \Delta \mathbf{g}_k$ maybe zero or negative, which could destroy the curvilinear search. Dai [15] proposed to use their geometric mean.

Next, we compare four sorts of strategies for the initial step size of curvilinear search: (1) Dai's step size (12), (2)–(3) absolute values of $\bar{\alpha}_{k+1}^{\mathrm{BB-I}}$ and $\bar{\alpha}_{k+1}^{\mathrm{BB-II}}$, (4) a fixed step size $\bar{\alpha}_{k+1}^{\mathrm{One}} = 1$. Using these strategies, we compute the largest Z-eigenvalue of a fourth order $10,000$ dimensional Hilbert tensor. All of the four approaches start from the same ten initial points and reach the same Z-eigenvector. Figure 5 illustrates counting results of the curvilinear

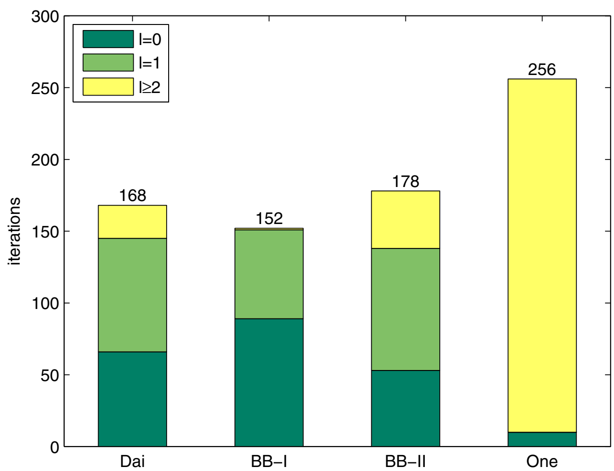


**Fig. 5** Comparisons of four sorts of step size strategies

search parameter $\ell$. Obviously, the fixed step size one performs poorly since $\ell$ is always great than or equal to 2. By exploiting the quasi-Newton condition approximately, BB-I and BB-II perform satisfactory, where BB-I seems better. The performance of Dai's step size is in the medium place of BB-I and BB-II. It only requires $\ell = 0.78$ times curvilinear search per iteration on average. We employ Dai's step size since it is positive and hence safe in theory.

## 6 Conclusion

We proposed an inexact steepest descent method processing on a unit sphere for generalized eigenvalues and associated eigenvectors of Hankel tensors. Owing to the fast computation framework for the products of a Hankel tensor and vectors, the new algorithm is fast and efficient as shown by some preliminary numerical experiments. Since the Hankel structure is well-exploited, the new method could deal with some large scale Hankel tensors, whose dimension is up to one million in a desktop computer.

**Acknowledgments** We thank Mr. Weiyang Ding and Dr. Ziyan Luo for the discussion on numerical experiments, and two referees for their valuable comments.

## References

1. Attouch, H., Bolte, J.: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. Math. Program. Ser. B **116**, 5–16 (2009)
2. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka–Łojasiewicz inequality. Math. Oper. Res. **35**, 438–457 (2010)
3. Bader, B., Kolda, T.: Efficient MATLAB computations with sparse and factored tensors. SIAM J. Sci. Comput. **30**, 205–231 (2007)
4. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**, 141–148 (1988)
5. Bolte, J., Daniilidis, A., Lewis, A.: The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM J. Optim. **17**, 1205–1223 (2006)
6. Boyer, R., De Lathauwer, L., Abed-Meraim, K.: Higher order tensor-based method for delayed exponential fitting. IEEE Trans. Signal Process. **55**, 2795–2809 (2007)
7. Chang, K.C., Pearson, K., Zhang, T.: On eigenvalue problems of real symmetric tensors. J. Math. Anal. Appl. **350**, 416–422 (2009)
8. Chen, L., Han, L., Zhou, L.: Computing tensor eigenvalues via homotopy methods, (2015). arXiv:1501.04201v3
9. Chen, Y., Dai, Y., Han, D., Sun, W.: Positive semidefinite generalized diffusion tensor imaging via quadratic semidefinite programming. SIAM J. Imaging Sci. **6**, 1531–1552 (2013)
10. Chen, Y., Qi, L., Wang, Q.: Positive semi-definiteness and sum-of-squares property of fourth order four dimensional Hankel tensors, (2015). arXiv:1502.04566v8
11. Choi, J.H., Vishwanathan, S.V.N.: DFacTo: distributed factorization of tensors, (2014). arXiv:1406.4519v1
12. Cichocki, A., Phan, A.-H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Trans. Fund. Electron. **E92–A**, 708–721 (2009)
13. Cooper, J., Dutle, A.: Spectra of uniform hypergraphs. Linear Algebra Appl. **436**, 3268–3292 (2012)
14. Cui, C., Dai, Y., Nie, J.: All real eigenvalues of symmetric tensors. SIAM J. Matrix Anal. Appl. **35**, 1582–1601 (2014)
15. Dai, Y.: A positive BB-like stepsize and an extension for symmetric linear systems, In: Workshop on Optimization for Modern Computation, Beijing, China, (2014), http://bicmr.pku.edu.cn/conference/opt-2014/slides/Yuhong-Dai
16. de Almeida, A.L.F., Kibangou, A.Y.: Distributed large-scale tensor decomposition, In: IEEE International Conference on Acoustics, Speech and Siganl Processing (ICASSP) (2014) 26–30

17. De Lathauwer, L., De Moor, B., Vandewalle, J.: On the best rank-1 and rank-($R_1, R_2, \ldots, R_N$) approximation of higher-order tensors. SIAM J. Matrix Anal. Appl. **21**, 1324–1342 (2000)
18. Ding, W., Qi, L., Wei, Y.: Fast Hankel tensor-vector product and its application to exponential data fitting. Numer. Linear Algebra Appl. **22**, 814–832 (2015)
19. Ding, W., Wei, Y.: Generalized tensor eigenvalue problems. SIAM J. Matrix Anal. Appl. **36**, 1073–1099 (2015)
20. Friedland, S., Nocedal, J., Overton, M.L.: The formulation and analysis of numerical methods for inverse eigenvalue problems. SIAM J. Numer. Anal. **24**, 634–667 (1987)
21. Goldfarb, D., Wen, Z., Yin, W.: A curvilinear search method for the $p$-harmonic flow on spheres. SIAM J. Imaging Sci. **2**, 84–109 (2009)
22. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. The Johns Hopkins University Press, Baltimore (2013). ISBN 978-1-4214-0794-4
23. Han, L.: An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors. Numer. Algebra Control Optim. **3**, 583–599 (2013)
24. 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)
25. Hao, C., Cui, C., Dai, Y.: A feasible trust-region method for calculating extreme Z-eigenvalues of symmetric tensors, Pacific J. Optim. **11**, 291–307 (2015)
26. Hillar, C.J. , Lim, L.-H.: Most tensor problems are NP-hard, J. ACM **60** (2013) article 45:1–39
27. Hu, S., Huang, Z., Qi, L.: Finding the extreme Z-eigenvalues of tensors via a sequential SDPs method. Numer. Linear Algebra Appl. **20**, 972–984 (2013)
28. Kang, U., Papalexakis, E., Harpale, A., Faloutsos, C.: GigaTensor: scaling tensor analysis up by 100 times—algorithms and discoveries, In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 316–324 (2012)
29. Kofidis, E., Regalia, P.A.: On the best rank-1 approximation of higher-order supersymmetric tensors. SIAM J. Matrix Anal. Appl. **23**, 863–884 (2002)
30. Kolda, T.G., Mayo, J.R.: Shifted power method for computing tensor eigenpairs. SIAM J. Matrix Anal. Appl. **32**, 1095–1124 (2011)
31. Kolda, T.G., Mayo, J.R.: An adaptive shifted power method for computing generalized tensor eigenpairs. SIAM J. Matrix Anal. Appl. **35**, 1563–1581 (2014)
32. Lim, L.-H.: Singular values and eigenvalues of tensors: a variational approach, In: Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP'05), **1**: 129–132 (2005)
33. Łojasiewicz, S.: Une propriété topologique des sous-ensembles analytiques réels, Les Équations aux Dérivées Partielles, Éditions du centre National de la Recherche Scientifique, Paris, 87–89 (1963)
34. Luque, J.-G., Thibon, J.-Y.: Hankel hyperdeterminants and Selberg integrals. J. Phys. A **36**, 5267–5292 (2003)
35. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text, In: Proceeding of the 7th ACM Conference on Recommender Systems, 165–172 (2013)
36. Ni, G., Qi, L., Bai, M.: Geometric measure of entanglement and U-eigenvalues of tensors. SIAM J. Matrix Anal. Appl. **35**, 73–87 (2014)
37. Ni, Q., Qi, L., Wang, F.: An eigenvalue method for testing positive definiteness of a multivariate form. IEEE Trans. Automat. Control **53**, 1096–1107 (2008)
38. Nie, J., Wang, L.: Semidefinite relaxations for best rank-1 tensor approximations. SIAM J. Matrix Anal. Appl. **35**, 1155–1179 (2014)
39. Oropeza, V., Sacchi, M.: Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis. Geophysics **76**, V25–V32 (2011)
40. Papy, J.M., De Lathauwer, L., Van Huffel, S.: Exponential data fitting using multilinear algebra: the single-channel and multi-channel case. Numer. Linear Algebra Appl. **12**, 809–826 (2005)
41. Papy, J.M., De Lathauwer, L., Van Huffel, S.: Exponential data fitting using multilinear algebra: the decimative case. J. Chemom. **23**, 341–351 (2009)
42. Qi, L.: Eigenvalues of a real supersymmetric tensor. J. Symb. Comput. **40**, 1302–1324 (2005)
43. Qi, L.: Hankel tensors: associated Hankel matrices and Vandermonde decomposition. Commun. Math. Sci. **13**, 113–125 (2015)
44. Qi, L., Wang, F., Wang, Y.: Z-eigenvalue methods for a global polynomial optimization problem. Math. Program. Ser. A **118**, 301–316 (2009)
45. Qi, L., Yu, G., Xu, Y.: Nonnegative diffusion orientation distribution function. J. Math. Imaging Vis. **45**, 103–113 (2013)
46. Schatz, M.D., Low, T.-M., Van De Geijn, R.A., Kolda, T.G.: Exploiting symmetry in tensors for high performance. SIAM J. Sci. Comput. **36**, C453–C479 (2014)

47. Schultz, T., Seidel, H.-P.: Estimating crossing fibers: a tensor decomposition approach. IEEE Trans. Vis. Comput. Gr. **14**, 1635–1642 (2008)
48. Smith, R.S.: Frequency domain subspace identification using nuclear norm minimization and Hankel matrix realizations. IEEE Trans. Automat. Control **59**, 2886–2896 (2014)
49. Song, Y., Qi, L.: Infinite and finite dimensional Hilbert tensors. Linear Algebra Appl. **451**, 1–14 (2014)
50. Trickett, S., Burroughs, L., Milton, A.: Interpolating using Hankel tensor completion, In: SEG Annual Meeting, 3634–3638 (2013)
51. Van Huffel, S.: Enhanced resolution based on minimum variance estimation and exponential data modeling. Signal Process. **33**, 333–355 (1993)
52. Van Huffel, S., Chen, H., Decanniere, C., Van Hecke, P.: Algorithm for time-domain NMR data fitting based on total least squares. J. Magn. Reson. Ser. A **110**, 228–237 (1994)
53. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Math. Program. Ser. A **142**, 397–434 (2013)
54. Xu, C.: Hankel tensors, Vandermonde tensors and their positivities, Linear Algebra Appl., in press (2015)