Numerical Algorithms
https://doi.org/10.1007/s11075-023-01593-y

ORIGINAL PAPER

A family of gradient methods using Householder transformation with application to hypergraph partitioning

Xin Zhang [1] · Jingya Chang[2] · Zhili Ge[3] · Zhou Sheng[4,5]

Received: 19 January 2023 / Accepted: 29 May 2023

Abstract
In this paper, we propose a constraint preserving algorithm for the smallest Z-eigenpair of the compact Laplacian tensor of an even-uniform hypergraph, where Householder transform is employed and a family of modified conjugate directions with sufficient descent is determined. Besides, we prove that there exists a positive step size in the new constraint preserving update scheme such that the Wolfe conditions hold. Based on these properties, we prove the convergence of the new algorithm. Furthermore, we apply our algorithm to the hypergraph partitioning and image segmentation, and numerical results are reported to illustrate the efficiency of the proposed algorithm.

Keywords Householder transform · Tensor eigenvalue · Conjugate gradient · Hypergraph partitioning · Image segmentation

✉ Zhou Sheng
szhou03@live.com; shengz@ahut.edu.cn

Xin Zhang
zhangxin0619@126.com

Jingya Chang
jychang@gdut.edu.cn

Zhili Ge
gezhilyly66@126.com

1 School of Sciences and Arts, Suqian University, Suqian 223800, China

2 School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou 510006, China

3 School of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing 210038, China

4 Department of Data Science, Anhui University of Technology, Maanshan 243002, China

5 Anhui Provincial Joint Key Laboratory of Disciplines for Industrial Big Data Analysis and Intelligent Decision, Maanshan 243032, China

1 Introduction

Clustering is an important method in machine learning, and it has applications in many fields [53], such as data mining [30], data reduction [33, 36, 37] image segmentation [38, 39], and object recognition [19]. Spectral clustering, which partitions objects based on spectral graph theory, is an important branch of partitional clustering approaches [13]. Compared with other partitional clustering algorithms, such as k-means and k-medoids, spectral clustering methods are not only robust and stable, but also easy to implement [6]. In fact, for graph spectral clustering problems, the objects are assumed to have pairwise correlations [11, 12]. However, the objects often have multiway affinities in many applications [25, 40]. Govindu [27] and Chen et al. [9] gave such an example shown in Fig. 1 that indicates the necessity of multiwise relationship. If we want to separate the points on two intersecting circles in Fig. 1, the distance between two points is meaningless [9, 27].

Thereby, tensor-based hypergraph spectral clustering approaches attract more and more attention. The proposal of the concept of tensor eigenvalue [45, 51] and the development of hypergraph theory [3, 13, 20, 35, 52, 55, 56] make it possible.

Bulò and Pelillo [3] first studied the spectral hypergraph theory via tensors. Before that, works on spectral hypergraph theory were via Laplacian matrices. Cooper and Dutle [13] proposed the concept of adjacency tensor for a uniform hypergraph and proved lots of natural analogs of basic results in spectral graph theory. Hu and Qi [34] first defined a Laplacian tensor for an even-uniform hypergraph. They proved that the smallest Z-eigenvalue of the Laplacian tensor is zero. Furthermore, the hypergraph is connected if and only if the second smallest Z-eigenvalue of the Laplacian tensor is positive. Hence, the second smallest Z-eigenvalue is called the algebraic connectivity of the hypergraph. Li et al. [44] and Xie and Chang [54] gave two variations of this kind of Laplacian tensors for an even-uniform hypergraph. Qi [52] proposed another definition of a Laplacian tensor for odd- and even-uniform hypergraphs, which is as

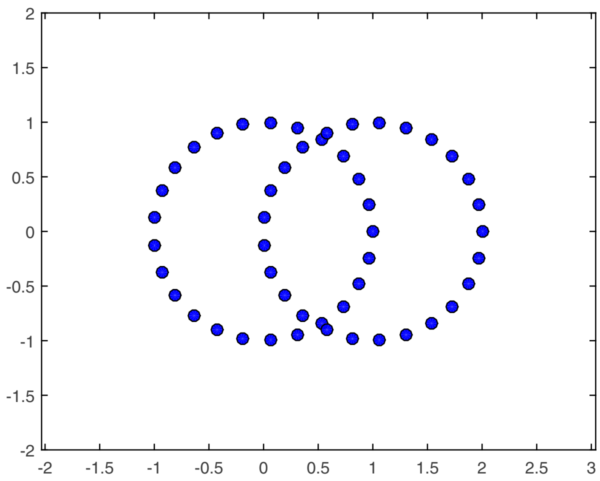


Fig. 1 Two intersecting circles

simple as their matrix counterparts in formalism. He also verified that the smallest H-eigenvalue of Qi's Laplacian tensor is zero. Bu et al. [2] argued that zero is also a Z-eigenvalue of Qi's Laplacian tensor. A normalized version of Qi's Laplacian tensor for a uniform hypergraph was considered in [35]. Chen and Qi [10] studied directed hypergraphs and defined the adjacency tensor, the Laplacian tensor, and the signless Laplacian tensor for a uniform directed hypergraph. Xie and Qi [56] studied spectral properties of the adjacency tensor, the Laplacian tensor, and the signless Laplacian tensor of a uniform directed hypergraph. Chen et al. [9] generalized Fiedler vector from an ordinary graph [21, 22] to an even-uniform hypergraph and extended the normalized Laplacian matrix of an ordinary graph to the normalized Laplacian tensor of an even-uniform hypergraph. Compared with the Laplacian tensor proposed by Hu and Qi [34], the normalized Laplacian tensor defined by Chen et al. [9] is favorable for partitioning even-uniform hypergraphs. Chen et al. [9] developed a novel tensor-based spectral method to partition vertices of the hypergraph and transformed the hypergraph partitioning problem into the smallest Z-eigenvalue problem of a hypergraph related tensor, which is constrained on a unit sphere. Numerical experiments illustrated that the new approach is effective and promising for some combinatorial optimization problems arising from subspace partitioning and face clustering.

The hypergraph partitioning problem is a challenging work, often involving large-scale computational tasks. The conjugate gradient (CG) method is well-suited for large-scale optimization problems, and there has been extensive research in this area. There are several classical CG methods, e.g., *FR-CG* [24], *PRP-CG* [49, 50], *HS-CG* [32], *CD-CG* [23], *LS-CG* [46], and *DY-CG* [16]. Before that, Hager and Zhang [28] proposed a new CG method with guaranteed descent, no matter what line search method is adopted. They also summarized different versions of nonlinear conjugate gradient methods in [29], and the global convergence result is established when the line search fulfills the Wolfe conditions [28, 29]. Recently, Yuan et al. [60] proposed a family of weak conjugate gradient formulae based on the classical formulae which has better theory properties. Abdollahi et al. [1] proposed a modified CG method to solve large-scale non-smooth problems. Additionally, many efforts have been made to modify the parameters of the CG method in order to improve their performance. We refer to [1, 60] and the references therein.

In this paper, we give a new algorithm for solving the Fiedler vector of an even-uniform hypergraph based on the normalized Laplacian tensor proposed in [9], where the smallest Z-eigenvalue and the corresponding eigenvector are computed. Since the smallest Z-eigenvalue problem is defined on a unit sphere, we apply the Householder transform to preserve the iterates on the sphere and obtain a new constraint preserving update scheme. Besides, we give a family of conjugate gradient directions with sufficient descent conditions, which are independent of the line search strategy. We also prove that there exists a positive step size in the new update scheme such that the Wolfe conditions hold. Based on these properties, we prove the convergence of the new algorithm and apply our algorithm to the hypergraph partitioning and image segmentation.

The outline of this paper is drawn as follows. In Sect. 2, we review some preliminaries about tensor eigenvalues and weighted hypergraphs. In Sect. 3, we give a constraint preserving algorithm for the smallest Z-eigenpair, where Householder transform is employed. We analyze the convergence of the proposed algorithm in Sect. 4. Section 5 reports the numerical results on hypergraph partitioning and image segmentation. Section 6 concludes this paper.

2 Some preliminaries

In this section, we first review some definitions of tensor, tensor eigenvalue, and the corresponding eigenvector, then introduce the concepts of weighted hypergraphs.

Let $\mathbb{R}$ be the real field. The kth order n-dimensional tensor consists of n^k entries in $\mathbb{R}$:

$$\mathcal{T} = (t_{i_1 i_2 \ldots i_k}), \quad t_{i_1 i_2 \ldots i_k} \in \mathbb{R}, \quad 1 \le i_1, i_2, \ldots, i_k \le n.$$

We denote the set of all kth order n-dimensional tensors by $\mathbb{R}^{[k,n]}$ throughout the rest of this paper. A tensor $\mathcal{T} \in \mathbb{R}^{[k,n]}$ is called symmetric [41] if its entries are invariant under any permutation of their indices, i.e.,

$$t_{\sigma(i_1 i_2 \ldots i_k)} = t_{i_1 i_2 \ldots i_k},$$

where $\sigma(i_1 i_2 \ldots i_k)$ is any permutation of $i_1 i_2 \ldots i_k$, $1 \le i_1 \le \cdots \le i_k \le n$. A tensor $\mathcal{T} \in \mathbb{R}^{[k,n]}$ is called semi-symmetric [48] if its entries are invariant under any permutation of their backward $k-1$ indices, i.e.,

$$t_{i\sigma(i_2 \ldots i_k)} = t_{i i_2 \ldots i_k},$$

where $1 \le i \le n$ and $\sigma(i_2 \ldots i_k)$ is any permutation of $i_2 \ldots i_k$, $1 \le i_2 \le \cdots \le i_k \le n$.

For any $\mathcal{T} \in \mathbb{R}^{[k,n]}$, there exists a homogeneous polynomial with degree k corresponding to it, which is defined by

$$\mathcal{T}\mathbf{x}^k = \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_k=1}^{n} t_{i_1 i_2 \ldots i_k} x_{i_1} x_{i_2} \cdots x_{i_k}.$$

Also, it corresponds a homogeneous polynomial map $\mathcal{T}\mathbf{x}^{k-1}$ with degree $k-1$, where

$$(\mathcal{T}\mathbf{x}^{k-1})_i = \sum_{i_2=1}^{n} \cdots \sum_{i_k=1}^{n} t_{i i_2 \ldots i_k} x_{i_2} \cdots x_{i_k}, \quad \text{for } i = 1, 2, \ldots, n. \tag{1}$$

Note that for a homogeneous polynomial, the associated tensor is not unique, but if restricted to symmetric tensor, it is unique. That is, there is a one-to-one correspondence between a symmetric tensor and a homogeneous polynomial.

Similarly, there is a one-to-one correspondence between a semi-symmetric tensor and a homogeneous polynomial map [48]. This means for any $\mathcal{T} \in \mathbb{R}^{[k,n]}$, there exists a unique semi-symmetric tensor $\mathcal{T}_s$, such that

$$\mathcal{T}\mathbf{x}^{k-1} = \mathcal{T}_s\mathbf{x}^{k-1}, \quad \text{for all } \mathbf{x} \in \mathbb{R}^n.$$

Ni and Qi [48] called $\mathcal{T}_s$ the associated semi-symmetric tensor of $\mathcal{T}$. Zhang et al. [62] proved that $\mathcal{T}\mathbf{x}^{k-1}$ is identical to $\mathbf{0}$ for all $\mathbf{x} \in \mathbb{R}^n$ if and only if $\mathcal{T}_s$ is a zero tensor.

The definition of tensor eigenvalues and eigenvectors were proposed by Qi [51] and Lim [45] in 2005 independently.

Definition 1 Let $\mathcal{T} \in \mathbb{R}^{[k,n]}$ and $\mathbb{C}$ be the complex field. Assume that $\mathcal{T}_s$ is not a zero tensor, where $\mathcal{T}_s$ is the associated semi-symmetric tenor of $\mathcal{T}$. We say $(\lambda, \mathbf{x}) \in \mathbb{C} \times \{\mathbb{C}^n \backslash \{0\}\}$ is an E-eigenpair of $\mathcal{T}$ if they satisfy the equations

$$\mathcal{T}\mathbf{x}^{k-1} = \lambda\mathbf{x} \text{ and } \mathbf{x}^T\mathbf{x} = 1, \tag{2}$$

where $\mathcal{T}\mathbf{x}^{k-1}$ is defined by (1). If $\mathbf{x}$ is a real eigenvector, then we call λ a Z-eigenvalue and call $\mathbf{x}$ the corresponding Z-eigenvector of $\mathcal{T}$.

When partitioning a hypergraph, we classify the vertices according to the relevance among the vertices. The relevance can be reflected by the weight of the edges in the hypergraph. Therefore, we first give the notion of undirected weighted hypergraphs or simply called weighted hypergraphs if there is no confusion.

Suppose $G = (V, E, \mathbf{w})$ is a k-uniform weighted hypergraph, where $V = \{1, 2, \ldots, n\}$ is the vertex set, $E = \{e_p \subseteq V : |e_p| = k, p = 1, 2, \ldots, m\}$ is the edge set, here $|\cdot|$ means the cardinality of a set, $\mathbf{w} = [w_p] \in \mathbb{R}^m$ is a positive vector whose component w_p denotes the weight of an edge $e_p \in E$. If k is even, G is called an even-uniform hypergraph. Especially, if $k = 2$, G is an ordinary graph. For each vertex $i \in V$, its degree is $d_i = \sum_{e_p \in E: i \in e_p} w_p$. If a hypergraph has no isolated vertices, then $\mathbf{d} = [d_i] \in \mathbb{R}^n$ is positive.

Two vertices i and j are called connected if there is a finite sequence of vertices $\{i, l_1, l_2, \ldots, l_t, j\}$ such that every two adjacent vertices belong to one edge of G. If any two vertices in G are connected to each other, G is called a connected hypergraph. A connected component of a hypergraph G is a connected subhypergraph which is not contained in any connected subhypergraph of G with more vertices or edges.

An intuitive interpretation of hypergraph partitioning is to divide the vertices with strong correlation into the same set and divide the vertices with weak correlation into different sets. Therefore, hypergraph partitioning needs to cut some edges of the original hypergraph, and this segmentation brings out a certain cost. The goal of the partitioning is to minimize the cost under some certain constraints. Different cost functions often correspond to different Laplacian tensors, resulting in different optimization models.

Suppose $G = (V, E, \mathbf{w})$ is a k-uniform weighted hypergraph with n vertices, $\mathbf{d} = [d_i] \in \mathbb{R}^n$ is a degree vector, where $d_i = \sum_{e_p \in E: i \in e_p} w_p$. Consider the bipartitioning problem of a hypergraph [9], and suppose the vertex set V is partitioned into two parts, denoted by X and $\overline{X} \equiv V \backslash X$. Assigning two different labels (e.g., ± 1) to the partition $(X, \overline{X})$, then an indicator vector $\mathbf{x} = [x_i] \in \mathbb{R}^n$ is obtained, where

$$x_i = \begin{cases} 1, & i \in X, \\ -1, & i \in \overline{X}. \end{cases}$$

Hu and Qi [35] proposed a kind of Laplacian tensor, and the corresponding cost of a partition $(X, \overline{X})$ is

$$\sum_{e_p \in E} w_p \left(\sum_{i \in e_p} \frac{x_i^k}{d_i} - k \prod_{j \in e_p} \frac{x_j}{\sqrt[k]{d_j}} \right). \tag{3}$$

Chen et al. [9] argued that (3) is an insufficient cost function for evaluating a partition of G, and gave an example to illustrate why it is insufficient. Meanwhile, they defined another kind of cost function, as follows:

$$\mathrm{cost}(\mathbf{x}) = \sum_{e_p \in E} \tau w_p \sum_{i \in e_p} \left(\frac{x_i}{\sqrt[k]{d_i}} - \frac{1}{k} \sum_{j \in e_p} \frac{x_j}{\sqrt[k]{d_j}} \right)^k , \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^n$ is an indicator vector, $\tau = \frac{k^k}{(k-1)^k + k - 1}$. It is noted that the cost function (4) is a homogeneous polynomial, and it determines a unique symmetric tensor $\widetilde{\mathcal{L}} \in \mathbb{R}^{[k,n]}$, such that $\mathrm{cost}(\mathbf{x}) = \widetilde{\mathcal{L}}\mathbf{x}^k$. They [9] called $\widetilde{\mathcal{L}}$ the normalized Laplacian tensor, and gave a symmetric decomposition of $\widetilde{\mathcal{L}}$ (see Lemma 2.1 in [9]).

Based on the relationship between the number of connected components of G and the number of linearly independent eigenvectors corresponding to the smallest Z-eigenvalue of $\widetilde{\mathcal{L}}$, Chen et al. [9] gave the definition of the Fiedler vector of a hypergraph, as follows:

Definition 2 ([9] Definition 2.4) Suppose $G = (V, E, \mathbf{w})$ is an even-uniform hypergraph and $\widetilde{\mathcal{L}}$ is its normalized Laplacian tensor. Let all the Z-eigenvalues of $\widetilde{\mathcal{L}}$ be ordered as

$$0 = \lambda_0 \leq \lambda_1 \ldots ,$$

Then, λ_1 is called the algebraic connectivity of G, denoted as $\lambda_1(G)$. Moreover, the Z-eigenvector of $\widetilde{\mathcal{L}}$ corresponding to λ_1 is called the Fiedler vector of G.

Chen et al. [9] proved that the Fiedler vector of the hypergraph G is the optimal solution of the following problem.

Theorem 1 *([9] Theorem 2.6) Suppose $G = (V, E, \mathbf{w})$ is an even-uniform weighted hypergraph and $\widetilde{\mathcal{L}}$ is its normalized Laplacian tensor. Then, the algebraic connectivity of G could be characterized as*

$$\lambda_1(G) = \begin{cases} \min \ \widetilde{\mathcal{L}}\mathbf{x}^k \\ s.t. \ \ \mathbf{x}^T\mathbf{x} = 1, \ \ \mathbf{x}^T \quad = 0, \end{cases} \tag{5}$$

where $\; = [\sqrt[k]{d_i}] \in \mathbb{R}^n$. Moreover, the optimal solution of (5) is the Fiedler vector of G.

The second smallest Z-eigenvalue of $\widetilde{\mathcal{L}}$ needs to be solved in (5). Since the hypergraph partitioning problem is often large scale, the existing algorithms [7, 14] for solving the second smallest eigenvalue are more suitable for small tensors. To obtain the Fiedler vector of an even-uniform hypergraph more efficiently, Chen et al. [9] proposed the definition of a compact Laplacian tensor, whose smallest Z-eigenvalue is the algebraic connectivity of the even-uniform hypergraph.

Suppose $G = (V, E, \mathbf{w})$ is a k-uniform weighted hypergraph with n vertices and $\widetilde{\mathcal{L}}$ is its normalized Laplacian tensor, $\mathbf{d} = [d_i] \in \mathbb{R}^n$ is its degree vector. Let $Q \in \mathbb{R}^{n\times(n-1)}$ and satisfy $Q^T Q = I_{n-1}$, $Q^T \ = \mathbf{0}$, where $\ = [\ \sqrt[k]{d_i}\ ] \in \mathbb{R}^n$. The compact Laplacian tensor of G is defined as $\mathcal{L}_C = \widetilde{\mathcal{L}} Q^k$, where

$$
\begin{aligned}
[\mathcal{L}_C]_{j_1 j_2 \ldots j_k} = [\widetilde{\mathcal{L}} Q^k]_{j_1 j_2 \ldots j_k} &= \sum_{i_1, i_2, \ldots, i_k = 1}^{n} \widetilde{\mathcal{L}}_{i_1 i_2 \ldots i_k} q_{i_1 j_1} q_{i_2 j_2} \cdots q_{i_k j_k} \\
&= \tau \sum_{e_p \in E_w} w_p \sum_{i \in e_p} [Q^T \mathbf{u}_{p,i}]_{j_1} [Q^T \mathbf{u}_{p,i}]_{j_2} \cdots [Q^T \mathbf{u}_{p,i}]_{j_k},
\end{aligned} \tag{6}
$$

where $j_s = 1, 2, \ldots, n-1$, $s = 1, 2, \ldots k$, $\tau = \frac{k^k}{(k-1)^k + k - 1}$, $\mathbf{u}_{p,i} = \frac{1}{\sqrt[k]{d_i}} \mathbf{e}_i - \frac{1}{k} \sum_{j \in e_p} \frac{1}{\sqrt[k]{d_j}} \mathbf{e}_j$ only if $i \in e_p$, $\mathbf{e}_i$ being the ith column of an identity matrix.

With the concept of the compact Laplacian tensor, the algebraic connectivity of G can be solved by minimizing the following Z-eigenvalue problem.

Theorem 2 *([9] Theorem 3.1) Suppose $G = (V, E, \mathbf{w})$ is an even-uniform hypergraph and $\mathcal{L}_C$ is its compact Laplacian tensor, Then, the algebraic connectivity $\lambda_1(G)$ is the smallest Z-eigenvalue of $\mathcal{L}_C$, i.e.,*

$$
\lambda_1(G) = \begin{cases} \min \ \ f(\mathbf{y}) = \frac{\mathcal{L}_C \mathbf{y}^k}{\|\mathbf{y}\|^k} \\ s.t. \ \ \mathbf{y}^T \mathbf{y} = 1. \end{cases} \tag{7}
$$

Let $\mathbf{y}^* \in \mathbb{R}^{n-1}$ be the optimal solution of the spherical optimization (7). Then, the Fiedler vector of G is $Q\mathbf{y}^* \in \mathbb{R}^n$.

Chen et al. [9] established a feasible optimization algorithm to compute the Fiedler vector according to the compact Laplacian tensor and the normalized Laplacian tensor. Numerical experiments in [9] illustrate that the new approach based on Fiedler vector of a hypergraph is effective and promising.

Based on these conclusions, we propose a constraint preserving algorithm for solving (7), get the smallest Z-eigenpair of the compact Laplacian tensor, and further apply it to the hypergraph partitioning and image segmentation problems.

3 Constraint preserving algorithm for the smallest *Z*-eigenpair

In this section, we solve the problem (7) to obtain the smallest Z-eigenpair of the compact Laplacian tensor $\mathcal{L}_C$, where the Householder transform is employed to preserve the spherical constraint.

Suppose $G = (V, E, \mathbf{w})$ is a k (k is even)-uniform weighted hypergraph with n vertices and m edges, $\mathcal{L}_C \in \mathbb{R}^{[k, n-1]}$ is the compact Laplacian tensor of G. Denote the spherical constraint of (7) by

$$\mathbb{S}^{n-2} \triangleq \{\mathbf{y} \in \mathbb{R}^{n-1} : \mathbf{y}^T\mathbf{y} = 1\}.$$

The gradient of the objective function $f(\mathbf{y}) = \frac{\mathcal{L}_C\mathbf{y}^k}{\|\mathbf{y}\|^k}$ at $\mathbf{y} \in \mathbb{R}^{n-1}$ [8] is

$$\mathbf{g}(\mathbf{y}) = \frac{k}{\|\mathbf{y}\|^{k+2}}(\|\mathbf{y}\|^2 \cdot \mathcal{L}_C\mathbf{y}^{k-1} - \mathcal{L}_C\mathbf{y}^k \cdot \mathbf{y}). \tag{8}$$

It is easy to verify that for any $\mathbf{y} \in \mathbb{R}^{n-1}$,

$$\mathbf{y}^T\mathbf{g}(\mathbf{y}) = 0 \tag{9}$$

is always true, this means $\mathbf{g}(\mathbf{y})$ is always on the tangent plane of $\mathbf{y}$.

For the spherical constraint in the Z-eigenvalue problem (7), there are several different treatments. The most common way is to directly normalize the iterates, such as the symmetric higher-order power method (S-HOPM) [18], the shifted symmetric higher-order power method (SS-HOPM) [42], and the generalized eigenproblem adaptive power (GEAP) method [43]. Yu et al. [57] and Hao et al. [31] adopt the following iterative scheme: $\mathbf{y}(\alpha) = \sqrt{1 - \alpha^2\|\mathbf{g}_c\|^2}\mathbf{y}_c + \alpha\mathbf{g}_c$ to ensure the new iterate is still on the sphere, where $\mathbf{y}_c$ is the current iterate, $\mathbf{g}_c$ is the gradient at $\mathbf{y}_c$, $0 < \alpha \leq 1/\|\mathbf{g}_c\|$. Another common method is to employ the Cayley transform to preserve iterates on the sphere [4, 5, 8, 9]. Cayley transformation matrix $C = (I + W)^{-1}(I - W)$ is an orthogonal matrix, where W is a skew-symmetric matrix. Inspired by this, we try to employ another orthogonal transformation—Householder transform, to guarantee the spherical constraint in the Z-eigenvalue problem (7).

Construct an $n - 1$ Householder matrix:

$$H = I_{n-1} - \frac{2\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}, \tag{10}$$

where $\mathbf{v} \in \mathbb{R}^{n-1}$. Suppose that $\mathbf{y}_c$ is the current iterate and satisfies $\mathbf{y}_c^T\mathbf{y}_c = 1$, $\mathbf{y}_{c+1}$ represents the next iterate. Let

$$\mathbf{y}_{c+1} = -H\mathbf{y}_c, \tag{11}$$

then $\mathbf{y}_{c+1}^T\mathbf{y}_{c+1} = 1$. The new iterate $\mathbf{y}_{c+1}$ is closely related to the choice of the vector $\mathbf{v}$. If $\mathbf{v}$ is an arbitrary vector in $\mathbb{R}^{n-1}$, $\mathbf{y}_{c+1}$ may deviate far from $\mathbf{y}_c$, and consequently, the iterative sequence may not be convergent. Therefore, we choose the current iterate $\mathbf{y}_c$ and a current descent direction to construct $\mathbf{v}$.

Suppose that $\mathbf{v} = \mathbf{y}_c + \alpha\mathbf{p}_c \neq \mathbf{0}$, where $\mathbf{p}_c$ is a descent direction and satisfies the sufficient descent condition

$$\mathbf{g}_c^T\mathbf{p}_c \leq -\eta\|\mathbf{g}_c\|^2 \tag{12}$$

for $\eta > 0$.

Based on the optimization techniques [47, 61], there exists such a direction that satisfies (12). For clarity, we will give the concrete formula about $\mathbf{p}_c$ in the latter of this section.

Substituting into (10) and (11) with $\mathbf{v} = \mathbf{y}_c + \alpha \mathbf{p}_c$, we have

$$\mathbf{y}_{c+1}(\alpha) = \frac{1 - \alpha^2 \|\mathbf{p}_c\|^2}{1 + 2\alpha \mathbf{y}_c^T \mathbf{p}_c + \alpha^2 \|\mathbf{p}_c\|^2} \mathbf{y}_c + 2\alpha \frac{1 + \alpha \mathbf{y}_c^T \mathbf{p}_c}{1 + 2\alpha \mathbf{y}_c^T \mathbf{p}_c + \alpha^2 \|\mathbf{p}_c\|^2} \mathbf{p}_c. \tag{13}$$

Different from the classical iterative method, (13) can be viewed as a curvilinear search path. That is, $\mathbf{p}_c$ is not along the rectilinear direction from $\mathbf{y}_c$ to $\mathbf{y}_{c+1}(\alpha)$. So, we first intuitively demonstrate that there exists such a direction that (12) and $\mathbf{g}_c^T \big(\mathbf{y}_{c+1}(\alpha) - \mathbf{y}_c\big) < 0$ hold.

It can be deduced from (9) that

$$\mathbf{g}_c^T \big(\mathbf{y}_{c+1}(\alpha) - \mathbf{y}_c\big) = \mathbf{g}_c^T \mathbf{y}_{c+1}(\alpha) = 2\alpha \frac{1 + \alpha \mathbf{y}_c^T \mathbf{p}_c}{1 + 2\alpha \mathbf{y}_c^T \mathbf{p}_c + \alpha^2 \|\mathbf{p}_c\|^2} \mathbf{g}_c^T \mathbf{p}_c. \tag{14}$$

From (9) and (14), it is easy to see that $\mathbf{g}_c^T \big(\mathbf{y}_{c+1}(\alpha) - \mathbf{y}_c\big) < 0$, if $\mathbf{p}_c = -\mathbf{g}_c$. But, if $\mathbf{p}_c \neq -\mathbf{g}_c$ is just a descent direction, then $\mathbf{g}_c^T \big(\mathbf{y}_{c+1}(\alpha) - \mathbf{y}_c\big) < 0$ implies

$$1 + \alpha \mathbf{y}_c^T \mathbf{p}_c > 0. \tag{15}$$

That is because the denominator of the right hand in (14), $1 + 2\alpha \mathbf{y}_c^T \mathbf{p}_c + \alpha^2 \|\mathbf{p}_c\|^2 = \mathbf{v}^T \mathbf{v} > 0$. Furthermore, $\mathbf{y}_c^T \mathbf{p}_c = \|\mathbf{y}_c\| \|\mathbf{p}_c\| \cos(\theta) = \|\mathbf{p}_c\| \cos(\theta)$, where θ is the angle between $\mathbf{y}_c$ and $\mathbf{p}_c$, and then, the inequality (15) requires $\cos(\theta) > -\frac{1}{\alpha \|\mathbf{p}_c\|}$, i.e.,

$$\theta \in \Big(0, \arccos\big(\max\{-1, -\frac{1}{\alpha \|\mathbf{p}_c\|}\}\big)\Big). \tag{16}$$

Obviously, $\arccos\big(\max\{-1, -\frac{1}{\alpha \|\mathbf{p}_c\|}\}\big) \in (\frac{\pi}{2}, \pi)$, this means $\mathbf{g}_c^T \big(\mathbf{y}_{c+1}(\alpha) - \mathbf{y}_c\big) < 0$ holds when $\theta \in (0, \frac{\pi}{2}]$.

On the other hand, the sufficient descent condition (12) implies $-\mathbf{g}_c^T \mathbf{p}_c \geq \eta \|\mathbf{g}_c\|^2$. Let $\bar{\theta}$ is the angle between $\mathbf{p}_c$ and $-\mathbf{g}_c$, then $\cos(\bar{\theta}) \geq \frac{\eta \|\mathbf{g}_c\|}{\|\mathbf{p}_c\|}$. According to the property of cosine function, it requires $\frac{\eta \|\mathbf{g}_c\|}{\|\mathbf{p}_c\|} \leq 1$, i.e., $\|\mathbf{p}_c\| \geq \eta \|\mathbf{g}_c\|$. Then, we have

$$\bar{\theta} \in \Big[0, \arccos\big(\frac{\eta \|\mathbf{g}_c\|}{\|\mathbf{p}_c\|}\big)\Big]. \tag{17}$$

Since $\mathbf{y}_c \perp \mathbf{g}_c$, the relationship between θ and $\bar{\theta}$ is as follows:

$$\theta = \begin{cases} \frac{\pi}{2} - \bar{\theta}, & \text{if } \theta \in (0, \frac{\pi}{2}], \\ \frac{\pi}{2} + \bar{\theta}, & \text{if } \theta \in \Big(\frac{\pi}{2}, \arccos\big(\max\{-1, -\frac{1}{\alpha \|\mathbf{p}_c\|}\}\big)\Big). \end{cases} \tag{18}$$

To avoid $\mathbf{p}_c$ being too close to $\mathbf{y}_c$ or $-\mathbf{y}_c$, θ can be neither too close to 0 nor too close to π. Taking this into account, we require $\bar{\theta} \leq \bar{\theta}_u$, where $\bar{\theta}_u$ is a fixed angle satisfying $\arccos\left(\frac{\eta\|\mathbf{g_c}\|}{\|\mathbf{p}_c\|}\right) \leq \bar{\theta}_u < \frac{\pi}{2}$. It follows from (18) that $\frac{\pi}{2} - \bar{\theta}_u \leq \theta \leq \frac{\pi}{2} + \bar{\theta}_u$. This ensures that θ is neither too close to 0 nor too close to π. It is worthy of noting that $\arccos\left(\frac{\eta\|\mathbf{g_c}\|}{\|\mathbf{p}_c\|}\right) \leq \bar{\theta}_u$ implies

$$\frac{\eta\|\mathbf{g_c}\|}{\|\mathbf{p}_c\|} \geq \cos(\bar{\theta}_u), \tag{19}$$

this means

$$\|\mathbf{p}_c\| \leq \eta\|\mathbf{g_c}\|/\cos(\bar{\theta}_u) = U_0\|\mathbf{g_c}\|. \tag{20}$$

where $U_0 = \eta/\cos(\bar{\theta}_u)$. In practice, we set $\bar{\theta}_u = \frac{\pi}{2} - \epsilon$, and judge whether (19) is true in the process of iteration. If it is not true, we set $\mathbf{p}_c = -\mathbf{g}_c$, then (15), (19) and (20) all hold.

From the meanings of θ and $\bar{\theta}$ whose ranges are expressed respectively by (16) and (17), we know that there exists such a direction $\mathbf{p}_c$ that (12) and $\mathbf{g}_c^T\left(\mathbf{y}_{c+1}(\alpha) - \mathbf{y}_c\right) < 0$ hold. The following is a figure that illustrates the relationship among $\mathbf{p}_c$, $\mathbf{y}_c$ and $\mathbf{g}_c$.

For spherical constraint problems, if the current iterate $\mathbf{y}_c$ is on the sphere, then $\mathbf{y}_{c+1}$ is also on the sphere if and only if the vector $\mathbf{y}_{c+1} - \mathbf{y}_c$ and $\mathbf{y}_{c+1} + \mathbf{y}_c$ are perpendicular to each other [5]; see Fig. 2.

We next verify the new iterate $\mathbf{y}_{c+1}(\alpha)$ computed by (13) can satisfy the property, i.e., $\left(\mathbf{y}_{c+1}(\alpha) - \mathbf{y}_c\right)^T\left(\mathbf{y}_{c+1}(\alpha) + \mathbf{y}_c\right) = 0$.

By direct calculation, we have

$$\mathbf{y}_{c+1}(\alpha) - \mathbf{y}_c = \frac{-2\alpha\mathbf{y}_c^T\mathbf{p}_c - 2\alpha^2\|\mathbf{p}_c\|^2}{1 + 2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2}\mathbf{y}_c + 2\alpha\frac{1 + \alpha\mathbf{y}_c^T\mathbf{p}_c}{1 + 2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2}\mathbf{p}_c,$$

$$\mathbf{y}_{c+1}(\alpha) + \mathbf{y}_c = \frac{2 + 2\alpha\mathbf{y}_c^T\mathbf{p}_c}{1 + 2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2}\mathbf{y}_c + 2\alpha\frac{1 + \alpha\mathbf{y}_c^T\mathbf{p}_c}{1 + 2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2}\mathbf{p}_c$$

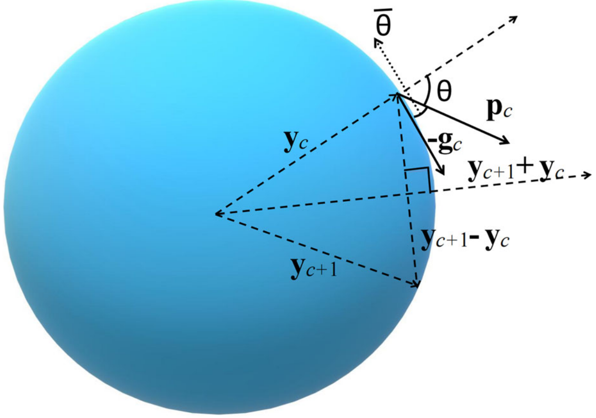


Fig. 2 A figure that illustrates the relationship among $\mathbf{p}_c$, $\mathbf{y}_c$ and $\mathbf{g}_c$, and the property of new iterate $\mathbf{y}_{c+1}$

then,

$$
\begin{aligned}
&\left(\mathbf{y}_{c+1}(\alpha)-\mathbf{y}_c\right)^T\left(\mathbf{y}_{c+1}(\alpha)+\mathbf{y}_c\right) \\
&=\frac{1}{\left(1+2\alpha\mathbf{y}_c^T\mathbf{p}_c+\alpha^2\|\mathbf{p}_c\|^2\right)^2}\Big(-4\alpha\mathbf{y}_c^T\mathbf{p}_c-4\alpha^2\|\mathbf{p}_c\|^2-4\alpha^2(\mathbf{y}_c^T\mathbf{p}_c)^2-4\alpha^3\mathbf{y}_c^T\mathbf{p}_c\|\mathbf{p}_c\|^2 \\
&\quad+4\alpha\mathbf{y}_c^T\mathbf{p}_c+4\alpha^2(\mathbf{y}_c^T\mathbf{p}_c)^2-4\alpha^3\mathbf{y}_c^T\mathbf{p}_c\|\mathbf{p}_c\|^2-4\alpha^4(\mathbf{y}_c^T\mathbf{p}_c)^2\|\mathbf{p}_c\|^2 \\
&\quad+4\alpha^2\|\mathbf{p}_c\|^2+8\alpha^3\mathbf{y}_c^T\mathbf{p}_c\|\mathbf{p}_c\|^2+4\alpha^4(\mathbf{y}_c^T\mathbf{p}_c)^2\|\mathbf{p}_c\|^2\Big) \\
&=0.
\end{aligned}
$$

In the following, we demonstrate how to compute $\mathbf{p}_c$ that makes the sufficient descent condition (12) hold.

Obviously, $\mathbf{p}_c = -\mathbf{g}_c$ is suitable and feasible. On the other hand, it is well known that the conjugate gradient (CG) method is efficient for large-scale problems [17]. So, we develop a family of sufficient descent directions $\mathbf{p}_c$ based on the classical CG directions in the following.

The framework of the conjugate gradient direction is defined by

$$
\mathbf{p}_c=\begin{cases}-\mathbf{g}_c, & c=1,\\ -\mathbf{g}_c+\beta_{c-1}\mathbf{p}_{c-1}, & c\geq 2.\end{cases} \tag{21}
$$

Different choices for β_c determine different conjugate gradient methods. We mainly consider several classical conjugate gradient methods, e.g., *FR-CG* [24], *PRP-CG* [49, 50], *HS-CG* [32], *CD-CG* [23], *LS-CG* [46], and *DY-CG* [16]. Their detailed formulas are as follows:

$$
\begin{aligned}
\beta_c^{FR}&=\frac{\|\mathbf{g}_{c+1}\|^2}{\|\mathbf{g}_c\|^2},\\
\beta_c^{PRP}&=\frac{\mathbf{g}_{c+1}^T(\mathbf{g}_{c+1}-\mathbf{g}_c)}{\|\mathbf{g}_c\|^2},\\
\beta_c^{HS}&=\frac{\mathbf{g}_{c+1}^T(\mathbf{g}_{c+1}-\mathbf{g}_c)}{\mathbf{p}_c^T(\mathbf{g}_{c+1}-\mathbf{g}_c)},\\
\beta_c^{CD}&=\frac{\|\mathbf{g}_{c+1}\|^2}{-\mathbf{p}_c^T\mathbf{g}_c},\\
\beta_c^{LS}&=\frac{\mathbf{g}_{c+1}^T(\mathbf{g}_{c+1}-\mathbf{g}_c)}{-\mathbf{p}_c^T\mathbf{g}_c},\\
\beta_c^{DY}&=\frac{\|\mathbf{g}_{c+1}\|^2}{\mathbf{p}_c^T(\mathbf{g}_{c+1}-\mathbf{g}_c)}.
\end{aligned} \tag{22}
$$

Besides, it is reported that good numerical results can be obtained by taking a hybrid scheme of DY and HS [17], where the corresponding parameter β_c is given by

$$
\beta_c^{DYHS}=\max\left\{0,\min\{\beta_c^{DY},\beta_c^{HS}\}\right\}. \tag{23}
$$

No matter which formula presented above is adopted, it has been proved that the sufficient descent property (12) plays an important role in convergence analysis for the CG methods [26, 28, 58, 59]. However, the CG directions corresponding to the above formulae (22) and (23) could not theoretically guarantee the sufficient descent condition. Dai [15] proved that under some certain conditions, the sufficient descent condition of DY direction holds for a certain number of iterations (see [15, Theorem 2.2]).

Yuan et al. [60] proposed a family of weak conjugate gradient formulae based on six classical formulae, as follows:

$$\mathbf{p}_c^* = -\mathbf{g}_c + \rho_0 \frac{\|\mathbf{g}_c\|}{\|\mathbf{p}_c\|}\mathbf{p}_c,$$

where $\rho_0 \in (0, 1)$, $\mathbf{p}_c$ is a classical CG direction. It is easy to verify that $\mathbf{p}_c^*$ satisfies $\mathbf{g}_c^T \mathbf{p}_c^* \leq -(1-\rho_0)\|\mathbf{g}_c\|^2$ and $\|\mathbf{p}_c^*\| \leq (1+\rho_0)\|\mathbf{g}_c\|$. The proposed family of weak conjugate gradient algorithms has better theory properties [60], while their advantage in numerical computation is not very obvious.

Hager and Zhang [28] proposed a new CG method with guaranteed descent, where the corresponding β_c is

$$\beta_c^N = \left((\mathbf{g}_{c+1} - \mathbf{g}_c) - 2\mathbf{p}_c \frac{\|\mathbf{g}_{c+1} - \mathbf{g}_c\|^2}{\mathbf{p}_c^T(\mathbf{g}_{c+1} - \mathbf{g}_c)}\right)^T \frac{\mathbf{g}_{c+1}}{\mathbf{p}_c^T(\mathbf{g}_{c+1} - \mathbf{g}_c)}.$$

The direction determined by β_c^N possesses the sufficient descent condition $\mathbf{g}_c^T \mathbf{p}_c \leq -\frac{7}{8}\|\mathbf{g}_c\|^2$. Hager and Zhang summarized different versions of nonlinear conjugate gradient methods in [29] and proposed a more general parameter denoted by β_c^{HZ}, as follows:

$$\begin{aligned}\beta_c^{HZ} &= \left((\mathbf{g}_{c+1} - \mathbf{g}_c) - b\,\mathbf{p}_c \frac{\|\mathbf{g}_{c+1} - \mathbf{g}_c\|^2}{\mathbf{p}_c^T(\mathbf{g}_{c+1} - \mathbf{g}_c)}\right)^T \frac{\mathbf{g}_{c+1}}{\mathbf{p}_c^T(\mathbf{g}_{c+1} - \mathbf{g}_c)}, \\ &= \beta_c^{HS} - b\left(\frac{\|\mathbf{g}_{c+1} - \mathbf{g}_c\|^2 \mathbf{g}_{c+1}^T \mathbf{p}_c}{\left(\mathbf{p}_c^T(\mathbf{g}_{c+1} - \mathbf{g}_c)\right)^2}\right).\end{aligned} \tag{24}$$

where $b > \frac{1}{4}$. If $b = 2$, β_c^{HZ} reduces to β_c^N. The direction $\mathbf{p}_c$ computed by (21) and β_c^{HZ} satisfies $\mathbf{g}_c^T \mathbf{p}_c \leq -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2$, and this result is independent of the inexact line search strategy. Furthermore, the global convergence result is established when the line search fulfills the Wolfe conditions [28, 29]. By the numerical results presented in [28], this new CG method is superior to some classical CG methods, such as PRP, DY, and the hybrid of DY and HS.

Inspired by the idea in [28, 29], we will further explore the CG directions satisfying the sufficient descent condition.

Substituting into (21) with (24), we have that

$$\begin{aligned}\mathbf{g}_c^T\mathbf{p}_c &= \mathbf{g}_c^T(-\mathbf{g}_c + \beta_{c-1}^{HZ}\mathbf{p}_{c-1})\\ &= -\|\mathbf{g}_c\|^2 + \left(\frac{\mathbf{g}_c^T(\mathbf{g}_c - \mathbf{g}_{c-1})(\mathbf{g}_c^T\mathbf{p}_{c-1})}{\mathbf{p}_{c-1}^T(\mathbf{g}_c - \mathbf{g}_{c-1})}\right) - b\left(\frac{\|\mathbf{g}_c - \mathbf{g}_{c-1}\|^2(\mathbf{g}_c^T\mathbf{p}_{c-1})^2}{\left(\mathbf{p}_{c-1}^T(\mathbf{g}_c - \mathbf{g}_{c-1})\right)^2}\right).\end{aligned} \tag{25}$$

Let $\mathbf{t} = \frac{1}{2\sqrt{b}}\mathbf{g}_c - \sqrt{b}\frac{\mathbf{g}_c^T\mathbf{p}_{c-1}}{\mathbf{p}_{c-1}^T(\mathbf{g}_c - \mathbf{g}_{c-1})}(\mathbf{g}_c - \mathbf{g}_{c-1})$, and substitute it into (25), it follows

$$\mathbf{g}_c^T\mathbf{p}_c = -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2 - \mathbf{t}^T\mathbf{t} \leq -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2.$$

By observation, we find that the third item of the second equality in (25) is strictly less than 0, which implies that β_c^{HZ} can actually make $\mathbf{g}_c^T\mathbf{p}_c$ become smaller than β_c^{HS}, i.e., it has enhanced descent [29]. Moreover, the second equality in (24) indicates that β_c^{HZ} can be regarded as subtracting an item from β_c^{HS}, and the subtracted item makes the negative square term appeared in (25). Hence, the CG direction computed by β_c^{HZ} satisfies the sufficient descent condition (12).

Motivated by this observation, we try to construct a family of CG directions with sufficient descent based on the classical formulae presented in (22). We call them the modified CG directions, where the corresponding parameters β_c are

$$\begin{aligned}\beta_c^{MFR} &= \beta_c^{FR} - b\left(\frac{\|\mathbf{g}_{c+1}\|^2\mathbf{g}_{c+1}^T\mathbf{p}_c}{\|\mathbf{g}_c\|^4}\right),\\ \beta_c^{MPRP} &= \beta_c^{PRP} - b\left(\frac{\|\mathbf{g}_{c+1} - \mathbf{g}_c\|^2\mathbf{g}_{c+1}^T\mathbf{p}_c}{\|\mathbf{g}_c\|^4}\right),\\ \beta_c^{MCD} &= \beta_c^{CD} - b\left(\frac{\|\mathbf{g}_{c+1}\|^2\mathbf{g}_{c+1}^T\mathbf{p}_c}{(\mathbf{p}_c^T\mathbf{g}_c)^2}\right),\\ \beta_c^{MLS} &= \beta_c^{LS} - b\left(\frac{\|\mathbf{g}_{c+1} - \mathbf{g}_c\|^2\mathbf{g}_{c+1}^T\mathbf{p}_c}{(\mathbf{p}_c^T\mathbf{g}_c)^2}\right),\\ \beta_c^{MDY} &= \beta_c^{DY} - b\left(\frac{\|\mathbf{g}_{c+1}\|^2\mathbf{g}_{c+1}^T\mathbf{p}_c}{\left(\mathbf{p}_c^T(\mathbf{g}_{c+1} - \mathbf{g}_c)\right)^2}\right),\end{aligned} \tag{26}$$

where $b > \frac{1}{4}$, β_c^{FR}, β_c^{PRP}, β_c^{CD}, β_c^{LS} and β_c^{CD} are defined by (22). The parameter β_c^{HZ} (24) can be regarded as a modified version of β_c^{HS}, so it is not listed in (26). Similar to (23), the hybrid of β_c^{MDY} and β_c^{MHS} is given by

$$\beta_c^{MDYHS} = \max\left\{0, \min\{\beta_c^{MDY}, \beta_c^{MHS}\}\right\}. \tag{27}$$

It is worthy noting that the denominator of the parameters β_c both in (22) and (26) cannot be zero. In the process of iteration, if $\mathbf{p}_c$ satisfies the sufficient descent condition (12) and the line search satisfies the Wolfe conditions, then the denominator is not zero when $\|\mathbf{g}_c\| > \epsilon$. From the sufficient descent condition, it can be deduced that $-\mathbf{p}_c^T\mathbf{g}_c \geq \eta\|\mathbf{g}_c\|^2 > \eta\epsilon^2$, and hence $(\mathbf{p}_c^T\mathbf{g}_c)^2 > \eta^2\epsilon^4$. From the Wolfe conditions (35), it follows that

$$(\mathbf{g}_{c+1} - \mathbf{g}_c)^T\mathbf{p}_c \geq (\bar{\sigma} - 1)\mathbf{g}_c^T\mathbf{p}_c > (1 - \bar{\sigma})\eta\epsilon^2,$$

then, we have

$$\left((\mathbf{g}_{c+1} - \mathbf{g}_c)^T\mathbf{p}_c\right)^2 > (1 - \bar{\sigma})^2\eta^2\epsilon^4.$$

The modified CG scheme has the following property.

Proposition 1 *Suppose that $\mathbf{p}_c$ is computed by (21) and (26), then for any (inexact) line search, $\mathbf{p}_c$ satisfies the descent condition $\mathbf{g}_c^T\mathbf{p}_c \leq -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2$, where $b > \frac{1}{4}$.*

Proof For modified FR formula in (26), we have

$$\begin{aligned}
\mathbf{g}_c^T\mathbf{p}_c &= \mathbf{g}_c^T(-\mathbf{g}_c + \beta_{c-1}^{MFR}\mathbf{p}_{c-1}) \\
&= -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2 - \left(\frac{1}{4b}\|\mathbf{g}_c\|^2 - \left(\frac{\|\mathbf{g}_c\|^2(\mathbf{g}_c^T\mathbf{p}_{c-1})}{\|\mathbf{g}_{c-1}\|^2}\right) + b\left(\frac{\|\mathbf{g}_c\|^2(\mathbf{g}_c^T\mathbf{p}_{c-1})^2}{\|\mathbf{g}_{c-1}\|^4}\right)\right) \\
&= -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2 - \left(\frac{1}{2\sqrt{b}}\mathbf{g}_c - \sqrt{b}\frac{\mathbf{g}_c^T\mathbf{p}_{c-1}}{\|\mathbf{g}_{c-1}\|^2}\mathbf{g}_c\right)^T\left(\frac{1}{2\sqrt{b}}\mathbf{g}_c - \sqrt{b}\frac{\mathbf{g}_c^T\mathbf{p}_{c-1}}{\|\mathbf{g}_{c-1}\|^2}\mathbf{g}_c\right) \\
&\leq -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2.
\end{aligned}$$

For modified PRP formula in (26),

$$\begin{aligned}
\mathbf{g}_c^T\mathbf{p}_c &= \mathbf{g}_c^T(-\mathbf{g}_c + \beta_{c-1}^{MPRP}\mathbf{p}_{c-1}) \\
&= -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2 - \left(\frac{1}{2\sqrt{b}}\mathbf{g}_c - \sqrt{b}\frac{\mathbf{g}_c^T\mathbf{p}_{c-1}}{\|\mathbf{g}_{c-1}\|^2}(\mathbf{g}_c - \mathbf{g}_{c-1})\right)^T \\
&\qquad \left(\frac{1}{2\sqrt{b}}\mathbf{g}_c - \sqrt{b}\frac{\mathbf{g}_c^T\mathbf{p}_{c-1}}{\|\mathbf{g}_{c-1}\|^2}(\mathbf{g}_c - \mathbf{g}_{c-1})\right) \\
&\leq -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2.
\end{aligned}$$

For modified CD formula in (26),

$$\begin{aligned}
\mathbf{g}_c^T\mathbf{p}_c &= \mathbf{g}_c^T(-\mathbf{g}_c + \beta_{c-1}^{MCD}\mathbf{p}_{c-1}) \\
&= -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2 - \left(\frac{1}{2\sqrt{b}}\mathbf{g}_c + \sqrt{b}\frac{\mathbf{g}_c^T\mathbf{p}_{c-1}}{\mathbf{p}_{c-1}^T\mathbf{g}_{c-1}}\mathbf{g}_c\right)^T\left(\frac{1}{2\sqrt{b}}\mathbf{g}_c + \sqrt{b}\frac{\mathbf{g}_c^T\mathbf{p}_{c-1}}{\mathbf{p}_{c-1}^T\mathbf{g}_{c-1}}\mathbf{g}_c\right) \\
&\leq -\left(1 - \frac{1}{4b}\right)\|\mathbf{g}_c\|^2.
\end{aligned}$$

For modified LS formula in (26),

$$
\begin{aligned}
\mathbf{g}_c^T \mathbf{p}_c &= \mathbf{g}_c^T (-\mathbf{g}_c + \beta_{c-1}^{MLS} \mathbf{p}_{c-1}) \\
&= -\left(1 - \frac{1}{4b}\right) \|\mathbf{g}_c\|^2 - \left(\frac{1}{2\sqrt{b}}\mathbf{g}_c + \sqrt{b}\frac{\mathbf{g}_c^T \mathbf{p}_{c-1}}{\mathbf{p}_{c-1}^T \mathbf{g}_{c-1}}\left(\mathbf{g}_c - \mathbf{g}_{c-1}\right)\right)^T \\
&\qquad \left(\frac{1}{2\sqrt{b}}\mathbf{g}_c + \sqrt{b}\frac{\mathbf{g}_c^T \mathbf{p}_{c-1}}{\mathbf{p}_{c-1}^T \mathbf{g}_{c-1}}\left(\mathbf{g}_c - \mathbf{g}_{c-1}\right)\right) \\
&\le -\left(1 - \frac{1}{4b}\right) \|\mathbf{g}_c\|^2.
\end{aligned}
$$

For modified DY formula in (26),

$$
\begin{aligned}
\mathbf{g}_c^T \mathbf{p}_c &= \mathbf{g}_c^T (-\mathbf{g}_c + \beta_{c-1}^{MDY} \mathbf{p}_{c-1}) \\
&= -\left(1 - \frac{1}{4b}\right) \|\mathbf{g}_c\|^2 - \left(\frac{1}{2\sqrt{b}}\mathbf{g}_c - \sqrt{b}\frac{\mathbf{g}_c^T \mathbf{p}_{c-1}}{\mathbf{d}_{c-1}^T\left(\mathbf{g}_c - \mathbf{g}_{c-1}\right)}\mathbf{g}_c\right)^T \\
&\qquad \left(\frac{1}{2\sqrt{b}}\mathbf{g}_c - \sqrt{b}\frac{\mathbf{g}_c^T \mathbf{p}_{c-1}}{\mathbf{d}_{c-1}^T\left(\mathbf{g}_c - \mathbf{g}_{c-1}\right)}\mathbf{g}_c\right) \\
&\le -\left(1 - \frac{1}{4b}\right) \|\mathbf{g}_c\|^2.
\end{aligned}
$$

□

Since the modified parameter β_c is obtained by subtracting a modification term from the classical formula, the probability of $\beta_c < 0$ may increase. As a result, we compute $\mathbf{p}_c$ by the following formula in practice:

$$\beta_c = \max\{\, 0,\ \beta_c^{(X)} \,\}, \tag{28}$$

where $\beta_c^{(X)}$ represents those different β_c given in (22) and (26). This is feasible, since $\beta_{c-1} = 0$ indicates $\mathbf{p}_c = -\mathbf{g}_c$. It is interesting that if $\mathbf{p}_c = -\mathbf{g}_c$, then the new iterate computed by (13) is the same as that in [8], which is determined by Cayley transform. The reason for this phenomenon is the gradient $\mathbf{g}_c$ always satisfies $\mathbf{y}_c^T \mathbf{g}_c = 0$, but it is not true for a general descent direction. Therefore, the new iterate scheme (13) derived by Householder transform is different from that derived by Cayley transform.

Next, we further prove that on the condition of $\mathbf{p}_c$ being descent, there exists a positive step size α_c in the new iterate scheme (13) such that the Wolfe conditions hold. In order to prove this property, we give the following two lemmas first.

Lemma 1 *Suppose that $\mathbf{y}_c$ is the current iterate, $\mathbf{p}_c$ is a descent search direction, $\mathbf{y}_{c+1}(\alpha)$ computed by (13) is a new iterate, then*

$$\left.\frac{d\, f\,(\mathbf{y}_{c+1}(\alpha))}{d\,\alpha}\right|_{\alpha=0} = 2\mathbf{g}_c^T \mathbf{p}_c. \tag{29}$$

where $\mathbf{g}_c = \mathbf{g}(\mathbf{y}_c)$.

Proof Taking derivative with respect to α, we have

$$\begin{aligned}\mathbf{y}_{c+1}^{'}(\alpha) = &\frac{(2+4\alpha\mathbf{y}_c^T\mathbf{p}_c)\mathbf{p}_c - 2\alpha\|\mathbf{p}_c\|^2\mathbf{y}_c}{1+2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2}\\ &- \frac{(2\mathbf{y}_c^T\mathbf{p}_c + 2\alpha\|\mathbf{p}_c\|^2)\left((2\alpha+2\alpha^2\mathbf{y}_c^T\mathbf{p}_c)\mathbf{p}_c - (\alpha^2\|\mathbf{p}_c\|^2-1)\mathbf{y}_c\right)}{(1+2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2)^2}.\end{aligned}$$

This indicates $\mathbf{y}_{c+1}^{'}(0) = 2\mathbf{p}_c - 2\mathbf{y}_c^T\mathbf{p}_c\mathbf{y}_c$. Combining with $\mathbf{y}_{c+1}(0) = \mathbf{y}_c$, we obtain

$$\left.\frac{d\,f\,(\mathbf{y}_{c+1}(\alpha))}{d\,\alpha}\right|_{\alpha=0} = \mathbf{g}(\mathbf{y}_{c+1}(0))^T\mathbf{y}_{c+1}^{'}(0) = 2\mathbf{g}(\mathbf{y}_c)^T\mathbf{p}_c - 2(\mathbf{y}_c^T\mathbf{p}_c)(\mathbf{g}(\mathbf{y}_c)^T\mathbf{y}_c).$$

Note that $\mathbf{y}^T\mathbf{g}(\mathbf{y}) = 0$, substituting it into the above formula, it concludes (29). □

Lemma 2 *Suppose that* $\mathbf{y}_c$ *is the current iterate,* $\mathbf{p}_c$ *is a descent search direction,* $\mathbf{y}_{c+1}(\alpha)$ *computed by (13) is a new iterate. Denote* $\hat{f}(\alpha) = f\left(\mathbf{y}_{c+1}(\alpha)\right)$*, then*

$$\alpha\hat{f}'(\alpha) = -2\triangledown f(\mathbf{y}_{c+1}(\alpha))^T\frac{\mathbf{y}_c+\alpha\mathbf{p}_c}{\|\mathbf{y}_c+\alpha\mathbf{p}_c\|^2}. \tag{30}$$

Proof It follows from (13) that

$$(1+2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2)\mathbf{y}_{c+1}(\alpha) = (1-\alpha^2\|\mathbf{p}_c\|^2)\mathbf{y}_c + (2\alpha+2\alpha^2\mathbf{y}_c^T\mathbf{p}_c)\mathbf{p}_c. \tag{31}$$

Taking derivative about α on both sides of (31), it follows

$$\begin{aligned}\left(2\mathbf{y}_c^T\mathbf{p}_c + 2\alpha\|\mathbf{p}_c\|^2\right)\mathbf{y}_{c+1}(\alpha) + \left(1+2\alpha\mathbf{y}_c^T\mathbf{p}_c+\alpha^2\|\mathbf{p}_c\|^2\right)\mathbf{y}_{c+1}'(\alpha)\\ = -2\alpha\|\mathbf{p}_c\|^2\mathbf{y}_c + (2+4\alpha\mathbf{y}_c^T\mathbf{p}_c)\mathbf{p}_c.\end{aligned} \tag{32}$$

Multiplying both sides of (32) by α, we obtain

$$\begin{aligned}\alpha\mathbf{y}_{c+1}'(\alpha) &= -\frac{2\mathbf{y}_c^T\mathbf{p}_c + 2\alpha\|\mathbf{p}_c\|^2}{1+2\alpha\mathbf{y}_c^T\mathbf{p}_c+\alpha^2\|\mathbf{p}_c\|^2}\mathbf{y}_{c+1}(\alpha) + \frac{-2\alpha^2\|\mathbf{p}_c\|^2\mathbf{y}_c + (2\alpha+4\alpha^2\mathbf{y}_c^T\mathbf{p}_c)\mathbf{p}_c}{1+2\alpha\mathbf{y}_c^T\mathbf{p}_c+\alpha^2\|\mathbf{p}_c\|^2}\\ &= -\frac{2\mathbf{y}_c^T\mathbf{p}_c + 2\alpha\|\mathbf{p}_c\|^2}{1+2\alpha\mathbf{y}_c^T\mathbf{p}_c+\alpha^2\|\mathbf{p}_c\|^2}\mathbf{y}_{c+1}(\alpha) + 2\mathbf{y}_{c+1}(\alpha) - 2\frac{\mathbf{y}_c+\alpha\mathbf{p}_c}{1+2\alpha\mathbf{y}_c^T\mathbf{p}_c+\alpha^2\|\mathbf{p}_c\|^2}.\end{aligned} \tag{33}$$

Notice that the denominator of (33) is actually $\|\mathbf{y}_c+\alpha\mathbf{p}_c\|^2$ for the reason that $\mathbf{y}_c^T\mathbf{y}_c = 1$. In addition, $\triangledown f(\mathbf{y}_{c+1}(\alpha))^T\mathbf{y}_{c+1}(\alpha) = 0$ holds because of (9), and hence, we have

$$\alpha\hat{f}'(\alpha) = \alpha\triangledown f(\mathbf{y}_{c+1}(\alpha))^T\mathbf{y}_{c+1}'(\alpha) = -2\triangledown f(\mathbf{y}_{c+1}(\alpha))^T\frac{\mathbf{y}_c+\alpha\mathbf{p}_c}{\|\mathbf{y}_c+\alpha\mathbf{p}_c\|^2}.$$

This completes the proof. □

By Lemmas 1 and 2, we can further prove the following result.

Lemma 3 *Suppose that* $\mathbf{y}_c$ *is the current iterate,* $\mathbf{p}_c$ *is a descent search direction,* $\mathbf{y}_{c+1}(\alpha)$ *computed by (13) is a new iterate. Then, for* $\rho \in (0, 2)$, $\bar{\sigma} \in [\frac{\rho}{2}, 1)$, *there exists* $\alpha_c \in (0, \bar{\alpha}\,]$, $\bar{\alpha} > 0$ *satisfying the following Wolfe conditions*

$$f\left(\mathbf{y}_{c+1}(\alpha_c)\right) - f(\mathbf{y}_c) \leq \rho\alpha_c\mathbf{g}_c^T\mathbf{p}_c, \tag{34}$$

$$\bigtriangledown f(\mathbf{y}_{c+1}(\alpha_c))^T\mathbf{p}_c \geq \bar{\sigma}\mathbf{g}_c^T\mathbf{p}_c, \tag{35}$$

where $\mathbf{g}_c = \mathbf{g}(\mathbf{y}_c)$. *Inequality (34) is also known as Armijo criterion.*

Proof Define a linear function $l(\alpha) = f(\mathbf{y}_c) + \rho\alpha\mathbf{g}_c^T\mathbf{p}_c$. It follows that $l'(\alpha) = \rho\mathbf{g}_c^T\mathbf{p}_c$. Let $\hat{f}(\alpha) = f\left(\mathbf{y}_{c+1}(\alpha)\right)$, then $\hat{f}(0) = f(\mathbf{y}_{c+1}(0)) = f(\mathbf{y}_c) = l(0)$. By Lemma 1, we have that $\hat{f}'(0) = 2\mathbf{g}_c^T\mathbf{p}_c$. Since $\mathbf{p}_c$ is a descent direction, when $0 < \rho < 2$, we obtain

$$\hat{f}'(0) < l'(0) < 0. \tag{36}$$

Note that $\|\mathbf{y}_{c+1}(\alpha)\| = 1$, i.e., $\hat{f}(\alpha) = f\left(\mathbf{y}_{c+1}(\alpha)\right)$ is defined on the unit sphere, hence it is bounded below. In addition, $l(\alpha)$ is a monotonically decreasing function on $(0, \infty)$, so the graph of $\hat{f}(\alpha)$ must intersect with the line $l(\alpha)$ at least once when $\alpha > 0$. Suppose $\bar{\alpha}$ is the smallest intersection point, then $\bar{\alpha} > 0$. Further, owing to (36), we obtain that the graph of $\hat{f}(\alpha)$ is below the line $l(\alpha)$ when $\rho \in (0, 2)$, $\alpha \in (0, \bar{\alpha})$. That is, when $\rho \in (0, 2)$, $\alpha \in (0, \bar{\alpha}\,]$, we have

$$f\left(\mathbf{y}_{c+1}(\alpha)\right) = \hat{f}(\alpha) \leq l(\alpha) = f(\mathbf{y}_c) + \rho\alpha\mathbf{g}_c^T\mathbf{p}_c, \tag{37}$$

where the equal holds if and only if $\alpha = \bar{\alpha}$. This concludes (34).

By the mean value theorem, there exists $\xi \in (0, \bar{\alpha})$ such that

$$f\left(\mathbf{y}_{c+1}(\bar{\alpha})\right) - f(\mathbf{y}_c) = \hat{f}(\bar{\alpha}) - \hat{f}(0) = \bar{\alpha}\hat{f}'(\xi).$$

According to Lemma 2, we have $\hat{f}'(\xi) = -\frac{2}{\xi}\bigtriangledown f(\mathbf{y}_{c+1}(\xi))^T\frac{\mathbf{y}_c+\xi\mathbf{p}_c}{\|\mathbf{y}_c+\xi\mathbf{p}_c\|^2}$, it then follows

$$f\left(\mathbf{y}_{c+1}(\bar{\alpha})\right) - f(\mathbf{y}_c) = -\frac{2\bar{\alpha}}{\xi}\bigtriangledown f(\mathbf{y}_{c+1}(\xi))^T\frac{\mathbf{y}_c + \xi\mathbf{p}_c}{\|\mathbf{y}_c + \xi\mathbf{p}_c\|^2}. \tag{38}$$

On the other hand, by (9) and (13), we have

$$\begin{aligned} 0 = \bigtriangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{y}_{c+1}(\xi) = \frac{1 - \xi^2\|\mathbf{p}_c\|^2}{\|\mathbf{y}_c + \xi\mathbf{p}_c\|^2}\bigtriangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{y}_c \\ + \frac{2\xi + 2\xi^2\mathbf{y}_c^T\mathbf{p}_c}{\|\mathbf{y}_c + \xi\mathbf{p}_c\|^2}\bigtriangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{p}_c. \end{aligned}$$

It then follows

$$-\frac{1-\xi^2\|\mathbf{p}_c\|^2}{\|\mathbf{y}_c+\xi\mathbf{p}_c\|^2}\triangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{y}_c=\frac{2\xi+2\xi^2\mathbf{y}_c^T\mathbf{p}_c}{\|\mathbf{y}_c+\xi\mathbf{p}_c\|^2}\triangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{p}_c. \tag{39}$$

Adding $-\frac{1-\xi^2\|\mathbf{p}_c\|^2}{\|\mathbf{y}_c+\xi\mathbf{p}_c\|^2}\triangledown f(\mathbf{y}_{c+1}(\xi))^T(\xi\mathbf{p}_c)$ to both sides of (39), we have that

$$\begin{aligned} &-(1-\xi^2\|\mathbf{p}_c\|^2)\triangledown f(\mathbf{y}_{c+1}(\xi))^T\frac{\mathbf{y}_c+\xi\mathbf{p}_c}{\|\mathbf{y}_c+\xi\mathbf{p}_c\|^2}\\ &=\frac{1+2\xi\mathbf{y}_c^T\mathbf{p}_c+\xi^2\|\mathbf{p}_c\|^2}{\|\mathbf{y}_c+\xi\mathbf{p}_c\|^2}\triangledown f(\mathbf{y}_{c+1}(\xi))^T(\xi\mathbf{p}_c)\\ &=\xi\triangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{p}_c. \end{aligned} \tag{40}$$

Equation (38) indicates that

$$\triangledown f(\mathbf{y}_{c+1}(\xi))^T\frac{\mathbf{y}_c+\xi\mathbf{p}_c}{\|\mathbf{y}_c+\xi\mathbf{p}_c\|^2}=-\frac{\xi}{2\bar{\alpha}}\left(f\left(\mathbf{y}_{c+1}(\bar{\alpha})\right)-f(\mathbf{y}_c)\right),$$

substituting it into (40), we can check that

$$(1-\xi^2\|\mathbf{p}_c\|^2)\left(f\left(\mathbf{y}_{c+1}(\bar{\alpha})\right)-f(\mathbf{y}_c)\right)=2\bar{\alpha}\triangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{p}_c. \tag{41}$$

Taking α as $\bar{\alpha}$ in (37), we get $f\left(\mathbf{y}_{c+1}(\bar{\alpha})\right)-f(\mathbf{y}_c)=\rho\bar{\alpha}\mathbf{g}_c^T\mathbf{p}_c$. Substituting it into (41), we have

$$(1-\xi^2\|\mathbf{p}_c\|^2)\rho\bar{\alpha}\mathbf{g}_c^T\mathbf{p}_c=2\bar{\alpha}\triangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{p}_c. \tag{42}$$

Since $\mathbf{g}_c^T\mathbf{p}_c<0$ and $\bar{\alpha}>0$, it can be deduced from (42) that

$$\triangledown f(\mathbf{y}_{c+1}(\xi))^T\mathbf{p}_c=\frac{\rho}{2}(1-\xi^2\|\mathbf{p}_c\|^2)\mathbf{g}_c^T\mathbf{p}_c>\frac{\rho}{2}\mathbf{g}_c^T\mathbf{p}_c. \tag{43}$$

Note that $\xi\in(0,\bar{\alpha})$, so the parameter ξ satisfying (43) can also make (37) hold. Let $\alpha_c=\xi$, $\bar{\sigma}\in[\frac{\rho}{2},1)$, then the conclusions (34) and (35) are both established. This completes the proof. □

Lemma 3 shows that Wolfe conditions are practicable for the curvilinear line search in the new iterate scheme (13). The inequality (34) guarantees the descent of the objective function, while the inequality (35) is mainly to prevent α_c from being too small.

Based on the above discussion, we present the following algorithm for solving (7).

Algorithm 1 Constraint preserving algorithm for the smallest Z-eigenpair.

Step 0. Choose initial parameters, $\rho \in (0, 2)$, $\bar{\sigma} \in (\rho/2, 1)$, $\bar{\alpha} = 1$, $\gamma \in (0.5, 1)$, $\epsilon > 0$, $\bar{\theta}_u = \frac{\pi}{2} - \epsilon$, and choose initial point $\mathbf{y}_1 \in \mathbb{R}^{n-1}$ such that $\mathbf{y}_1^T \mathbf{y}_1 = 1$. Let $c = 1$.
Step 1. Compute $\lambda_c = f(\mathbf{y}_c)$ and $\mathbf{g}(\mathbf{y}_c)$, then compute a descent direction $\mathbf{p}_c$ satisfying (12).
Step 2. Choose the smallest nonnegative integer l and determine $\alpha_c = \gamma^l \bar{\alpha}$ such that α_c satisfies (34) and (35) .
Step 3. Check whether $\mathbf{p}_c$ and α_c satisfy (19). If it is true, go to next step; otherwise, set $\mathbf{p}_c = -\mathbf{g}_c$ and go to Step 2.
Step 4. Calculate the new iterate $\mathbf{y}_{c+1}$ by (13).
Step 5. If $\|\mathbf{g}(\mathbf{y}_{c+1})\| < \epsilon$, set $\lambda^* = f(\mathbf{y}_{c+1})$, $\mathbf{y}^* = \mathbf{y}_{c+1}$ and stop; otherwise, update $\bar{\alpha}$ and let $c = c+1$, go to Step 1.

Remark 1 In Algorithm 1, $\mathbf{p}_c$ can be chosen as $-\mathbf{g}_c$, it can also be the modified CG directions computed by (21), (26) and (28). Additionally, in an effort to compare the efficiency of Algorithm 1 with different descent directions, we will test the algorithm with classical CG directions in Sect. 5 as well, which are determined by (21), (22) and (28).

4 Convergence analysis

In this section, we analyze the convergence of Algorithm 1. Because the feasible set $\mathbb{S}^{n-2}$ of problem (7) is a compact set, and the objective function is at least twice continuously differentiable on $\mathbb{S}^{n-2}$, there exists a constant M such that

$$|f(\mathbf{y})| \leq M, \quad \|\mathbf{g}(\mathbf{y})\| \leq M, \quad \forall \mathbf{y} \in \mathbb{S}^{n-2}. \tag{44}$$

Theorem 3 *Suppose that* $\{\lambda_c\}$ *is generated by Algorithm 1. Then* $\{\lambda_c\}$ *is a monotonically decreasing sequence, and there exists* λ^* *such that*

$$\lim_{c \to \infty} \lambda_c = \lambda^*.$$

Proof Since $\{\lambda_c\}$ is a monotonically decreasing sequence and bounded below, it converges to a unique point λ^*. □

Theorem 4 *Suppose that* $\{\mathbf{y}_c\}$ *is generated by Algorithm 1, and the step size* α_c *satisfies the Armijo condition (34), then the sequence* $\{\mathbf{y}_c\}$ *has at least one accumulation point. And we have*

$$\lim_{c \to \infty} \|\mathbf{g}(\mathbf{y}_c)\| = 0. \tag{45}$$

That is, every accumulation point of $\{\mathbf{y}_c\}$ *is a Z-eigenvector associated to a Z-eigenvalue* λ^*.

Proof Theorem 3 indicates that $\lim_{c \to \infty} f(\mathbf{y}_c) = \lambda^*$. Furthermore,

$$\lim_{c \to \infty} (f(\mathbf{y}_c) - f(\mathbf{y}_{c+1})) = 0. \tag{46}$$

According to the sufficient descent condition (12) and the Armijo condition (34), we have that $f(\mathbf{y}_{c+1}(\alpha_c)) - f(\mathbf{y}_c) \leq -\eta\rho\alpha_c\|\mathbf{g}_c\|^2$, i.e.,

$$f(\mathbf{y}_c) - f(\mathbf{y}_{c+1}(\alpha_c)) \geq \eta\rho\alpha_c\|\mathbf{g}_c\|^2. \tag{47}$$

From (46) and (47), it follows

$$\lim_{c\to\infty} \alpha_c\|\mathbf{g}_c\|^2 = 0. \tag{48}$$

Next, we prove (45) by contradiction. Assume that the conclusion (45) is not true, then there exists a constant $a > 0$ and a subsequence of $\{\mathbf{y}_c\}$, denoted by $\{\mathbf{y}_{c_j}\}$, such that $\|\mathbf{g}_{c_j}\| \geq a$. For clarity, we denote the index set of this subsequence as $\mathbb{J}$, it is deduced from (48) that the limit of $\{\alpha_{c_j}\}$ on $\mathbb{J}$ is 0.

From the choice of α_c in the second step of Algorithm 1 and the Armijo condition (34), we can find an integer number $N > 0$ such that for any $c_j \in \mathbb{J}$, $c_j \geq N$, it holds that

$$\frac{f(\mathbf{y}_{c_j}) - f\left(\mathbf{y}_{c_j+1}(\frac{\alpha_{c_j}}{\gamma})\right)}{\frac{\alpha_{c_j}}{\gamma}} < -\rho\mathbf{g}_{c_j}^T\mathbf{p}_{c_j}.$$

Note that $\mathbf{y}_{c_j} = \mathbf{y}_{c_{j+1}}(0)$, then by the mean value theorem, there exists $t \in (0, 1)$ satisfying

$$\left(\triangledown f\left(\mathbf{y}_{c_j+1}(t\frac{\alpha_{c_j}}{\gamma})\right)\right)^T \left(\frac{\mathbf{y}_{c_j} - \mathbf{y}_{c_j+1}(\frac{\alpha_{c_j}}{\gamma})}{\frac{\alpha_{c_j}}{\gamma}}\right) < -\rho\mathbf{g}_{c_j}^T\mathbf{p}_{c_j}. \tag{49}$$

Since the limit of $\{\alpha_{c_j}\}$ on $\mathbb{J}$ is 0, and $\mathbf{y}_{c_j}, \mathbf{y}_{c_j+1}(t\frac{\alpha_{c_j}}{\gamma}) \in \mathbb{S}^{n-2}$, where $\mathbb{S}^{n-2}$ is a compact set, there exists an index set $\tilde{\mathbb{J}} \subset \mathbb{J}$ such that $\{\mathbf{y}_{c_j+1}(t\frac{\alpha_{c_j}}{\gamma})\}_{c_j\in\tilde{\mathbb{J}}} \to \tilde{\mathbf{y}}^*$ and $\|\tilde{\mathbf{y}}^*\| = 1$. In a consequence, $\{\triangledown f\left(\mathbf{y}_{c_j+1}(t\frac{\alpha_{c_j}}{\gamma})\right)\}_{c_j\in\tilde{\mathbb{J}}} \to \triangledown f(\tilde{\mathbf{y}}^*) = \mathbf{g}(\tilde{\mathbf{y}}^*) \triangleq \tilde{\mathbf{g}}^*$. Additionally, it can be concluded from (12) and (20) that $\{\mathbf{p}_{c_j}\}_{c_j\in\tilde{\mathbb{J}}}$ is bounded. So there is another index set $\bar{\mathbb{J}} \subset \tilde{\mathbb{J}}$, such that $\{\mathbf{p}_{c_j}\}_{c_j\in\bar{\mathbb{J}}} \to \tilde{\mathbf{p}}^*$, and $\tilde{\mathbf{g}}^{*T}\tilde{\mathbf{p}}^* <0$. It follows from (13) that

$$\mathbf{y}_c - \mathbf{y}_{c+1}(\alpha) = 2\alpha\frac{\mathbf{y}_c^T\mathbf{p}_c + \alpha\|\mathbf{p}_c\|^2}{1 + 2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2}\mathbf{y}_c - 2\alpha\frac{1 + \alpha\mathbf{y}_c^T\mathbf{p}_c}{1 + 2\alpha\mathbf{y}_c^T\mathbf{p}_c + \alpha^2\|\mathbf{p}_c\|^2}\mathbf{p}_c, \quad \forall c \geq 1.$$

Taking limit of (49) on $\bar{\mathbb{J}}$, we obtain

$$\tilde{\mathbf{g}}^{*T}(2(\tilde{\mathbf{y}}^{*T}\tilde{\mathbf{p}}^*)\tilde{\mathbf{y}}^* - 2\tilde{\mathbf{p}}^*) < -\rho\tilde{\mathbf{g}}^{*T}\tilde{\mathbf{p}}^*.$$

Combining with $\tilde{\mathbf{g}}^{*T}\tilde{\mathbf{y}}^* = 0$, it follows

$$-\tilde{\mathbf{g}}^{*T}\tilde{\mathbf{p}}^* < -\frac{\rho}{2}\tilde{\mathbf{g}}^{*T}\tilde{\mathbf{p}}^*.$$

Furthermore, $0 < \rho < 2$ implies $\tilde{\mathbf{g}}^{*T}\tilde{\mathbf{p}}^* > 0$, this contradicts that $\tilde{\mathbf{p}}^*$ is a descent direction. Hence, the assumption is not correct, (45) holds.

Moreover, suppose $\mathbf{y}^*$ is an accumulation point of $\{\mathbf{y}_c\}$, then $\mathbf{y}^* \in \mathbb{S}^{n-2}$ and $\|\mathbf{g}(\mathbf{y}^*)\| = 0$. According to [8, Lemma 2], $\mathbf{y}^*$ is a Z-eigenvector associated to λ^*. □

5 Numerical tests on hypergraph partitioning and image segmentation

In this section, we conduct some experiments on hypergraph partitioning and image segmentation to illustrate the effectiveness of the proposed algorithm. All codes were written and performed by MATLAB R2014b on a Windows 10 laptop machine equipped with Intel(R) Core(TM) i7-10510U, 2.3 GHz, and 16 G RAM.

Since the smallest Z-eigenvalue problem that we solve was proposed by Chen et al. [9], we mainly compare our algorithm with that in [9], where trust region method and Cayley tramsform were employed.

We first provide the parameters for Algorithm 1: $\rho = 0.001$, $\bar{\sigma} = 0.25$, $\epsilon = 10^{-6}$, i.e., the stop criteria is $\|\mathbf{g}(\mathbf{y}_c)\| \leq 10^{-6}$, which is the same as that in [9]. The initial point $\mathbf{y}_1$ is also generated in the same way as that provided in [9], as follows: first, generate a random vector whose components have a standard Gaussian distribution, then normalize this vector to obtain $\mathbf{y}_1$.

Example 1 ([9]) For two intersecting circles shown in Fig. 1, whose centers are $(0, 0)$ and $(1, 0)$, respectively, and radii are both 1. Our task is to estimate centers and radii of the two intersecting circles on the condition that n ($n \geq 40$) points on the circles are known in advance.

The same way as presented in [9] is used to generate synthetic points and construct k-uniform weighted hypergraphs for this example. For the sake of completeness, we describe the process as below.

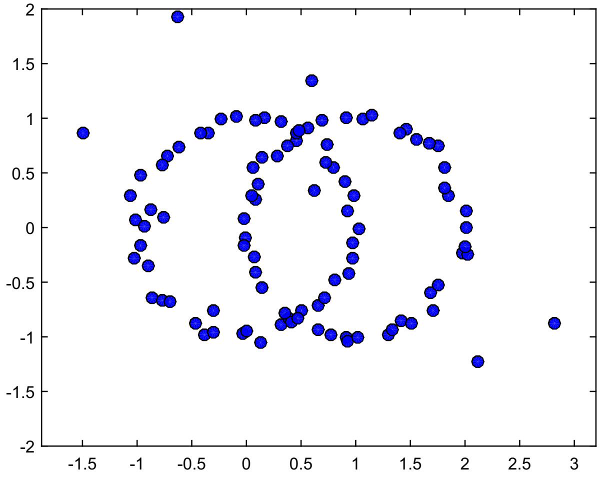
For a given n, which is the number of vertices, $[n \times 45\%]$ points from each circle are chosen, and then Gaussian white noise whose standard deviation is 0.05 is employed to corrupt coordinates of these points. The reminder $[n \times 10\%]$ points are outliers, which are randomly distributed in the rectangular area of $\{-2 \leq x \leq 3, -2 \leq y \leq 2\}$. Here, $[\cdot]$ indicates rounding operation. Figure 3 a and b illustrate the locations of synthetic points with $n = 100$ and $n = 400$ respectively.

In order to separate the sample points on the two two crossing circles, these points are treated as vertices to construct a k-uniform weighted hypergraph $G = (V, E, \mathbf{w})$, where $k \geq 4$ is even.

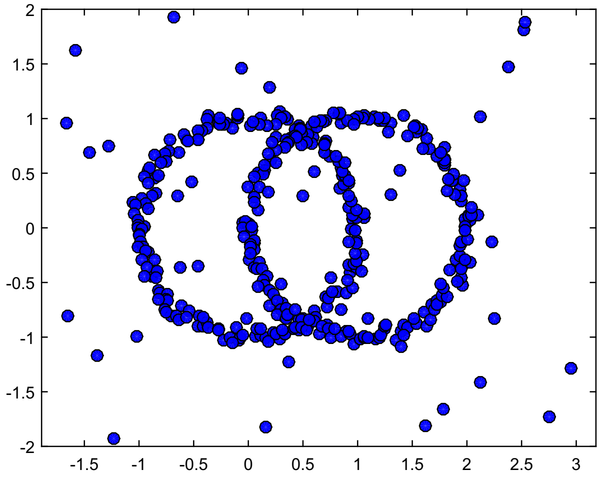
The process of constructing a hypergraph is as follows:

1. Construct a complete graph which is connected.
2. For each edge $\{i, j\}$ of the complete graph, add $k - 2$ random vertices which are sampled uniformly from vertex set $V \backslash \{i, j\}$.
3. Repeat the second step for $k - 1$ times.

By the above process, a k-uniform hypergraph with $m = (k-1)\binom{n}{2}$ edges can be obtained. For each edge e_p ($p = 1, 2, \ldots, m$) of the hypergraph, we fit locations of



(a) $n = 100$



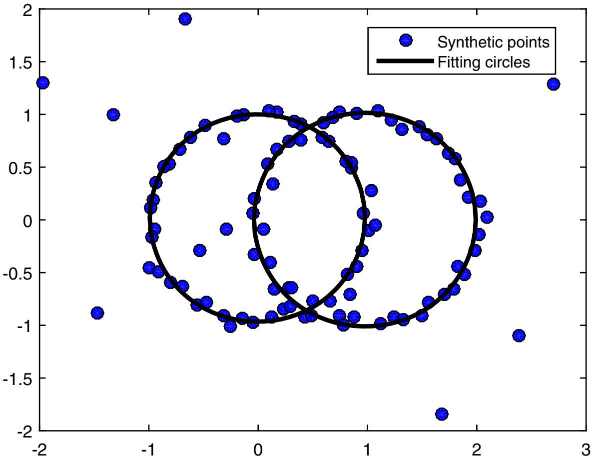
(b) $n = 400$

Fig. 3 Synthetic points for circle partitioning

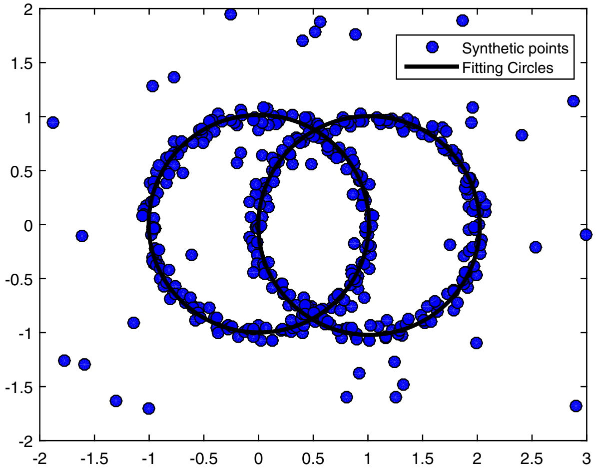
points in e_p onto a circle and denote the fitting error as r_p. Let s be the sample standard deviation of fitting errors $\{r_p\}_{p=1,\ldots,m}$. Then, the weight for an edge e_p is $w_p = e^{-\frac{r_p}{s}}$.

Then, we can use Algorithm 1 to compute the smallest Z-eigenpair of the compact Laplacian tensor of the hypergraph constructed above and further obtain its Fiedler vector, so that we can classify vertices into two groups with the Fiedler vector. When the partition of vertices is obtained, we estimate two circles using points from these two clusters, respectively. The estimated error of a circle is defined as the distance between true and estimated center plus the difference between true and estimated radius. The total estimated error is the mean of two estimated errors. In Fig. 4 a and b, we report the partitioning results when $n = 100$ and $n = 400$, respectively.

Next, we test Algorithm 1 and the algorithm in [9] on Example 1 and compare the efficiency of the two algorithms. We test 4-uniform undirected weighted hypergraphs where the number of vertices ranges from 50 to 800. It is worth noting that the synthetic points and the hypergraphs are different for each execution of the procedure; as a consequence, the partitioning results and estimated errors are not exactly the same. In order to reduce the impact on testing results, we run the two algorithms 100 times



(a) $k = 4$, $n = 100$

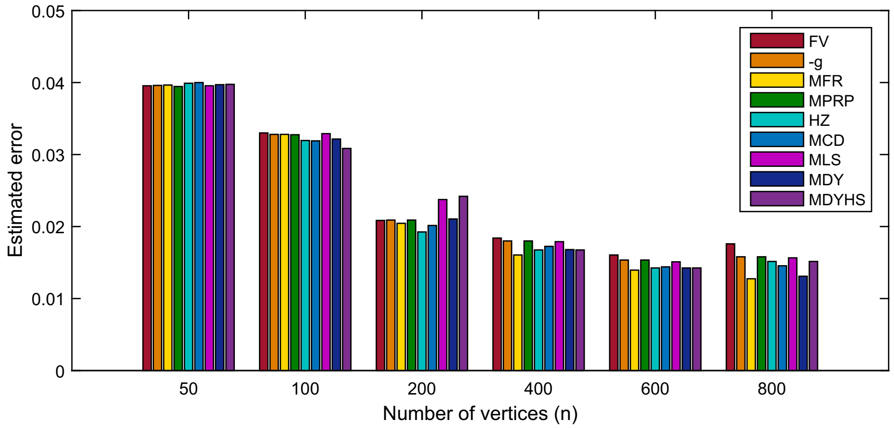


(b) $k = 4$, $n = 400$

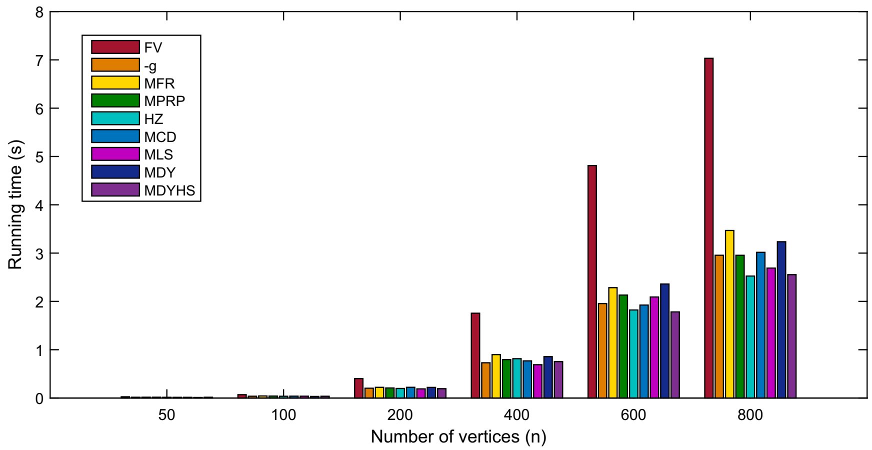
Fig. 4 Partitioning results on intersecting circles by Algorithm 1 for some certain k and n, then compute the mean of the estimated errors and running time. Details are reported in Fig. 5.

Remark 2 In Fig. 5, the legend "FV" represents the trust region algorithm in [9], "$-\mathbf{g}$" means the search direction $\mathbf{p}_c = -\mathbf{g}_c$ in Algorithm 1, "MFR," "MPRP," "MCD," "MLS," and "MDY" indicate that $\mathbf{p}_c$ is the modified CG direction proposed in this paper, which is computed by (21), (26) and (28). "HZ" computed by (21), (24), and (28) refers to the direction proposed in [28], and "MDYHS" implies $\mathbf{p}_c$ is the hybrid of MDY and MHS, which is determined by (21) and (27). The parameter $b = 1$ in formulae (24) and (26).

From Fig. 5a, we see that different algorithms or the same algorithm with different search directions can all work well for partitioning two intersecting circles. As the number of vertices increases, the estimated error of all methods decreases, but when



(a) Estimated errors v.s. number of vertices



(b) Running time v.s. number of vertices

Fig. 5 Variation of estimated errors and running time with the number of vertices

Table 1 Numerical results from Example 1

Direction	$n = 50$		$n = 100$		$n = 200$		$n = 400$		$n = 600$		$n = 800$	
	Iter	Time(s)	Iter	Time(s)	Iter	Time(s)	Iter	Time(s)	Iter	Time(s)	Iter	Time(s)
$-\mathbf{g}$	43.4	0.0164	33	0.0375	31	0.2032	29.5	0.7299	29	1.9558	30.5	2.9554
FR	41.9	0.0238	34.1	0.0730	28.6	0.2475	29.6	0.9045	28.6	2.5453	30.1	3.2393
MFR	42.9	0.0171	34.8	0.0433	31.1	0.2215	30.1	0.8989	28.5	2.2844	31	3.4682
PRP	43.9	0.0247	33.6	0.0736	31.1	0.2539	29.5	0.8444	28.8	2.1699	30.6	3.1742
MPRP	42	0.0172	32.9	0.0412	30.9	0.2086	29.5	0.7958	28.9	2.1317	30.5	2.9548
HS	52.7	0.0238	69.3	0.0899	58.4	0.4316	47.8	1.4609	48.3	3.3843	49.6	5.5481
HZ	41.1	0.0154	31.5	0.0369	28.6	0.1977	29.1	0.8142	25	1.8228	25.3	2.5249
CD	84.1	0.1745	84.4	0.5690	78.7	3.1682	78	11.949	75	33.375	72.7	44.042
MCD	30.8	0.0146	29.2	0.0389	28.5	0.2221	27.4	0.7694	25	1.9248	26.6	3.0160
LS	37.6	0.0268	31.6	0.0697	29.4	0.3379	30.2	1.2431	28.6	3.3439	28.6	4.7671
MLS	35.6	0.0149	28.3	0.0381	26.3	0.1895	26.4	0.6900	25.7	2.0918	26.3	2.6901
DY	63.8	0.0375	94.4	0.2287	105.4	1.6274	106	7.0143	108	18.492	92.6	21.161
MDY	29.1	0.0117	27	0.0328	28.5	0.2200	26.8	0.8566	25.7	2.3606	25.4	3.2354
DYHS	39.7	0.0145	37.6	0.0406	36.4	0.2159	33.5	0.7607	32.9	2.0310	32.5	2.8312
MDYHS	41.4	0.0162	31.5	0.0369	28.6	0.1936	29.6	0.7544	25.8	1.7827	25.8	2.5540

$n > 600$, the increase of n has no obvious effect on reducing the estimated error or even no reduction. For a fixed n, the estimated errors are relatively close when executing different algorithms or the same algorithm with different search directions, but there is a certain gap in the running time. Algorithm 1 with different search directions proposed in this paper have advantages in running time.

In order to further explore the impact of the modified CG directions and the classical ones on the efficiency of Algorithm 1, we present in Table 1 the average number of iterations (Iter.) and the average running time (Time(s)) required by Algorithm 1 with different directions. In Table 1, "FR," "PRP," "HS," "CD," "LS," and "DY" indicate the classical CG directions are adopted in Algorithm 1, which are computed by (21), (22), and (28). "DYHS" represents the hybrid of DY and HS, which is determined by (21) and (23). The remainder can be seen in Remark 2.

From Table 1, the efficiency of Algorithm 1 can be greatly improved when using the MCD and MDY directions, compared with the corresponding classical counterparts. The modified HZ and MLS schemes perform better than the classical ones. When the algorithm employs the formulae of MFR, MPRP, and MDYHS, there is not obvious difference from the classical counterparts. It is worthy noting that when the classical FR and PRP schemes are adopted in Algorithm 1, the ascent direction occurs frequently during the execution of the algorithm, and $-\mathbf{g}_c$ is used instead when this happens. As can be seen from this table, if $\mathbf{p}_c = -\mathbf{g}_c$, Algorithm 1 also performs well.

Example 2 Separate the foreground and background from the picture in Fig. 6.

Fig. 6 Original image

In this example, the process of constructing a uniform hypergraph is similar to that in Example 1, and the way of assigning weights is the same as that in [9]. We omit them here.

When the weighted hypergraph is obtained, we can use Algorithm 1 to compute the smallest Z-eigenpair of the compact Laplacian tensor and consequently get the Fiedler vector. According to the Fiedler vector, we can cluster the points and thus complete the segmentation of the image. The segmentation results are shown in Fig. 7 a and b.

Next, we give the numerical performance of Algorithm 1 and the algorithm in [9], which are both used to compute the smallest Z-eigenpair of the compact Laplacian tensor corresponding to the constructed hypergraph. In Table 2, "Iter." and "Time(s)" respectively refer to the number of iterations and the running time, and $\|\mathbf{g}(\mathbf{y}^*)\|$ means the norm of gradient when the algorithm terminates. "FV" represents the trust region algorithm in [9], where a subproblem is solved in each iteration, so we give the number

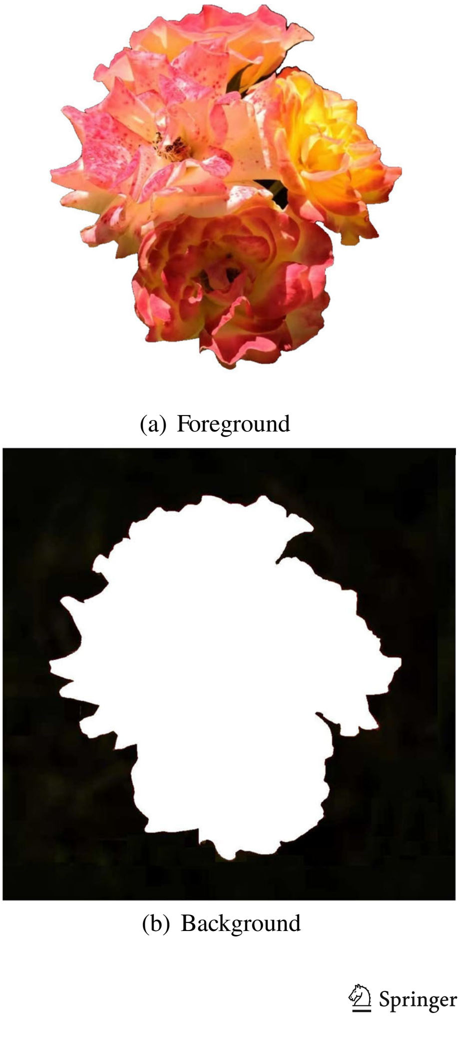


Fig. 7 Image segmentation by Fiedler vectors of 4-uniform hypergraphs

(a) Foreground

(b) Background

Table 2 Numerical results on Example 2

Method	Iter	Time(s)	$\|\mathbf{g}(\mathbf{y}^*)\|$	Method	Iter	Time(s)	$\|\mathbf{g}(\mathbf{y}^*)\|$
FV	11/104	0.1134	4.6664e−09	$-\mathbf{g}$	95	0.1052	8.1912e−07
FR	55	0.1104	9.2855e−07	MFR	33	0.0629	4.3845e−07
PRP	133	0.1274	2.6339e−07	MPRP	35	0.0612	3.0692e−07
HS	46	0.0858	7.8421e−07	HZ	33	0.0588	6.7830e−07
CD	79	0.4164	9.2915e−07	MCD	59	0.3172	9.6187e−07
LS	111	0.1348	2.6759e−07	MLS	27	0.0580	6.9250e−07
DY	31	0.0589	9.2295e−07	MDY	27	0.0497	8.5261e−07
DYHS	31	0.0600	3.9439e−07	MDYHS	25	0.0466	6.4465e−07

of both outer iteration and inner iteration for FV. The remainder are all the same as that in Example 1, and the parameter $b = 0.5$ in formulae (24) and (26).

Since the hypergraph constructed in this example has only 49 vertices, all methods can obtain the results in a short time. When the modified CG scheme is employed, both the number of iterations and running time perform better. The trust region algorithm in [9] need 11 outer iterations, and about 10 inner iterations are required in each outer iteration. When the outer iteration reaches the tenth times, the total number of inner iterations is 94, and $\|\mathbf{g}(\mathbf{y}_c)\| = 1.1011e - 06$ does not satisfy the stop criteria. When the algorithm terminates, the number of outer iteration increases 1, while the inner iterations increases 10, and $\|\mathbf{g}(\mathbf{y}^*)\|$ reaches $4.6664e - 09$. That is why $\|\mathbf{g}(\mathbf{y}^*)\|$ obtained by FV is less than that by other methods.

6 Conclusion

This paper proposed a family of constraint preserving gradient methods with applications to the hypergraph partitioning and image segmentation. We transformed the hypergraph partitioning problem into the problem of solving the smallest Z-eigenvalue of a hypergraph related tensor, which is constrained on a unit sphere. We used Householder transform to preserve the sphere constraint and a family of modified conjugate gradient directions as descent directions. We also proved that the convergence of the proposed algorithm. The numerical results illustrated that the efficiency of the proposed algorithm.

Acknowledgements The authors would like to thank Dr. Yannan Chen for his insightful discussions on hypergraph partitioning and image segmentation.

Author Contributions The first author prepared the mathematica programs of the presented algorithms, all authors wrote the main manuscript text and prepared the tables and figures, and all authors reviewed the manuscript.

Funding This paper was supported by Suqian Sci & Tech Program (Grant No. Z2020135 and K202112), the National Natural Science Foundation of China (Grant No. 11901118, 12001281 and 62073087), the Anhui Provincial Natural Science Foundation (Grant No. 2208085QA07), and the Youth Foundation of Anhui University of Technology (Grant No. QZ202114) and was sponsored by Qing Lan Project.

Availability of supporting data The authors confirm that all data generated or analyzed during this study are included in this paper.

Declarations

Ethics approval and consent to participate Not applicable

Conflict of interest The authors declare no competing interests.

References

1. Abdollahi, F., Fatemi, M.: A modified conjugate gradient method for general convex functions. Numer Algor. **92**, 1485–1502 (2023)
2. Bu, C., Fan, Y., Zhou, J.: Laplacian and signless Laplacian Z-eigenvalues of uniform hypergraphs. Front. Math. China. **11**, 511–520 (2016)
3. Bulò, S.R., Pelillo, M.: New bounds on the clique number of graphs based on spectral hypergraph theory. In: Stützle, T. (ed.) Learning and intelligent optimization, pp. 45–48. Springer Verlag, Berlin (2009)
4. Chang, J., Chen, Y., Qi, L.: Computing eigenvalues of large scale sparse tensors arising from a hypergraph. SIAM J. Sci. Comput. **38**(6), A3618–A3643 (2016)
5. Chang, J., Ding, W., Qi, L., Yan, H.: Computing the p-spectral radii of uniform hypergraphs with applications. J. Sci. Comput. **75**, 1–25 (2018)
6. Chang, J., Qi, L., Yan, H.: Hypergraph clustering using a new Laplacian tensor with applications in image processing. SIAM J. Imaging. Sci. **13**(3), 1157–1178 (2020)
7. Chen, L., Han, L., Zhou, L.: Computing tensor eigenvalues via homotopy methods. SIAM J. Matrix Anal. Appl. **37**, 290–319 (2016)
8. Chen, Y., Qi, L., Wang, Q.: Computing extreme eigenvalues of large scale Hankel tensors. J. Sci. Comput. **68**, 716–738 (2016)
9. Chen, Y., Qi, L., Zhang, X.: The Fiedler vector of a Laplacian tensor for hypergraph partitioning. SIAM J. Sci. Comput. **39**(6), A2508–A2537 (2017)
10. Chen, Z., Qi, L.: Circulant tensors with applications to spectral hypergraph theory and stochastic process. J. Ind. Manag. Optim. **12**, 1227–1247 (2016)
11. Chung, F.: Spectral graph theory. AMS, Providence RI (1997)
12. Chung, F., Lu, L.: Complex graphs and networks. AMS, Providence, RI (2006)
13. Cooper, J., Dutle, A.: Spectra of uniform hypergraphs. Linear Algebra Appl. **436**(9), 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.: New properties of a nonlinear conjugate gradient method. Numer. Math. **89**, 83–98 (2001)
16. Dai, Y., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**(1), 177–182 (1999)
17. Dai, Y., Yuan, Y.: Nonlinear conjugate gradient methods. Shanghai Science and Technology Press, Shanghai (2000)
18. 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)
19. Dorai, C., Jain, A.K.: Shape spectra based view grouping for free-form objects. In Proceedings of the International Conference on Image Processing, IEEE **3**, 340–343 (1995)
20. Fan, Y., Khan, M.I., Tan, Y.: The largest H-eigenvalue and spectral radius of Laplacian tensor of non-odd-bipartite generalized power hypergraphs. Linear Algebra Appl. **504**, 487–502 (2016)
21. Fiedler, M.: Algebraic connectivity of graphs. Czechoslovak Math. J. **23**, 298–305 (1973)
22. Fiedler, M.: A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czechoslovak Math. J. **25**, 619–633 (1973)
23. Fletcher, R.: Practical methods of optimization vol. 1: Unconstrained Optimization. John Wiley & Sons, New York (1987)
24. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)

25. Ghoshdastidar, D., Dukkipati, A.: A provable generalized tensor spectral method for uniform hypergraph partitioning. In Proceedings of the International Conference on Machine Learning 400–409 (2015)
26. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM J. Optim. **2**, 21–42 (1992)
27. Govindu, V.M., Van Loan, C.F.: A tensor decomposition for geometric grouping and segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) **1**, 1150–1157 (2005)
28. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**, 170–192 (2005)
29. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**(1), 35–58 (2006)
30. Han, J., Pei, J., Kamber, M.: Data mining: concepts and techniques. Elsevier, New York (2011)
31. 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)
32. Hestenes, M.R., Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Standards Sect. **5**(49), 409–436 (1952)
33. Hinneburg, A., Keim, D.A.: An efficient approach to clustering in large multimedia databases with noise. In Proceedings of KDD 58–65 (1998)
34. Hu, S., Qi, L.: Algebraic connectivity of an even uniform hypergraph. J. Comb. Optim. **24**, 564–579 (2012)
35. Hu, S., Qi, L.: The Laplacian of a uniform hypergraph. J. Comb. Optim. **29**(2), 331–366 (2015)
36. Huang, Z.: A fast clustering algorithm to cluster very large categorical data sets in data mining. Data Min. Knowl. Discov. **3**, 34–39 (1997)
37. Jiang, D., Wu, S., Chen, G., Ooi, B.C., Tan, K.-L., Xu, J.: epiC: an extensible and scalable system for processing Big Data. VLDB J. **25**, 3–26 (2016)
38. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**, 264–323 (1999)
39. Kang, S.H., Shafei, B., Steidl, G.: Supervised and transductive multi-class segmentation using p-Laplacians and RKHS methods. J. Vis. Commun. Image Represent. **25**(5), 1136–1148 (2014)
40. Klamt, S., Haus, U.-U., Theis, F.: Hypergraphs and cellular networks. PLoS Comput. Biol. **5**(5), e1000385 (2009)
41. Kofidis, E., Regalia, P.A.: On the best rank-1 approximation of higher-order supersymmetric tensors. SIAM J. Matrix Anal. Appl. **23**(3), 863–884 (2002)
42. Kolda, T.G., Mayo, J.R.: Shifted power method for computing tensor eigenpairs. SIAM J. Matrix Anal. Appl. **32**(4), 1095–1124 (2011)
43. Kolda, T.G., Mayo, J.R.: An adaptive shifted power method for computing generalized tensor eigenpairs. SIAM J. Matrix Anal. Appl. **35**(4), 1563–1581 (2014)
44. Li, G., Qi, L., Yu, G.: The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory. Numer. Linear Algebra Appl. **20**(6), 1001–1029 (2013)
45. 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)
46. Liu, L., Storey, C.: Efficient generalized conjugate gradient algorithms, Part 1: Theory. J. Optim. Theory Appl. **69**, 129–137 (1991)
47. Ni, Q.: Optimization methods and programming. Science Press, Beijing (2009)
48. Ni, Q., Qi, L.: A quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map. J. Glob. Optim. **61**, 627–641 (2015)
49. Polak, E., Ribière, G.: Note sur la convergence de directions conjugées. Rev. Francaise Informat. Recherche Opertionelle, 3e Année **16**, 35–43 (1969)
50. Polyak, B.T.: The conjugate gradient method in extreme problems. USSR Comp. Math. Math. Phys. **9**, 94–112 (1969)
51. Qi, L.: Eigenvalues of a real supersymmetric tensor. J. Symb. Comput. **40**(6), 1302–1324 (2005)
52. Qi, L.: H^+-eigenvalues of Laplacian and signless Lapaclian tensors. Comm. Math. Sci. **12**(6), 1045–1064 (2014)
53. Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., Ding, W., Lin, C.-T.: A review of clustering techniques and developments. Neurocomputing **267**, 664–681 (2017)

54. Xie, J., Chang, A.: A new type of Laplacian tensor and its Z-eigenvalues of an even uniform hypergraph. Int. J. Appl. Math. Stat. **31**, 9–19 (2013)
55. Xie, J., Qi, L.: The clique and coclique numbers' bounds based on the H-eigenvalues of uniform hypergraphs. Int. J. Numer. Anal, Mod. **12**(2), 318–327 (2015)
56. Xie, J., Qi, L.: Spectral directed hypergraph theory via tensors. Linear Multilinear Algebra **64**, 780–794 (2016)
57. Yu, G., Yu, Z., Xu, Y., et al.: An adaptive gradient method for computing generalized tensor eigenpairs. Comput. Optim. Appl. **65**, 781–797 (2016)
58. Yuan, G., Meng, Z., Li, Y.: A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations. J. Optim. Theory. Appl. **168**, 129–152 (2016)
59. Yuan, G., Wei, Z., Yang, Y.: The global convergence of the Polak-Ribière-Polyak conjugate gradient algorithm under inexact line search for nonconvex functions. J. Comput. Appl. Math. **362**, 262–275 (2019)
60. Yuan, G., Wang, X., Sheng, Z.: Family weak conjugate gradient algorithms and their convergence analysis for nonconvex functions. Numer Algor. **84**, 935–956 (2020)
61. Yuan, Y., Sun, W.: Optimization theory and methods. Science Press, Beijing (1999)
62. Zhang, X., Ni, Q., Ge, Z.: A convergent Newton algorithm for computing Z-eigenvalues of an almost nonnegative irreducible tensor. Optim. Method. Softw. **35**(2), 377–393 (2020)