

## ALL REAL EIGENVALUES OF SYMMETRIC TENSORS

CHUN-FENG CUI, YU-HONG DAI, AND JIAWANG NIE

ABSTRACT. This paper studies how to compute all real eigenvalues, associated to real eigenvectors, of a symmetric tensor. As is well known, the largest or smallest eigenvalue can be found by solving a polynomial optimization problem, while the other middle ones cannot. We propose a new approach for computing all real eigenvalues sequentially, from the largest to the smallest. It uses Jacobian semidefinite relaxations in polynomial optimization. We show that each eigenvalue can be computed by solving a finite hierarchy of semidefinite relaxations. Numerical experiments are presented to show how to do this.

## 1. INTRODUCTION

Let  $\mathbb{R}$  be the real field, and let  $m$  and  $n$  be positive integers. An  $n$ -dimensional tensor of order  $m$  is an array indexed by integer tuples  $(i_1, \dots, i_m)$  with  $1 \leq i_j \leq n$  ( $j = 1, \dots, m$ ). Let  $\mathbb{T}^m(\mathbb{R}^n)$  denote the space of all such real tensors. A tensor  $\mathcal{A} \in \mathbb{T}^m(\mathbb{R}^n)$  is indexed as

$$\mathcal{A} = (\mathcal{A}_{i_1 \dots i_m})_{1 \leq i_1, \dots, i_m \leq n}.$$

The tensor  $\mathcal{A}$  is *symmetric* if each entry  $\mathcal{A}_{i_1 \dots i_m}$  is invariant with respect to all permutations of  $(i_1, \dots, i_m)$ . Let  $\mathbb{S}^m(\mathbb{R}^n)$  be the space of all symmetric tensors in  $\mathbb{T}^m(\mathbb{R}^n)$ . For  $\mathcal{A} \in \mathbb{S}^m(\mathbb{R}^n)$ , we denote the polynomial

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

Clearly,  $\mathcal{A}x^m$  is a form (i.e., a homogenous polynomial) of degree  $m$  in  $x := (x_1, \dots, x_n)$ . For a positive integer  $k \leq m$ , denote

$$x^{[k]} := ((x_1)^k, \dots, (x_n)^k).$$

Define  $\mathcal{A}x^k$  to be the symmetric tensor in  $\mathbb{S}^{m-k}(\mathbb{R}^n)$  such that

$$(\mathcal{A}x^k)_{i_1, \dots, i_{m-k}} := \sum_{1 \leq j_1, \dots, j_k \leq n} \mathcal{A}_{i_1 \dots i_{m-k} j_1 \dots j_k} x_{j_1} \cdots x_{j_k}.$$

So,  $\mathcal{A}x^{m-1}$  is an  $n$ -dimensional vector.

An important property of symmetric tensors is their eigenvalues. Eigenvalues of tensors are introduced in Qi [28] and Lim [20]. Unlike matrices, there are various definitions of eigenvalues for tensors. Useful ones include H-eigenvalues, Z-eigenvalues (cf. [28]), and D-eigenvalues (cf. [32]). Eigenvalues of symmetric tensors have applications in signal processing (cf. [30]), diffusion tensor imaging (DTI) (cf. [4, 32, 33]), automatic control (cf. [22]), etc. The tensor eigenvalue

1991 *Mathematics Subject Classification.* 15A18, 15A69, 90C22.

*Key words and phrases.* symmetric tensors, eigenvalues of tensors, polynomial optimization, Lasserre's hierarchy, semidefinite relaxation.

problem is an important subject of multi-linear algebra. We refer to [14, 21, 29] for introductions to tensors and their applications.

Since there are various definitions of eigenvalues, we here give a unified approach to define them. It is a variation of the approach introduced in [3, 20, 28]. Let  $\mathbb{C}$  be the complex field.

**Definition 1.1.** *Let  $\mathcal{A} \in \mathcal{S}^m(\mathbb{R}^n)$  and  $\mathcal{B} \in \mathcal{S}^{m'}(\mathbb{R}^n)$  be two symmetric tensors (their orders  $m, m'$  are not necessarily equal). A number  $\lambda \in \mathbb{C}$  is a  $\mathcal{B}$ -eigenvalue of  $\mathcal{A}$  if there exists  $u \in \mathbb{C}^n$  such that*

$$(1.1) \quad \mathcal{A}u^{m-1} = \lambda \mathcal{B}u^{m'-1}, \quad \mathcal{B}u^{m'} = 1.$$

*Such  $u$  is called a  $\mathcal{B}$ -eigenvector associated to  $\lambda$ , and such  $(\lambda, u)$  is called a  $\mathcal{B}$ -eigenpair of  $\mathcal{A}$ .*

For cleanness of the paper, when the tensor  $\mathcal{B}$  is clear in the context,  $\mathcal{B}$ -eigenvalues (resp.,  $\mathcal{B}$ -eigenvectors,  $\mathcal{B}$ -eigenpairs) are just simply called eigenvalues (resp., eigenvectors, eigenpairs). When an eigenvalue  $\lambda$  is real, it may not have a real eigenvector  $u$ . An eigenpair  $(\lambda, u)$  is called *real* if both  $\lambda$  and  $u$  are real. Throughout the paper, for convenience, we call that  $\lambda$  is a real eigenvalue if  $\lambda$  is real and it has a real eigenvector. By the largest (resp., smallest) eigenvalue, we mean the largest (resp., smallest) real eigenvalue. In the paper, we only discuss how to compute real eigenvalues.

The following special cases of  $\mathcal{B}$ -eigenvalues are well known.

- When  $m' = m$  and  $\mathcal{B}$  is the identity tensor (i.e.,  $\mathcal{B}x^m = x_1^m + \cdots + x_n^m$ ), the  $\mathcal{B}$ -eigenvalues are just the H-eigenvalues (cf. [28]). When  $m$  is even, a number  $\lambda$  is a real H-eigenvalue of  $\mathcal{A}$  if there exists  $u \in \mathbb{R}^n$  such that

$$\mathcal{A}u^{m-1} = \lambda u^{[m-1]}, \quad u_1^m + \cdots + u_n^m = 1.$$

Such  $(\lambda, u)$  is called an H-eigenpair.

- When  $m' = 2$  and  $\mathcal{B}$  is such that  $\mathcal{B}x^2 = x_1^2 + \cdots + x_n^2$ , the  $\mathcal{B}$ -eigenvalues are just the Z-eigenvalues (cf. [28]). Equivalently, a number  $\lambda$  is a real Z-eigenvalue if there exists  $u \in \mathbb{R}^n$  such that

$$\mathcal{A}u^{m-1} = \lambda u, \quad u_1^2 + \cdots + u_n^2 = 1.$$

Such  $(\lambda, u)$  is called a Z-eigenpair.

- Let  $D \in \mathbb{R}^{n \times n}$  be a symmetric positive definite matrix. When  $m' = 2$  and  $\mathcal{B}$  is such that  $\mathcal{B}x^2 = x^T D x$ , the  $\mathcal{B}$ -eigenvalues are just the D-eigenvalues (cf. [32]). Equivalently, a number  $\lambda$  is a real D-eigenvalue if there exists  $u \in \mathbb{R}^n$  such that

$$\mathcal{A}u^{m-1} = \lambda D u, \quad u^T D u = 1.$$

Such  $(\lambda, u)$  is called a D-eigenpair.

The problem of computing eigenvalues of higher order tensors (i.e.,  $m \geq 3$ ) is NP-hard (cf. [12]). Recently, there exists much work for computing the largest (or smallest) eigenvalues of symmetric tensors. Qi et al. [31] proposed an elimination method for computing the largest Z-eigenvalue when  $n = 2$  and  $m = 3$ . Hu et al. [13] used a sequence of semidefinite relaxations for computing extreme Z-eigenvalues. Kolda et al. [15] presented a shifted power method for computing Z-eigenvalues. Zhang et al. [38] proposed a modified power method. Han [8] introduced an unconstrained optimization method for even order symmetric tensors.

Hao et al. [9] presented a sequential subspace projection method for computing extreme Z-eigenvalues.

The existing methods are mostly for computing the largest or smallest eigenvalues. However, there are very few methods for computing the other middle eigenvalues. Computing the second or other largest eigenvalues for symmetric tensors is also an important problem in some applications. In DTI [4, 33], the three largest Z-eigenvalues of a diffusion tensor describe the diffusion coefficients in different directions. As shown by Li et al. [19], the second largest Z-eigenvalue for the characteristic tensor of a hypergraph can be used to get a lower bound for its bipartition width.

The main goal of this paper is to compute all real eigenvalues of a symmetric tensor. For  $\mathcal{A} \in \mathcal{S}^m(\mathbb{R}^n)$ ,  $\mathcal{B} \in \mathcal{S}^{m'}(\mathbb{R}^n)$ , it holds that

$$\nabla \mathcal{A}x^m = m \mathcal{A}x^{m-1}, \quad \nabla \mathcal{B}x^{m'} = m' \mathcal{B}x^{m'-1}.$$

Here, the symbol  $\nabla$  denotes the gradient in  $x$ . Thus, (1.1) is equivalent to

$$\frac{1}{m} \nabla \mathcal{A}u^m = \frac{1}{m'} \lambda \nabla \mathcal{B}u^{m'}, \quad \mathcal{B}u^{m'} = 1.$$

Then,  $(\lambda, u)$  is a  $\mathcal{B}$ -eigenpair if and only if  $u$  is a critical point of the problem

$$(1.2) \quad \max \mathcal{A}x^m \quad \text{s.t.} \quad \mathcal{B}x^{m'} = 1.$$

Moreover, the critical value associated to  $u$  is  $\lambda$ , because

$$u^T \nabla \mathcal{A}u^m = m \mathcal{A}u^m, \quad u^T \nabla \mathcal{B}u^{m'} = m' \mathcal{B}u^{m'}.$$

This shows that  $(\lambda, u)$  is a  $\mathcal{B}$ -eigenpair if and only if  $u$  is a critical point of (1.2) with the critical value  $\lambda$ . The polynomial optimization problem (1.2) has finitely many critical values (cf. [26]), including both complex and real ones. That is, every symmetric tensor  $\mathcal{A}$  has finitely many complex and real  $\mathcal{B}$ -eigenvalues. We order the real  $\mathcal{B}$ -eigenvalues monotonically as  $\lambda_1 > \lambda_2 > \dots > \lambda_K$ . For convenience, denote  $\lambda_{\max} := \lambda_1$  and  $\lambda_{\min} := \lambda_K$ .

In this paper, we study how to compute all real eigenvalues. Mathematically, this is equivalent to finding all the real critical values of (1.2), which is a polynomial optimization problem. The semidefinite relaxation method by Lasserre [16] can be applied to get the largest or smallest eigenvalue. To get other middle eigenvalues, we need to use new techniques. Recently, Nie [26] proposed a method for computing the hierarchy of local minimums in polynomial optimization, which uses the Jacobian SDP relaxation method from [24]. We mainly follow the approach in [26] to compute all real eigenvalues sequentially. Indeed, by this approach, each real eigenvalue can be obtained by solving a finite hierarchy of semidefinite relaxations. This is an attractive property that most other numerical methods do not have.

The paper is organized as follows. In §2, we present some preliminaries in polynomial optimization. In §3, we propose semidefinite relaxations for computing all real eigenvalues sequentially. In §4, we report extensive numerical examples to show how to compute all real eigenvalues.

## 2. PRELIMINARIES

This section reviews some basics in polynomial optimization. We refer to [5, 17, 18] for details.

Denote by  $\mathbb{R}[x] := \mathbb{R}[x_1, \dots, x_n]$  the ring of polynomials in  $x := (x_1, \dots, x_n)$  with real coefficients. For a degree  $d$ ,  $\mathbb{R}[x]_d$  denotes the space of all polynomials in  $\mathbb{R}[x]$  whose degrees are at most  $d$ . The dimension of the space  $\mathbb{R}[x]_d$  is  $\binom{n+d}{d}$ . An ideal of  $\mathbb{R}[x]$  is a subset  $J$  of  $\mathbb{R}[x]$  such that  $J \cdot \mathbb{R}[x] \subseteq J$  and  $J + J \subseteq J$ . For a tuple  $\phi := (\phi_1, \dots, \phi_r)$  of polynomials in  $\mathbb{R}[x]$ , the ideal generated by  $\phi$  is the smallest ideal containing all  $\phi_i$ , which is the set  $\phi_1 \cdot \mathbb{R}[x] + \dots + \phi_r \cdot \mathbb{R}[x]$  and is denoted by  $I(\phi)$ . The set

$$I_k(\phi) := \phi_1 \cdot \mathbb{R}[x]_{k-\deg(\phi_1)} + \dots + \phi_r \cdot \mathbb{R}[x]_{k-\deg(\phi_r)}$$

is called the  $k$ -th truncation of the ideal  $I(\phi)$ . Clearly,

$$\bigcup_{k \in \mathbb{N}} I_k(\phi) = I(\phi).$$

A polynomial  $\sigma \in \mathbb{R}[x]$  is called a sum of squares (SOS) if there exist  $p_1, \dots, p_k \in \mathbb{R}[x]$  such that  $\sigma = p_1^2 + \dots + p_k^2$ . Let  $\Sigma[x]$  be the set of all SOS polynomials and

$$\Sigma[x]_m := \Sigma[x] \cap \mathbb{R}[x]_m.$$

Both  $\Sigma[x]$  and  $\Sigma[x]_m$  are convex cones. As is well known, each SOS polynomial is nonnegative everywhere, while the reverse is not necessarily true. We refer to [34] for a survey on SOS and nonnegative polynomials. Let  $\psi := (\psi_1, \dots, \psi_t)$  be a tuple of polynomials in  $\mathbb{R}[x]$ . The set

$$Q_N(\psi) := \Sigma[x]_{2N} + \psi_1 \cdot \Sigma[x]_{2N-\deg(\psi_1)} + \dots + \psi_t \cdot \Sigma[x]_{2N-\deg(\psi_t)}$$

is called the  $N$ -th truncation of the quadratic module generated by  $\psi$ . The union

$$Q(\psi) := \bigcup_{N \in \mathbb{N}} Q_N(\psi)$$

is called the quadratic module generated by  $\psi$ .

Let  $\mathbb{N}$  be the set of nonnegative integers. For  $x := (x_1, \dots, x_n)$  and  $\alpha := (\alpha_1, \dots, \alpha_n)$ , denote  $x^\alpha := x_1^{\alpha_1} \dots x_n^{\alpha_n}$  and  $|\alpha| := \alpha_1 + \dots + \alpha_n$ . For  $d \in \mathbb{N}$ , denote

$$\mathbb{N}_d^n := \{\alpha \in \mathbb{N}^n : |\alpha| \leq d\}.$$

The space dual to  $\mathbb{R}[x]_d$  is the set of all *truncated multi-sequences* (tms') of degree  $d$ , which is denoted by  $\mathbb{R}_d^{\mathbb{N}_d^n}$ . A vector  $y$  in  $\mathbb{R}_d^{\mathbb{N}_d^n}$  is indexed by  $\alpha \in \mathbb{N}_d^n$ , i.e.,

$$y = (y_\alpha)_{\alpha \in \mathbb{N}_d^n}.$$

Each  $y \in \mathbb{R}_d^{\mathbb{N}_d^n}$  defines the linear functional  $\mathcal{L}_y$  acting on  $\mathbb{R}[x]_d$  as

$$\mathcal{L}_y(x^\alpha) = y_\alpha, \quad \forall \alpha \in \mathbb{N}_d^n.$$

Let  $q \in \mathbb{R}[x]_{2k}$ . For each  $y \in \mathbb{R}_k^{\mathbb{N}_{2k}^n}$ , the function  $\mathcal{L}_y(qp^2)$  is a quadratic form in  $\text{vec}(p)$ , the coefficient vector of polynomial  $p$  with  $\deg(qp^2) \leq 2k$ . Let  $L_q^{(k)}(y)$  be the symmetric matrix such that

$$\mathcal{L}_y(qp^2) = \text{vec}(p)^\top \left( L_q^{(k)}(y) \right) \text{vec}(p).$$

The matrix  $L_q^{(k)}(y)$  is called the  $k$ -th localizing matrix of  $q$  generated by  $y$ . It is linear in  $y$ . For instance, when  $n = 2$ ,  $k = 2$  and  $q = 1 - x_1^2 - x_2^2$ , we have

$$L_{1-x_1^2-x_2^2}^{(2)}(y) = \begin{pmatrix} y_{00} - y_{20} - y_{02} & y_{10} - y_{30} - y_{12} & y_{01} - y_{21} - y_{03} \\ y_{10} - y_{30} - y_{12} & y_{20} - y_{40} - y_{22} & y_{11} - y_{31} - y_{13} \\ y_{01} - y_{21} - y_{03} & y_{11} - y_{31} - y_{13} & y_{02} - y_{22} - y_{04} \end{pmatrix}.$$

When  $q = 1$  (i.e., the constant one polynomial),  $L_1^{(k)}(y)$  is called the  $k$ -th moment matrix generated by  $y$ , and is denoted as  $M_k(y)$ . For instance, when  $n = 2$  and  $k = 2$ ,

$$M_2(y) = \begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{pmatrix}.$$

### 3. SEMIDEFINITE RELAXATIONS FOR COMPUTING ALL REAL EIGENVALUES

In this section, we show how to compute all real eigenvalues sequentially. The Jacobian SDP relaxation technique in [24] is a useful tool for this purpose.

Let  $\mathcal{A} \in \mathbf{S}^m(\mathbb{R}^n)$  and  $\mathcal{B} \in \mathbf{S}^{m'}(\mathbb{R}^n)$ . For convenience, denote  $f(x) := \mathcal{A}x^m$  and  $g(x) := \mathcal{B}x^{m'} - 1$ . Then (1.2) is the same as

$$(3.1) \quad \max f(x) \quad \text{s.t.} \quad g(x) = 0.$$

In the introduction, we have seen that  $(\lambda, u)$  is a  $\mathcal{B}$ -eigenpair if and only if  $\lambda$  is a critical value of (3.1), and  $u$  is an associated critical point. The problem (3.1) always has finitely many critical values (cf. [26]), including both complex and real ones. So,  $\mathcal{A}$  has finitely many complex and real eigenvalues. We order the real eigenvalues monotonically as

$$\lambda_1 > \lambda_2 > \cdots > \lambda_K,$$

where  $K$  is the total number of distinct real eigenvalues. Denote

$$\mathcal{W} := \{x \in \mathbb{R}^n \mid \text{rank}[\nabla f(x) \quad \nabla g(x)] \leq 1\}.$$

Clearly, if  $(\lambda, u)$  is a real  $\mathcal{B}$ -eigenpair of  $\mathcal{A}$ , then  $u \in \mathcal{W}$ . The description of the set  $\mathcal{W}$  does not use the Lagrange multiplier. This is an advantage in computation. Suppose  $g(x) = 0$  is a smooth real hypersurface (i.e.,  $\nabla g(x) \neq 0$  for all real points on  $g(x) = 0$ ). It follows from Definition 1.1 that any  $u \in \mathcal{W}$  satisfying  $g(u) = 0$  is a  $\mathcal{B}$ -eigenvector of  $\mathcal{A}$ , associated to the eigenvalue  $\lambda = f(u)$ . For the frequently used Z-eigenvalues (i.e.,  $g(x) = x^T x - 1$ ) and H-eigenvalues (i.e.,  $g(x) = x_1^m + \cdots + x_n^m - 1$ ), the hypersurface  $g(x) = 0$  is smooth.

A point  $u$  belongs to  $\mathcal{W}$  if and only if

$$f_{x_i}(u)g_{x_j}(u) - f_{x_j}(u)g_{x_i}(u) = 0 \quad (1 \leq i < j \leq n),$$

where  $f_{x_i} = \frac{\partial}{\partial x_i} f(x)$  and  $g_{x_i} = \frac{\partial}{\partial x_i} g(x)$ . There are totally  $\frac{1}{2}n(n-1)$  equations. Indeed, the number of defining equations for  $\mathcal{W}$  can be dropped to  $2n-3$  (cf. [1, Chap. 5]). It suffices to use the following  $2n-3$  equations (cf. [1, 24]):

$$(3.2) \quad h_r := \sum_{i+j=r+2} (f_{x_i}g_{x_j} - f_{x_j}g_{x_i}) = 0 \quad (r = 1, \dots, 2n-3).$$

For convenience, let  $h_{2n-2} := g$  and

$$(3.3) \quad h := (h_1, \dots, h_{2n-2}).$$

Clearly, (3.1) is equivalent to the maximization problem

$$(3.4) \quad \max f(x) \quad \text{s.t.} \quad h_r(x) = 0 \quad (r = 1, \dots, 2n-2).$$

When the real hypersurface  $g(x) = 0$  is smooth, a point  $u$  is feasible for (3.4) if and only if  $u$  is a critical point of (3.1), i.e.,  $u$  is a  $\mathcal{B}$ -eigenvector. This implies that the objective value of (3.4) at any feasible point is a  $\mathcal{B}$ -eigenvalue of  $\mathcal{A}$ . Thus, the objective values on feasible points are  $\lambda_1, \dots, \lambda_K$ .

In the following, we show how to compute all real eigenvalues sequentially. That is, we compute  $\lambda_1$  first, then  $\lambda_2$  second, and then  $\lambda_3, \dots$  if they exist.

**3.1. The largest eigenvalue.** The largest eigenvalue  $\lambda_1$  is the maximum value of problem (3.4). Write the polynomial  $f(x) = \mathcal{A}x^m$  as

$$f(x) = \sum_{\alpha \in \mathbb{N}^n: |\alpha|=m} f_\alpha x^\alpha.$$

For a tms  $y \in \mathbb{R}^{\mathbb{N}_{2N}^n}$  with degree  $2N \geq m$ , denote

$$\langle f, y \rangle := \sum_{\alpha \in \mathbb{N}^n: |\alpha|=m} f_\alpha y_\alpha.$$

Clearly,  $\langle f, y \rangle$  is a linear function in  $y$ . Denote

$$N_0 := \lceil (m + m' - 2)/2 \rceil.$$

Lasserre's hierarchy of semidefinite relaxations (cf. [16]) for solving (3.4) is ( $N = N_0, N_0 + 1, \dots$ )

$$(3.5) \quad \begin{cases} \rho_N^{(1)} := \max & \langle f, y \rangle \\ \text{s.t.} & L_{h_r}^{(N)}(y) = 0 \quad (r = 1, \dots, 2n - 2), \\ & y_0 = 1, M_N(y) \succeq 0. \end{cases}$$

Let  $h$  be the tuple as in (3.3). The dual problem of (3.5) is then

$$(3.6) \quad \eta_N^{(1)} := \min \quad \gamma \quad \text{s.t.} \quad \gamma - f \in I_{2N}(h) + \Sigma[x]_{2N}.$$

It can be shown that the optimal values  $\rho_N^{(1)}, \eta_N^{(1)}$  are upper bounds for  $\lambda_1$ . Both sequences  $\{\rho_N^{(1)}\}$  and  $\{\eta_N^{(1)}\}$  are monotonically decreasing. That is

$$\begin{aligned} \rho_{N_0}^{(1)} &\geq \rho_{N_0+1}^{(1)} \geq \dots \geq \rho_N^{(1)} \geq \dots \geq \lambda_1, \\ \eta_{N_0}^{(1)} &\geq \eta_{N_0+1}^{(1)} \geq \dots \geq \eta_N^{(1)} \geq \dots \geq \lambda_1. \end{aligned}$$

By the weak duality, we also have

$$\rho_N^{(1)} \leq \eta_N^{(1)} \quad (N = N_0, N_0 + 1, \dots).$$

In fact, they both have the nice property of converging to  $\lambda_1$  in finitely many steps, i.e.,  $\rho_N^{(1)} = \eta_N^{(1)} = \lambda_1$  for all  $N$  large enough.

**Theorem 3.1.** *Let  $\mathcal{A} \in \mathcal{S}^m(\mathbb{R}^n)$  and  $\mathcal{B} \in \mathcal{S}^{m'}(\mathbb{R}^n)$ . Suppose the real hypersurface  $\mathcal{B}x^{m'} = 1$  is smooth. Let  $\lambda_1$  be the largest real  $\mathcal{B}$ -eigenvalue of  $\mathcal{A}$ . Then, we have the following properties:*

- (i) *It holds that  $\rho_N^{(1)} = \eta_N^{(1)} = \lambda_1$  for all  $N$  large enough.*
- (ii) *Suppose  $\lambda_1$  has finitely many real eigenvectors on  $\mathcal{B}x^{m'} = 1$ . If  $N$  is large enough, then, for every optimizer  $y^*$  of (3.5), there exists an integer  $t \leq N$  such that*

$$(3.7) \quad \text{rank } M_{t-N_0}(y^*) = \text{rank } M_t(y^*).$$

*Proof.* Note that  $-\lambda_1$  is the minimum value of

$$\min -f(x) \quad \text{s.t.} \quad g(x) = 0.$$

The polynomials  $h_1, \dots, h_{2n-3}$  are constructed by using Jacobian SDP relaxations in [24]. The relaxations (3.2), (3.4), (3.5)-(3.6) are specializations of the semidefinite relaxations (4.5), (4.6), (4.7)-(4.8) constructed in [26]. Thus, the items (i)-(ii) can be implied by Theorem 4.1 of [26].  $\square$

In computation, a practical issue is how to determine whether  $\rho_N^{(1)} = \eta_N^{(1)} = \lambda_1$  or not, because  $\lambda_1$  is typically unknown. This can be done by checking the rank condition (3.7). If it is satisfied, then we can get

$$\ell := \text{rank } M_t(y^*)$$

distinct feasible points  $u_1, \dots, u_\ell$  of (3.4), such that each  $u_i$  is a maximizer of (3.4) and  $f(u_i) = \rho_N^{(1)} = \eta_N^{(1)} = \lambda_1$ . They can be computed by the method in Henrion and Lasserre [10]. In other words, if (3.7) holds, then  $\rho_N^{(1)} = \eta_N^{(1)} = \lambda_1$ , and such  $u_1, \dots, u_\ell$  are the associated  $\mathcal{B}$ -eigenvectors. So, by solving (3.5)-(3.6), we not only get the largest eigenvalue  $\lambda_1$ , but also its  $\mathcal{B}$ -eigenvectors. As shown in Theorem 3.1 (ii), if there are finitely many real  $\mathcal{B}$ -eigenvectors (this is the general case, cf. [2]), then (3.7) must be satisfied. So, (3.7) is generally sufficient and necessary for checking convergence of semidefinite relaxations (3.5)-(3.6). The rank condition (3.7) is called flatness. It is a very useful tool for solving truncated moment problems (cf. Curto and Fialkow [6]). The software `GloptiPoly 3` (cf. [11]) can be applied to solve the semidefinite relaxations (3.5)-(3.6).

In Theorem 3.1, the relaxations (3.5)-(3.6) are assumed to be solved exactly. However, in practice, they are often solved approximately, due to round-off errors. Suppose  $\tilde{\rho}_N^{(1)}, \tilde{\eta}_N^{(1)}$  are numerically computed optimal values of (3.5)-(3.6), respectively. Then  $\tilde{\rho}_N^{(1)} = \tilde{\eta}_N^{(1)} = \lambda_1$  may not hold exactly, but they are approximately true. The errors depend on the accuracy of solving (3.5)-(3.6). We refer to Chapter 7 of the book [36] for error analysis in semidefinite programming. When approximately optimal solutions of (3.5)-(3.6) are computed, the rank condition (3.7) will be satisfied approximately. This issue was discussed in [25, §3].

**Remark 3.2.** *Suppose the rank condition (3.7) is satisfied. If  $\text{rank } M_N(y^*)$  is maximum among the set of all optimizers of (3.5), then we can get all maximizers of (3.4) (cf. [18, §6.6]). In such case, we can get all the  $\mathcal{B}$ -eigenvectors associated to  $\lambda_1$ . Therefore, when (3.5)-(3.6) are solved by primal-dual interior point methods, typically we can get all the  $\mathcal{B}$ -eigenvectors associated to  $\lambda_1$  (cf. [26]). However, if there are infinitely many  $\mathcal{B}$ -eigenvectors lying on  $\mathcal{B}x^{m'} = 1$ , (3.7) is typically not satisfied.*

To check the condition (3.7), we need to evaluate the ranks of matrices  $M_{t-N_0}(y^*)$  and  $M_t(y^*)$ . In numerical computation, sometimes this would be a very difficult issue because of round-off errors. The rank of a matrix equals to the number of its positive singular values. In practice, we can evaluate the rank as the number of singular values bigger than a tolerance (say,  $10^{-6}$ ). By this way, if there is a sufficiently small perturbation on a matrix, its evaluated rank will not change. We refer to the book [7] for evaluating matrix ranks numerically.

**3.2. The second and other largest eigenvalues.** Suppose the  $k$ -th largest eigenvalue  $\lambda_k$  of  $\mathcal{A}$  is known. We want to compute the  $(k+1)$ -th largest eigenvalue  $\lambda_{k+1}$ , if it exists. Let  $\delta \in \mathbb{R}$  be such that

$$(3.8) \quad 0 < \delta < \lambda_k - \lambda_{k+1}.$$

Consider the optimization problem

$$(3.9) \quad \begin{cases} \max & f(x) \\ \text{s.t.} & h_r(x) = 0 \quad (r = 1, \dots, 2n-2), \\ & f(x) \leq \lambda_k - \delta. \end{cases}$$

When (3.8) is satisfied, the optimal value of (3.9) is  $\lambda_{k+1}$ . Lasserre's hierarchy of semidefinite relaxations for solving (3.9) is  $(N = N_0, N_0 + 1, \dots)$

$$(3.10) \quad \begin{cases} \rho_N^{(k+1)} := \max & \langle f, y \rangle \\ \text{s.t.} & L_{h_r}^{(N)}(y) = 0 \quad (r = 1, \dots, 2n-2), \\ & y_0 = 1, L_{\lambda_k - \delta - f}^{(N)}(y) \succeq 0, M_N(y) \succeq 0. \end{cases}$$

Its dual problem is then

$$(3.11) \quad \eta_N^{(k+1)} := \min \quad \gamma \quad \text{s.t.} \quad \gamma - f \in I_{2N}(h) + Q_N(\lambda_k - \delta - f).$$

Semidefinite relaxations (3.10)-(3.11) have the following properties:

**Theorem 3.3.** *Suppose the real hypersurface  $\mathcal{B}x^{m'} = 1$  is smooth. Let  $\lambda_k$  (resp.,  $\lambda_{k+1}$ ) be the  $k$ -th (resp.,  $(k+1)$ -th) largest  $\mathcal{B}$ -eigenvalue of  $\mathcal{A}$ . For all  $\delta$  satisfying (3.8), we have the following properties:*

- (i) *For all  $N$  big enough, we have  $\rho_N^{(k+1)} = \eta_N^{(k+1)} = \lambda_{k+1}$ .*
- (ii) *Suppose  $\lambda_{k+1}$  has finitely many eigenvectors on  $\mathcal{B}x^{m'} = 1$ . If  $N$  is large enough, then for every optimizer  $y^*$  of (3.10), there exists an integer  $t \leq N$  such that (3.7) holds.*

*Proof.* Note that  $-\lambda_k$  is the  $k$ -th smallest critical value of

$$\min \quad -f(x) \quad \text{s.t.} \quad g(x) = 0.$$

The polynomials  $h_1, \dots, h_{2n-3}$  are constructed by using Jacobian SDP relaxations in [24]. The semidefinite relaxations (3.9)-(3.11) are specializations of (4.9)-(4.11) in [26]. Thus, the items (i)-(ii) can be obtained by Theorem 4.3 of [26].  $\square$

**Remark 3.4.** *The finite convergence of  $\rho_N^{(k+1)}$  and  $\eta_N^{(k+1)}$  to  $\lambda_{k+1}$  can be identified by checking the rank condition (3.7). If it is satisfied, we can get  $\ell$   $\mathcal{B}$ -eigenvectors associated to  $\lambda_{k+1}$ . When the semidefinite relaxations (3.10) and (3.11) are solved by primal-dual interior point methods, typically we can get all  $\mathcal{B}$ -eigenvectors, provided there are finitely many ones. The rank condition (3.7) is generally sufficient and necessary for checking the finite convergence of the sequences  $\{\rho_N^{(k+1)}\}$  and  $\{\eta_N^{(k+1)}\}$ . We refer to Remark 3.2. We also refer to the discussions before and after Remark 3.2, about the numerical issues related to (3.7), (3.10) and (3.11).*

In practice, we usually do not know whether  $\lambda_{k+1}$  exists or not. Even if it exists, we do not know how small  $\delta$  should be chosen to satisfy (3.8). Interestingly, this

issue can be fixed by solving the optimization problem

$$(3.12) \quad \begin{cases} \chi_k := \min & f(x) \\ \text{s.t.} & h_r(x) = 0 \ (r = 1, \dots, 2n-2), \\ & f(x) \geq \lambda_k - \delta. \end{cases}$$

The following proposition is useful.

**Proposition 3.5.** *Suppose the real hypersurface  $\mathcal{B}x^{m'} = 1$  is smooth. Let  $\lambda_k$  (resp.,  $\lambda_{min}$ ) be the  $k$ -th largest (resp., smallest)  $\mathcal{B}$ -eigenvalue of  $\mathcal{A}$ . For all  $\delta > 0$ , we have the following properties:*

- (i) *The relaxation (3.10) is infeasible for some  $N$  if and only if  $\lambda_k - \delta < \lambda_{min}$ .*
- (ii) *If  $\chi_k = \lambda_k$  and  $\lambda_{k+1}$  exists, then  $\lambda_{k+1} < \lambda_k - \delta$ , i.e., (3.8) holds.*
- (iii) *If  $\chi_k = \lambda_k$  and (3.10) is infeasible for some  $N$ , then  $\lambda_k = \lambda_{min}$  and  $\lambda_{k+1}$  does not exist.*

*Proof.* (i) This can be implied by Theorem 4.3 (i) of [26].

(ii) Clearly,  $\chi_k$  is the smallest  $\mathcal{B}$ -eigenvalue greater than or equal to  $\lambda_k - \delta$ . If  $\lambda_{k+1}$  exists and  $\chi_k = \lambda_k$ , we must have  $\lambda_{k+1} < \lambda_k - \delta$ .

(iii) From (i), we know  $\lambda_k - \delta < \lambda_{min}$ . If otherwise  $\lambda_{min} < \lambda_k$ , then  $\lambda_{k+1}$  exists and  $\lambda_{k+1} < \lambda_k - \delta$  by (ii). This results in the contradiction  $\lambda_{k+1} < \lambda_{min}$ . So,  $\lambda_{min} = \lambda_k$ .  $\square$

The problem (3.12) is also a polynomial optimization problem. Similar semidefinite relaxations like (3.10)-(3.11) can be constructed to solve it. The hierarchy of such relaxations can also be shown to have finite convergence (cf. [26]), by similar arguments. Thus, the optimal value  $\chi_k$  of (3.12) can be computed by solving its semidefinite relaxations. For  $\delta > 0$  sufficiently small, we must have  $\chi_k = \lambda_k$ , no matter if  $\lambda_{k+1}$  exists or not. This is because  $\chi_k$  is the smallest  $\mathcal{B}$ -eigenvalue greater than or equal to  $\lambda_k - \delta$ .

The existence of  $\lambda_{k+1}$  and the relation (3.8) can be checked as follows. First, we choose a small value (say, 0.05) for  $\delta$ , and then solve (3.12). If  $\chi_k < \lambda_k$ , we decrease the value  $\delta$  as  $\delta := \delta/5$  and solve (3.12) again. Repeat this process until we get  $\chi_k = \lambda_k$ . (This process must stop when  $\delta > 0$  is sufficiently small.) After  $\chi_k = \lambda_k$  is reached, there are only two possibilities: 1) If  $\lambda_{k+1}$  does not exist, then  $\lambda_k = \lambda_{min}$ . By Proposition 3.5(i), the relaxation (3.10) must be infeasible for some  $N$ . This then confirms the nonexistence of  $\lambda_{k+1}$ , by Proposition 3.5(iii). 2) If  $\lambda_{k+1}$  exists, then  $\lambda_{k+1} < \lambda_k - \delta$ , by Proposition 3.5(ii). So, (3.8) is satisfied. Then, by Theorem 3.3(i), we have  $\rho_N^{(k+1)} = \lambda_{k+1}$  for  $N$  sufficiently large. In summary, if  $\lambda_{k+1}$  does not exist, we can get a certificate for that; if it exists, we can get  $\lambda_{k+1}$  by solving the relaxation (3.10).

We would like to point out that some variations of eigenvalue problems can also be solved by using similar semidefinite relaxations. The largest real eigenvalue in an interval  $[a, b]$  is the optimal value of the problem

$$(3.13) \quad \begin{cases} \max & f(x) \\ \text{s.t.} & h_r(x) = 0 \ (r = 1, \dots, 2n-2), \\ & a \leq f(x) \leq b. \end{cases}$$

If, in advance, we know there exists an eigenvector  $u$  for  $\lambda_{k+1}$  lying in some region, say, defined by some polynomial inequalities  $p_1(x) \geq 0, \dots, p_s(x) \geq 0$ , then we can

get such  $u$  by solving the optimization problem

$$(3.14) \quad \begin{cases} \max & f(x) \\ \text{s.t.} & h_r(x) = 0 \ (r = 1, \dots, 2n - 2), \\ & f(x) \leq \lambda_k - \delta, \\ & p_1(x) \geq 0, \dots, p_s(x) \geq 0. \end{cases}$$

Similar semidefinite relaxations like (3.10)-(3.11) can be constructed to solve such polynomial optimization problems, and we can get the desired eigenpairs.

**3.3. Getting all real eigenpairs.** We can compute all real  $\mathcal{B}$ -eigenvalues sequentially as follows. First, we compute the largest one  $\lambda_1$  by solving the hierarchy of semidefinite relaxations (3.5)-(3.6). As shown in Theorem 3.1, this hierarchy converges in finitely many steps. After getting  $\lambda_1$ , we solve the hierarchy of (3.10)-(3.11) for  $k = 1$ . If  $\chi_1 = \lambda_1$  and (3.10) is infeasible for some  $N$ , then  $\lambda_1$  is the smallest eigenvalue. If  $\chi_1 = \lambda_1$  and (3.10) is feasible for all  $N$ , then  $\rho_N^{(2)} = \lambda_2$  for  $N$  big enough. Repeating this procedure, we can get  $\lambda_3, \lambda_4, \dots$  if they exist, or we get the smallest eigenvalue and stop.

As above, we get the following algorithm.

**Algorithm 3.6.** *Computing all real  $\mathcal{B}$ -eigenpairs of a symmetric tensor  $\mathcal{A}$ .*

**Step 0:** Choose a small positive value  $\delta_0$  (e.g., 0.05). Let  $k = 1$ .

**Step 1:** Solve the hierarchy of (3.5) and get the largest eigenvalue  $\lambda_1$ .

**Step 2:** Let  $\delta = \delta_0$  and solve the optimal value  $\chi_k$  of (3.12). If  $\chi_k = \lambda_k$ , then go to Step 3; If  $\chi_k < \lambda_k$ , let  $\delta := \min(\delta/5, \lambda_k - \chi_k)$ , and compute  $\chi_k$ . Repeat this process until  $\chi_k = \lambda_k$ .

**Step 3:** Solve the hierarchy of (3.10). If (3.10) is infeasible for some order  $N$ , then  $\lambda_k$  is the smallest eigenvalue and stop. Otherwise, we can get the next largest eigenvalue  $\lambda_{k+1}$ .

**Step 4:** Let  $k := k + 1$  and go to Step 2.

In Step 2, if  $\chi_k < \lambda_k$ , we should expect  $\delta < \lambda_k - \chi_k$ . This is why we update  $\delta$  as the minimum of  $\delta/5$  and  $\lambda_k - \chi_k$ .

#### 4. NUMERICAL EXPERIMENTS

In this section, we report numerical experiments for showing how to compute all real eigenvalues. The computation is implemented in a Thinkpad W520 Laptop, with an Intel® dual core CPU at 2.20GHz  $\times$  2 and 8GB of RAM, in a Windows 7 operating system. We use the software Matlab 2013a and GloptiPoly 3 [11] to solve the semidefinite relaxations for polynomial optimization problems. In the display of numerical results, we only show four decimal digits.

By the definition of  $\mathcal{B}$ -eigenvalues as in (1.1),  $(\lambda, u)$  is an eigenpair if and only if  $((-1)^{m-m'}\lambda, -u)$  is an eigenpair. For H-eigenvalues ( $m = m'$ ), the H-eigenvectors always appear in  $\pm$  pairs; so we only list H-eigenvectors  $u$  satisfying  $\Sigma_i u_i \geq 0$ . For Z-eigenvalues ( $m' = 2$ ), when  $m$  is even, the Z-eigenvectors appear in  $\pm$  pairs, and we only list those  $u$  satisfying  $\Sigma_i u_i \geq 0$ ; when  $m$  is odd,  $(\lambda, u)$  is a Z-eigenpair if and only if  $(-\lambda, -u)$  is a Z-eigenpair, and they appear in  $\pm$  pairs.

If the rank condition (3.7) is satisfied, then we can get the  $\mathcal{B}$ -eigenvalue  $\lambda_k$  and  $\ell := \text{rank } M_t(y^*)$   $\mathcal{B}$ -eigenvectors associated to  $\lambda_k$ . When primal-dual interior point methods are applied to solve the semidefinite relaxations and (3.7) holds, generally all  $\mathcal{B}$ -eigenvectors associated to  $\lambda_k$  can be obtained. We refer to Remarks 3.2 and

3.4. In our numerical experiments, the SDP solver `SeDuMi` [35] is called by the software `GloptiPoly 3`. The solver `SeDuMi` is based on primal-dual interior point methods. So, when the rank condition (3.7) is satisfied, we typically get all  $\mathcal{B}$ -eigenvectors of  $\lambda_k$ . In such cases, the real geometric multiplicities of computed eigenvalues are also known. In the display of our numerical results, we use the notation  $\lambda^{(\ell)}$  to mean that  $\ell$  distinct  $\mathcal{B}$ -eigenvectors (modulo scaling) are found for the eigenvalue  $\lambda$ .

When  $\lambda_k$  has infinitely many  $\mathcal{B}$ -eigenvectors on  $\mathcal{B}x^{m'} = 1$ , the rank condition (3.7) is typically not satisfied. To the best of the authors' knowledge, for such cases, it is a theoretically open question to check the convergence of (3.5)-(3.6) and (3.10)-(3.11), although they are proved to have finite convergence in Theorems 3.1 and 3.3. However, in practice, this issue can be fixed heuristically as follows. The sequence  $\{\rho_N^{(k)}\}$  always has finite convergence to  $\lambda_k$ . After an approximate convergence of  $\rho_N^{(k)}$  is observed, we can use such  $\rho_N^{(k)}$  as an approximation of  $\lambda_k$ . Let  $\epsilon > 0$  be small such that  $\lambda_k$  is a unique  $\mathcal{B}$ -eigenvalue of  $\mathcal{A}$  in the interval  $[\lambda_k - \epsilon, \lambda_k + \epsilon]$ . Choose a generic vector  $c \in \mathbb{R}^n$  and then solve the problem

$$(4.1) \quad \begin{cases} \min & c^T x \\ \text{s.t.} & h_r(x) = 0 \ (r = 1, \dots, 2n-2), \\ & \lambda_k - \epsilon \leq \mathcal{A}x^m \leq \lambda_k + \epsilon. \end{cases}$$

When  $c$  is generic, (4.1) has a unique minimizer, which is a  $\mathcal{B}$ -eigenvector associated to  $\lambda_k$ . We can construct semidefinite relaxations, like (3.10)-(3.11), for solving (4.1). A  $\mathcal{B}$ -eigenvector can be found by solving the semidefinite relaxations (cf. [25, §3]). In practice, a generic  $c$  can be chosen as a random vector in  $\mathbb{R}^n$ . In Matlab, we can set  $\mathbf{c} = \mathbf{randn}(n, 1)$ . A small enough  $\epsilon$  can be chosen as follows. We first assign a small value to  $\epsilon$ , say, 0.05. After solving (4.1), we are done if a  $\mathcal{B}$ -eigenvector associated to  $\lambda_k$  is found; otherwise, update  $\epsilon := \epsilon/5$  and solve (4.1) again. Repeat this process, until a  $\mathcal{B}$ -eigenvector  $u$  associated to  $\lambda_k$  is found. Once  $u$  is obtained, we check the equation  $\mathcal{A}u^{m-1} = \lambda_k \mathcal{B}u^{m'-1}$ . If it is satisfied, then  $(\lambda_k, u)$  is confirmed to be an eigenpair. In our examples, we use the superscript  $(\star)$  to mean that an eigenvector is computed by solving (4.1).

**Example 4.1.** ([28]) Consider the tensor  $\mathcal{A} \in \mathbf{S}^4(\mathbb{R}^3)$  such that

$$\mathcal{A}x^4 = x_1^4 + 2x_2^4 + 3x_3^4.$$

It is a diagonal tensor (i.e., its entries  $\mathcal{A}_{i_1 i_2 i_3}$  are all zeros except for  $i_1 = i_2 = i_3$ ). Its  $Z$ -eigenvalues were computed by  $Qi$  [28, Proposition 9]. For this tensor, the optimization problem (3.4) is

$$\begin{aligned} \max & \quad x_1^4 + 2x_2^4 + 3x_3^4 \\ \text{s.t.} & \quad 2x_1x_2^3 - x_2x_1^3 = 0, \quad 3x_1x_3^3 - x_3x_1^3 = 0, \\ & \quad 3x_2x_3^3 - 2x_3x_2^3 = 0, \quad x_1^2 + x_2^2 + x_3^2 = 1. \end{aligned}$$

Using Algorithm 3.6, we get all the real  $Z$ -eigenvalues and  $Z$ -eigenvectors, which are shown in Table 4.1. The computation takes about 9 seconds.

**Example 4.2.** For the diagonal tensor  $\mathcal{D} \in \mathbf{S}^5(\mathbb{R}^4)$  such that  $\mathcal{D}x^5 = x_1^5 + 2x_2^5 - 3x_3^5 - 4x_4^5$ , its orthogonal transformations have the same  $Z$ -eigenvalues as  $\mathcal{D}$  (cf.  $Qi$  [28, Theorem 7]). Consider  $\mathcal{A} \in \mathbf{S}^5(\mathbb{R}^4)$  such that  $\mathcal{A}x^5 = \mathcal{D}(Px)^5$ , where

$$P = (I - 2w_1w_1^T)(I - 2w_2w_2^T)(I - 2w_3w_3^T)$$

TABLE 4.1. Z-eigenpairs of the tensor in Example 4.1

$k$	1	2	3	4	5	6	7
$\lambda_k$	3.0000	2.0000	1.2000 <sup>(2)</sup>	1.0000	0.7500 <sup>(2)</sup>	0.6667 <sup>(2)</sup>	0.5455 <sup>(4)</sup>
$u_k$	0.0000	0.0000	0.0000	1.0000	0.8660	0.8165	$\pm 0.7386$
	0.0000	1.0000	0.7746	0.0000	0.0000	$\pm 0.5773$	$\pm 0.5222$
	1.0000	0.0000	$\pm 0.6324$	0.0000	$\pm 0.5000$	0.0001	0.4264

and  $w_1, w_2, w_3$  are randomly generated unit vectors. The order  $m = 5$  is odd, so the Z-eigenvalues of  $\mathcal{A}$  appear in  $\pm$  pairs. Using Algorithm 3.6, we get all 30 real Z-eigenvalues. It takes about 400 seconds. The nonnegative Z-eigenvalues are

$$4.0000, \quad 3.0000, \quad 2.0000, \quad 1.2163, \quad 1.0000, \quad 0.9611, \quad 0.8543, \quad 0.6057, \\ 0.5550, \quad 0.5402, \quad 0.4805, \quad 0.3887, \quad 0.3466, \quad 0.3261, \quad 0.2518.$$

For cleanness, the Z-eigenvectors are not shown.

**Example 4.3.** ([28, Example 3]) Consider the tensor  $\mathcal{A} \in \mathbb{S}^4(\mathbb{R}^3)$  such that

$$\mathcal{A}x^4 = 2x_1^4 + 3x_2^4 + 5x_3^4 + 4ax_1^2x_2x_3,$$

where  $a$  is a parameter. The polynomial optimization problem (3.4) is

$$\begin{aligned} \max \quad & 2x_1^4 + 3x_2^4 + 5x_3^4 + 4ax_1^2x_2x_3 \\ \text{s.t.} \quad & x_1^{p-1}(3x_2^3 + ax_1^2x_3) - x_2^{p-1}(2x_1^3 + 2ax_1x_2x_3) = 0, \\ & x_1^{p-1}(5x_3^3 + ax_1^2x_2) - x_3^{p-1}(2x_1^3 + 2ax_1x_2x_3) = 0, \\ & x_2^{p-1}(5x_3^3 + ax_1^2x_2) - x_3^{p-1}(3x_2^3 + ax_1^2x_3) = 0, \\ & x_1^p + x_2^p + x_3^p = 1, \end{aligned}$$

where  $p = 2$  for Z-eigenvalues and  $p = 4$  for H-eigenvalues. Using Algorithm 3.6, we get all the real Z and H eigenvalues, which are shown in Table 4.2. For each value of  $a$ , it takes a couple of seconds (from 5 to 20). For cleanness, the eigenvectors

TABLE 4.2. Z-eigenvalues and H-eigenvalues of the tensor in Example 4.3

	Z-eigenvalues						
$a = 0$	5.0000	3.0000	2.0000	1.8750 <sup>(2)</sup>	1.4286 <sup>(2)</sup>	1.2000 <sup>(2)</sup>	0.9679 <sup>(4)</sup>
$a = 0.25$	5.0000	3.0000	2.0000	1.8750 <sup>(2)</sup>	1.4412 <sup>(2)</sup>	1.2150 <sup>(2)</sup>	1.0881 <sup>(2)</sup> 0.8646 <sup>(2)</sup>
$a = 0.5$	5.0000	3.0000	2.0000	1.8750 <sup>(2)</sup>	1.4783 <sup>(2)</sup>	1.2593 <sup>(2)</sup>	1.2069 <sup>(2)</sup> 0.7243 <sup>(2)</sup>
$a = 1$	5.0000	3.0000	2.0000	1.8750 <sup>(2)</sup>	1.6133 <sup>(2)</sup>	0.4787 <sup>(2)</sup>	
$a = 3$	5.0000	3.0000	2.2147 <sup>(2)</sup>	2.0000	1.8750 <sup>(2)</sup>	-0.5126 <sup>(2)</sup>	
	H-eigenvalues						
$a = 0$	5.0000	3.0000	2.0000				
$a = 0.25$	5.0009 <sup>(2)</sup>	5.0000	3.0000	2.0000	1.9310 <sup>(2)</sup>		
$a = 0.5$	5.0137 <sup>(2)</sup>	5.0000	3.0000	2.0000	1.7517 <sup>(2)</sup>		
$a = 1$	5.1812 <sup>(2)</sup>	5.0000	3.0000	2.0000	1.2269 <sup>(2)</sup>		
$a = 3$	7.4505 <sup>(2)</sup>	5.0000	3.0000	2.0000	-1.3952 <sup>(2)</sup>		

are not shown.

**Example 4.4.** ([28, Example 4]) Let  $\mathcal{A} \in \mathbb{S}^4(\mathbb{R}^2)$  be the tensor such that

$$\mathcal{A}x^4 = 3x_1^4 + x_2^4 + 6ax_1^2x_2^2,$$

where  $a$  is a parameter. As shown in [28], this tensor always has two  $Z$ -eigenvalues  $\lambda = 3, \lambda = 1$ . When  $a < \frac{1}{3}$  or  $a > 1$ ,  $\mathcal{A}$  has another double  $Z$ -eigenvalue

$$\frac{3(9a^3 - 6a^2 - 3a + 2)}{2(3a - 2)^2}.$$

For some values of  $a$ , the  $Z$ -eigenvalues are shown in Table 4.3. For each case of  $a$ , the computation takes about 1 second.

TABLE 4.3.  $Z$ -Eigenvalues of the tensor in Example 4.4

	$\lambda_1$	$\lambda_2$	$\lambda_3$		$\lambda_1$	$\lambda_2$	$\lambda_3$
$a = -1$	3.0000	1.0000	$-0.6000^{(2)}$	$a = 0$	3.0000	1.0000	$0.7500^{(2)}$
$a = 0.25$	3.0000	1.0000	$0.9750^{(2)}$	$a = 0.5$	3.0000	1.0000	
$a = 2$	$4.1250^{(2)}$	3.0000	1.0000				

**Example 4.5.** ([15, Example 3.5], [27, Example 3.4]) Consider the tensor  $\mathcal{A} \in \mathbb{S}^4(\mathbb{R}^3)$  such that

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

Using Algorithm 3.6, we get all the real  $Z$ -eigenvalues and  $Z$ -eigenvectors. They are shown in Table 4.4. The computation takes about 9 seconds.

TABLE 4.4.  $Z$ -eigenpairs of the tensor in Example 4.5

$k$	1	2	3	4	5	6	7	8	9	10	11
$\lambda_k$	0.8893	0.8169	0.5105	0.3633	0.2682	0.2628	0.2433	0.1735	-0.0451	-0.5629	-1.0954
$u_k$	0.6672	0.2471	-0.7027	0.8412	-0.2635	0.4722	0.3598	-0.7780	0.5150	0.2676	0.6447
	0.7160	0.6099	0.4362	0.6616	-0.1318	0.4425	0.8870	0.9895	0.0947	-0.1088	0.3357
	0.9073	0.2531	0.7797	0.6135	0.1250	0.1762	-0.1796	0.9678	-0.5915	0.7467	0.3043

**Example 4.6.** ([31, Example 9.1]) Consider the tensor  $\mathcal{A} \in \mathbb{S}^3(\mathbb{R}^6)$  such that

$$\mathcal{A}x^3 = x_1^3 + \cdots + x_6^3 + 30x_1^2x_2 + \cdots + 30x_5^2x_6.$$

It is a cubic tensor of dimension six. Its  $Z$ -eigenvalues appear in  $\pm$  pairs. In total, there are 19 nonnegative  $Z$ -eigenvalues:

$$\begin{aligned} &16.2345 \quad 15.4552 \quad 15.4298 \quad 10.9710 \quad 8.7347 \quad 8.6596 \quad 8.5979 \quad 8.1888 \quad 7.2165 \quad 6.0000 \\ &5.5674 \quad 5.5668 \quad 5.5218 \quad 5.4817 \quad 5.1402 \quad 4.3358 \quad 4.2464 \quad 4.0225 \quad 3.9992. \end{aligned}$$

It takes about 10870 seconds to compute them. For cleanness, the  $Z$ -eigenvectors are not shown.

Characteristic tensors of hypergraphs have important applications, as shown in Li et al. [19]. The second largest  $Z$ -eigenvalue can be used to get a lower bound for the bipartition width. The following is such an example.

**Example 4.7.** ([19, Example 6.4]) Consider the tensor  $\mathcal{A} \in \mathbb{S}^4(\mathbb{R}^6)$  such that

$$\begin{aligned} -\mathcal{A}x^4 &= (x_1 - x_2)^4 + (x_1 - x_3)^4 + (x_1 - x_4)^4 + (x_1 - x_5)^4 + (x_1 - x_6)^4 \\ &+ (x_2 - x_3)^4 + (x_2 - x_4)^4 + (x_2 - x_5)^4 + (x_2 - x_6)^4 \\ &+ (x_3 - x_4)^4 + (x_3 - x_5)^4 + (x_3 - x_6)^4 \\ &+ (x_4 - x_5)^4 + (x_4 - x_6)^4 + (x_5 - x_6)^4. \end{aligned}$$

The polynomial  $\mathcal{A}x^4$  is symmetric in  $x$ . Every permutation of a  $Z$ -eigenvector is also a  $Z$ -eigenvector. So, we can add extra conditions  $x_1 \geq x_2 \geq \dots \geq x_6$  to (3.4) and (3.9), while not changing eigenvalues. Then we solve the corresponding semidefinite relaxations. The tensor  $\mathcal{A}$  has five real  $Z$ -eigenvalues, which are respectively

$$\lambda_1 = 0.0000, \lambda_2 = -4.0000, \lambda_3 = -4.5000, \lambda_4 = -6.0000, \lambda_5 = -7.2000.$$

The  $Z$ -eigenvectors, whose entries are ordered monotonically decreasing, are shown in Table 4.5. It takes about 280 seconds to get them. In the computation of  $\lambda_3$ , the

TABLE 4.5.  $Z$ -eigenpairs of the tensor in Example 4.7

$k$	$\lambda_k$	$u_k^T$
1	0.0000	(0.4082 0.4082 0.4082 0.4082 0.4082 0.4082 0.4082)
2	-4.0000 <sup>(20)</sup>	(0.4082 0.4082 0.4082 -0.4082 -0.4082 -0.4082)
3	-4.5000 <sup>(*)</sup>	(0.2887 0.2887 0.2887 0.2887 -0.5774 -0.5774)
4	-6.0000 <sup>(15)</sup>	(0.7071 0.0000 0.0000 0.0000 0.0000 -0.7071)
5	-7.2000 <sup>(6)</sup>	(0.1826 0.1826 0.1826 0.1826 0.1826 -0.9129)

rank condition (3.7) is not satisfied. We get one of its  $Z$ -eigenvectors by solving (4.1).

**Example 4.8.** ([37, Example 2]) Consider the tensor  $\mathcal{A} \in \mathbb{S}^4(\mathbb{R}^5)$  such that

$$\mathcal{A}x^4 = (x_1 + x_2 + x_3 + x_4)^4 + (x_2 + x_3 + x_4 + x_5)^4.$$

Using Algorithm 3.6, we get all the three real  $Z$ -eigenvalues of this tensor, which are respectively

$$\lambda_1 = 24.5000, \quad \lambda_2 = 0.5000, \quad \lambda_3 = 0.0000.$$

It takes about 320 seconds to get them. The  $Z$ -eigenvectors are shown in Table 4.6. There are infinitely many  $Z$ -eigenvectors for  $\lambda_3$ . In the computation of  $\lambda_3$ , the rank

TABLE 4.6.  $Z$ -eigenpairs of the tensor in Example 4.8

$k$	$\lambda_k$	$u_k^T$
1	24.5000	(0.2673 0.5345 0.5345 0.5345 0.2673)
2	0.5000	(0.7071 0.0000 0.0000 0.0000 -0.7071)
3	0.0000 <sup>(*)</sup>	(0.5253 0.3021 -0.4781 -0.3472 0.5318)

condition (3.7) is not satisfied. So, we solve (4.1) and get one of its  $Z$ -eigenvectors.

**Example 4.9.** ([2, Example 5.7]) Consider the cubic tensor  $\mathcal{A} \in \mathbb{S}^3(\mathbb{R}^3)$  such that

$$\mathcal{A}x^3 = 2x_1^3 + 3x_1x_2^2 + 3x_1x_3^2.$$

Using Algorithm 3.6 we get two real  $Z$ -eigenvalues, which are  $\lambda_1 = 2$  and  $\lambda_2 = -2$ . Their  $Z$ -eigenvectors are  $(1, 0, 0)$  and  $(-1, 0, 0)$  respectively. It takes about 1 second to compute them.

**Example 4.10.** ([2, Example 5.8]) Consider the tensor  $\mathcal{A} \in \mathbb{S}^6(\mathbb{R}^3)$  such that

$$\mathcal{A}x^6 = x_1^4x_2^2 + x_1^2x_2^4 + x_3^6 - 3x_1^2x_2^2x_3^2,$$

which is the Motzkin polynomial. Since  $\mathcal{A}x^6$  has only even powers in each of  $x_1, x_2, x_3$ , we can add the extra conditions  $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$  to (3.4) and

(3.9), while not changing eigenvalues. Then we solve the corresponding semidefinite relaxations. The tensor  $\mathcal{A}$  has three real  $H$ -eigenvalues. Using Algorithm 3.6, we get all of them, which are respectively

$$\lambda_1 = 1.0000, \lambda_2 = 0.0555, \lambda_3 = 0.0000.$$

The  $H$ -eigenvectors are shown in Table 4.7. It takes about 30 seconds.

TABLE 4.7.  $H$ -eigenpairs of the tensor in Example 4.10

$k$	$\lambda_k$	$u_k^T$
1	1.0000 <sup>(3)</sup>	(0.0000 0.0000 1.0000)
		(0.8909 ±0.8909 0.0000)
2	0.0555 <sup>(8)</sup>	(0.4487 ±0.9823 ±0.6735)
		(0.9823 ±0.4487 ±0.6735)
3	0.0000 <sup>(6)</sup>	(1.0000 0.0000 0.0000)
		(0.0000 1.0000 0.0000)
		(0.8327 ±0.8327 ±0.8327)

**Example 4.11.** ([27, Example 3.5]) Consider the tensor  $\mathcal{A} \in \mathbf{S}^3(\mathbb{R}^n)$  such that

$$\mathcal{A}_{ijk} = \frac{(-1)^i}{i} + \frac{(-1)^j}{j} + \frac{(-1)^k}{k}.$$

For the case  $n = 5$ , we get all the real  $Z$ -eigenvalues, which are respectively

$$\lambda_1 = 9.9779, \lambda_2 = 4.2876, \lambda_3 = 0.0000, \lambda_4 = -4.2876, \lambda_5 = -9.9779.$$

The computation takes about 150 seconds. The  $Z$ -eigenvectors of  $\lambda_1, \lambda_2, \lambda_3$  are shown in Table 4.8. The  $Z$ -eigenvector of  $\lambda_4$  (resp.,  $\lambda_5$ ) is just the negative of that of  $\lambda_2$  (resp.,  $\lambda_1$ ). In the computation of  $\lambda_3$ , the rank condition (3.7) is not satisfied.

TABLE 4.8.  $Z$ -eigenpairs of the tensor in Example 4.11

$k$	$\lambda_k$	$u_k^T$
1	9.9779	(-0.7313 -0.1375 -0.4674 -0.2365 -0.4146)
2	4.2876	(-0.1859 0.7158 0.2149 0.5655 0.2950)
3	0.0000 <sup>(*)</sup>	( 0.5072 -0.0980 0.4280 -0.7344 -0.1028)

We get a  $Z$ -eigenvector for  $\lambda_3$  by solving (4.1).

**Example 4.12.** ([27]) Consider the tensor  $\mathcal{A} \in \mathbf{S}^4(\mathbb{R}^n)$  such that

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

For the case  $n = 5$ , we get all the real  $Z$ -eigenvalues which are respectively

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

The  $Z$ -eigenvectors are shown in Table 4.9. It takes about 370 seconds. In the computation of  $\lambda_3$ , the rank condition (3.7) is not satisfied. We get a  $Z$ -eigenvector for  $\lambda_3$  by solving (4.1).

TABLE 4.9. Z-eigenpairs of the tensor in Example 4.12

$k$	$\lambda_k$	$u_k^T$
1	7.2595	( 0.2686 0.6150 0.3959 -0.1872 -0.5982)
2	4.6408	(-0.5055 0.1228 0.6382 0.5669 -0.0256)
3	0.0000 <sup>(*)</sup>	( 0.5935 0.3675 -0.1224 0.5449 0.4341)
4	-3.9204	(-0.1785 0.4847 0.7023 0.2742 -0.4060)
5	-8.8463	(-0.5809 -0.3563 0.1959 0.5680 0.4179)

**Example 4.13.** Consider the tensor  $\mathcal{A} \in \mathbb{S}^4(\mathbb{R}^n)$  such that

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

For the case  $n = 5$ , we get all the real Z-eigenvalues which are respectively

$$\lambda_1 = 34.5304, \quad \lambda_2 = 0.0000, \quad \lambda_3 = -101.1994.$$

The Z-eigenvectors are displayed in Table 4.10. It takes about 170 seconds to compute them. In the computation of  $\lambda_2$ , the rank condition (3.7) is not satisfied. We

TABLE 4.10. Z-eigenpairs of the tensor in Example 4.13

$k$	$\lambda_k$	$u_k^T$
1	34.5304	( 0.6665 0.1089 0.4132 0.6070 -0.0692)
2	0.0000 <sup>(*)</sup>	(-0.7276 -0.1080 0.4238 0.5178 -0.1060)
3	-101.1994	( 0.2248 0.5541 0.3744 0.2600 0.6953)

get a Z-eigenvector for  $\lambda_2$  by solving (4.1).

**Example 4.14.** Consider the tensor  $\mathcal{A} \in \mathbb{S}^5(\mathbb{R}^n)$  such that

$$\mathcal{A}_{i_1 \dots i_5} = \ln(i_1) + \dots + \ln(i_5).$$

For the case  $n = 4$ , we get all the real Z-eigenvalues which are respectively

$$\lambda_1 = 132.3070, \quad \lambda_2 = 0.7074, \quad \lambda_3 = 0.0000, \quad \lambda_4 = -0.7074, \quad \lambda_5 = -132.3070.$$

The Z-eigenvectors of  $\lambda_1, \lambda_2, \lambda_3$  are shown in Table 4.11. The Z-eigenvector of  $\lambda_4$  (resp.,  $\lambda_5$ ) is just the negative of that of  $\lambda_2$  (resp.,  $\lambda_1$ ). It takes about 420 seconds to compute them. In the computation of  $\lambda_3$ , the rank condition (3.7) is not satisfied.

TABLE 4.11. Z-eigenpairs of the tensor in Example 4.14

$k$	$\lambda_k$	$u_k^T$
1	132.3070	( 0.4030 0.4844 0.5319 0.5657)
2	0.7074	(-0.9054 -0.3082 0.0411 0.2890)
3	0.0000 <sup>(*)</sup>	(-0.8543 0.2645 0.4168 0.1617)

We get a Z-eigenvector for  $\lambda_3$  by solving (4.1).

**Example 4.15. (random tensors)** An interesting question is to determine the number of real Z-eigenvalues for the symmetric tensors. Cartwright and Sturmfels [2, Theorem 5.5] showed that every symmetric tensor  $\mathcal{A}$  of order  $m$  and dimension  $n$  has at most

$$M(m, n) := \frac{(m-1)^n - 1}{m-2}$$

distinct complex  $Z$ -eigenvalues. In [2],  $(\lambda, u)$  and  $((-1)^m \lambda, -u)$  are considered to be the same eigenpair. To be consistent with [2], for odd ordered tensors, we here only count their nonnegative  $Z$ -eigenvalues. Furthermore, they also showed that when  $\mathcal{A}$  is generic,  $\mathcal{A}$  has exactly  $M(m, n)$  distinct complex  $Z$ -eigenvalues. Clearly,  $M(m, n)$  is an upper bound for the number of real  $Z$ -eigenvalues. But it might not be sharp for generic tensors. In this example, we explore possibilities of distributions of the numbers of real  $Z$ -eigenvalues. For each  $(m, n)$ , we generate 50 symmetric tensors randomly. Each symmetric tensor is generated as the symmetrization of a random nonsymmetric tensor  $\mathbf{randn}(\mathbf{i}_1, \dots, \mathbf{i}_m)$  in Matlab. The number of their real  $Z$ -eigenvalues are shown in Table 4.12. The notation  $k^{\{\mu\}}$  means that there are  $\mu$  instances for which the number of real  $Z$ -eigenvalues equals to  $k$ . The table

TABLE 4.12. Numbers of real  $Z$ -eigenvalues of random symmetric tensors

$(m, n)$	$M(m, n)$	Numbers of real $Z$ -eigenvalues
(3, 5)	31	$7, 9^{\{5\}}, 11^{\{7\}}, 13^{\{4\}}, 15^{\{6\}}, 17^{\{2\}}, 19^{\{17\}}, 21^{\{7\}}, 23$
(3, 4)	15	$3^{\{5\}}, 5^{\{8\}}, 7^{\{8\}}, 9^{\{11\}}, 11^{\{17\}}, 13$
(3, 3)	7	$1^{\{4\}}, 3^{\{17\}}, 5^{\{16\}}, 7^{\{13\}}$
(4, 4)	40	$8^{\{2\}}, 10^{\{3\}}, 12^{\{8\}}, 14^{\{6\}}, 16^{\{8\}}, 18^{\{12\}}, 20^{\{6\}}, 22^{\{3\}}, 24, 28$
(4, 3)	13	$3^{\{3\}}, 5^{\{5\}}, 7^{\{27\}}, 9^{\{8\}}, 11^{\{7\}}$
(5, 4)	85	$13, 15^{\{4\}}, 17^{\{8\}}, 19^{\{2\}}, 21^{\{8\}}, 23^{\{8\}}, 25^{\{7\}}, 27^{\{9\}}, 31^{\{2\}}, 33$
(5, 3)	21	$5^{\{2\}}, 7^{\{9\}}, 9^{\{13\}}, 11^{\{17\}}, 13^{\{7\}}, 15^{\{2\}}$

confirms that  $M(m, n)$  is an upper bound for the numbers of real  $Z$ -eigenvalues. Moreover, the numbers of real  $Z$ -eigenvalues are not evenly distributed. We do not know the reason for such distributions.

Theoretically, Algorithm 3.6 is able to compute all real eigenvalues for all symmetric tensors, provided that the computer has sufficient capacity. In practice, the sizes of symmetric tensors, for which the eigenvalues can be computed by Algorithm 3.6, depend on the computer memory and the relaxation order  $N$ . The length of the variable  $y$  in (3.5) and (3.10) is  $\binom{n+2N}{2N}$ . It grows fast in the order  $N$ . In our computational experiences, for general tensors, a small order  $N$  is often enough. This fact was observed for random tensors in Example 4.15. However, for some special tensors, a big order  $N$  might be required. For such cases, it is often very hard to compute all real eigenvalues.

A different approach for computing all real eigenvalues is based on solving the system (1.1) directly for its real solutions. This can be done by using the numerical solver `NSolve` provided by `Mathematica`. Generally, `NSolve` can solve relatively small problems. The following is such an example.

**Example 4.16.** Consider the symmetric tensor  $\mathcal{A} \in \mathbb{S}^4(\mathbb{R}^n)$  such that

$$\begin{aligned} \mathcal{A}x^4 = & (x_1 - x_2)^4 + \cdots + (x_1 - x_n)^4 + (x_2 - x_3)^4 + \cdots + (x_2 - x_n)^4 \\ & + \cdots + (x_{n-1} - x_n)^4. \end{aligned}$$

Like in Example 4.7, we can add extra conditions  $x_1 \geq x_2 \geq \cdots \geq x_n$  to (3.4) and (3.9), while not changing eigenvalues. We compute its real  $Z$ -eigenvalues. The computational results are shown in Table 4.13. For the case  $n = 4$ , Algorithm 3.6 takes about 3 seconds, while `NSolve` takes about 19 seconds. They both get all

TABLE 4.13. Z-eigenpairs of the tensor in Example 4.16

	<i>alg.</i>	<i>time(s)</i>	<i>Z-eigenvalues</i>				
$n = 4$	Alg. 3.6	3.6	5.3333	5.0000	4.0000	0.0000	
	NSolve	19.3	5.3333	5.0000	4.0000	0.0000	
$n = 5$	Alg. 3.6	274.5	6.2500	5.5000	4.2500	4.1667	0.0000
	NSolve	—	—	—	—	—	—
$n = 6$	Alg. 3.6	280.2	7.2000	6.0000	4.5000	4.0000	0.0000
	NSolve	—	—	—	—	—	—
$n = 7$	Alg. 3.6	9565.6	8.1667 <sup>(2)</sup>	6.5000	4.9000 <sup>(2)</sup>	4.8846 <sup>(2)</sup>	4.7500
	NSolve	—	4.1667	4.0883 <sup>(2)</sup>	0.0000	—	—
$n = 8$	Alg. 3.6	938.2	9.1429 <sup>(2)</sup>	7.0000	5.3333 <sup>(2)</sup>	—	—
	NSolve	—	—	—	—	—	—
$n = 9$	Alg. 3.6	4173.8	10.1250 <sup>(2)</sup>	7.5000	5.7857 <sup>(2)</sup>	—	—
	NSolve	—	—	—	—	—	—
$n = 10$	Alg. 3.6	15310.5	11.1111 <sup>(2)</sup>	8.0000	6.2500 <sup>(2)</sup>	—	—
	NSolve	—	—	—	—	—	—

the real Z-eigenvalues correctly. For the case  $n = 5$ , Algorithm 3.6 gets all the real Z-eigenvalues in about 274 seconds, while NSolve can't get answers in 5 hours (we terminated the computation after 5 hours). In Table 4.13, “—” means that no computational results are returned. We can also get all real eigenvalues for  $n = 6, 7$ . For the bigger  $n = 8, 9, 10$ , we can get the first three largest Z-eigenvalues, but the other smaller Z-eigenvalues cannot be obtained. This is because, for such cases, we need to use the relaxation order  $N = 4$ , which causes the computer to run out of memory. For  $n = 8, 9, 10$ , the reported time is only for the first three biggest Z-eigenvalues. For the values of  $n$  bigger than 10, the computer runs out of memory and we cannot get the eigenvalues.

In Algorithm 3.6, if the real eigenvalues are not separated well, then the positive number  $\delta > 0$  need to be chosen very small. We consider the following example, thanks to an anonymous referee.

**Example 4.17.** Consider the tensor  $\mathcal{A} \in \mathbb{S}^3(\mathbb{R}^2)$  such that

$$\mathcal{A}_{111} = 1, \quad \mathcal{A}_{222} = 1 + 10^{-6},$$

and all the other entries are zeros. In Algorithm 3.6, to get the real Z-eigenvalues correctly, the value of  $\delta$  decreased to be smaller than  $10^{-6}$  during the loop. The computed nonnegative real Z-eigenvalues are

$$\lambda_1 = 1.000001, \quad \lambda_2 = 1.000000, \quad \lambda_3 = 0.707107.$$

The whole computation takes about 2 seconds.

We conclude this section by exploring how Algorithm 3.6 scales in terms of sizes of tensors.

**Example 4.18.** We explore the sizes of symmetric tensors for which Algorithm 3.6 can get all their Z-eigenvalues. Randomly generated symmetric tensors are tested, in the same way as in Example 4.15. The dimensions and orders of random symmetric tensors, whose real Z-eigenvalues can be found by Algorithm 3.6, are shown in Table 4.14. In the left half of Table 4.14, we choose values  $n = 3, 4, 5, 6$ .

For each  $n$  of them, we list the values of  $m > 2$  such that we can find all real  $Z$ -eigenvalues by Algorithm 3.6. Similarly, in the right half of Table 4.14, we choose values  $m = 3, 4, 5, 6$ . For each  $m$  of them, we list the values of  $n$  such that we can find all real  $Z$ -eigenvalues by Algorithm 3.6.

TABLE 4.14. Scaling of Algorithm 3.6 for computing all the  $Z$ -eigenvalues

$n$	$m$	$m$	$n$
3	3 4 5 6 7 8 9 10	3	2 3 4 5 6 7
4	3 4 5 6	4	2 3 4 5 6
5	3 4 5	5	2 3 4 5
6	3 4	6	2 3 4

**Acknowledgement** Chun-Feng Cui was partially supported by the Chinese NSF Grant (no. 11301016). Yu-Hong Dai was partially supported by the Chinese NSF grants (nos. 11331012 and 81173633) and the China National Funds for Distinguished Young Scientists (no. 11125107). Jiawang Nie was partially supported by the NSF grants DMS-0844775, DMS-1417985.

## REFERENCES

- [1] W. BRUNS AND U. VETTER. *Determinantal rings*, Lecture Notes in Mathematicas 1327, Springer-Verlag, 1988.
- [2] D. CARTWRIGHT AND B. STURMFELS. *The number of eigenvalues of a tensor*, Linear Algebra and its Applications, 438 (2013), pp. 942–952.
- [3] K. C. CHANG, K. PEARSON, AND T. ZHANG. *Perron-Frobenius theorem for nonnegative tensors*, Communications in Mathematical Sciences, 6 (2008), pp. 507–520.
- [4] Y. CHEN, Y.-H. DAI, D. HAN, AND W. SUN. *Positive semidefinite generalized diffusion tensor imaging via quadratic semidefinite programming*, SIAM Journal on Imaging Sciences, 6 (2013), pp. 1531–1552.
- [5] D. COX, J. LITTLE, AND D. O’SHEA. *Ideals, varieties, and algorithms: An Introduction to computational algebraic geometry and commutative algebra*, Springer, 2007.
- [6] R. CURTO AND L. FIALKOW. *Truncated  $K$ -moment problems in several variables*, Journal of Operator Theory, 54 (2005), pp. 189–226.
- [7] J. DEMMEL. *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, 1997.
- [8] L. HAN. *An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors*, Numerical Algebra, Control and Optimization, 3 (2013), pp. 583–599.
- [9] C. HAO, C. CUI, AND Y.-H. DAI. *A sequential subspace projection method for extreme  $Z$ -eigenvalues of supersymmetric tensors*, Numerical Linear Algebra with Applications, to appear, 2014.
- [10] D. HENRION AND J. B. LASSERRE. *Detecting global optimality and extracting solutions in GloptiPoly*, Positive polynomials in control, Lecture Notes in Control and Information Science, Springer, Berlin, 312 (2005), pp. 293–310.
- [11] D. HENRION, J. B. LASSERRE, AND J. LOEFBERG. *GloptiPoly 3: moments, optimization and semidefinite programming*, Optimization Methods and Software, Vol. 24, No. 4–5, pp. 761–779, 2009.
- [12] C. HILLAR AND L.-H. LIM. *Most tensor problems are NP-hard*, Journal of the ACM, 60 (2013), no. 6.
- [13] S. HU, Z. H. HUANG, AND L. QI. *Finding the extreme  $Z$ -eigenvalues of tensors via a sequential semidefinite programming method*, Numerical Linear Algebra with Applications, 20 (2013), pp. 972–984.
- [14] T. G. KOLDA AND B. W. BADER. *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.

- [15] T. G. KOLDA AND J. R. MAYO. *Shifted power method for computing tensor eigenpairs*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 1095–1124.
- [16] J. B. LASSERRE. *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization, 11 (2001), pp. 796–817.
- [17] J. B. LASSERRE. *Moments, positive polynomials and their applications*, Imperial College Press, 2009.
- [18] M. LAURENT. *Sums of squares, moment matrices and optimization over polynomials*, Emerging Applications of Algebraic Geometry, IMA Volumes in Mathematics and its Applications (Eds. M. Putinar and S. Sullivant), Springer, 149 (2009), pp. 157–270.
- [19] G. LI, L. QI, AND G. YU. *The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory*, Numerical Linear Algebra with Applications, 20 (2013), pp. 1001–1029.
- [20] L.-H. LIM. *Singular values and eigenvalues of tensors: a variational approach*, Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP '05), 1 (2005), pp. 129–132.
- [21] L.-H. LIM. *Tensors and hypermatrices*, L. Hogben (Ed.), Handbook of Linear Algebra, 2nd Ed., CRC Press, Boca Raton, FL, 2013.
- [22] Q. NI, L. QI, AND F. WANG. *An eigenvalue method for testing positive definiteness of a multivariate form*, Automatic Control, IEEE Transactions on, 53 (2008), pp. 1096–1107.
- [23] G. NI, L. QI, F. WANG, AND Y. WANG. *The degree of the E-characteristic polynomial of an even order tensor*, Journal of Mathematical Analysis and Applications, 329 (2007), pp. 1218–1229.
- [24] J. NIE. *An exact Jacobian SDP relaxation for polynomial optimization*, Mathematical Programming, 137 (2013), pp. 225–255.
- [25] J. NIE. *Certifying convergence of Lasserre’s hierarchy via flat truncation*, Mathematical Programming, Ser. A, 142 (2013), no. 1-2, pp. 485–510.
- [26] J. NIE. *The hierarchy of local minimums in polynomial optimization*, Mathematical Programming, to appear.
- [27] J. NIE AND L. WANG. *Semidefinite relaxations for best rank-1 tensor approximations*, SIAM Journal on Matrix Analysis and Applications, Vol. 35, No. 3, pp. 1155–1179, 2014.
- [28] L. QI. *Eigenvalues of a real supersymmetric tensor*, Journal of Symbolic Computation, 40 (2005), pp. 1302–1324.
- [29] L. QI, W. SUN, AND Y. WANG. *Numerical multilinear algebra and its applications*, Frontiers of Mathematics in China, 2 (2007), pp. 501–526.
- [30] L. QI AND K. L. TEO. *Multivariate polynomial minimization and its application in signal processing*, Journal of Global Optimization, 26 (2003), pp. 419–433.
- [31] L. QI, F. WANG, AND Y. WANG. *Z-eigenvalue methods for a global polynomial optimization problem*, Mathematical Programming, 118 (2009), pp. 301–316.
- [32] L. QI, Y. WANG, AND E. X. WU. *D-eigenvalues of diffusion kurtosis tensors*, Journal of Computational and Applied Mathematics, 221 (2008), pp. 150–157.
- [33] L. QI, G. YU, AND E. X. WU. *Higher order positive semidefinite diffusion tensor imaging*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 416–433.
- [34] B. REZNICK. *Some concrete aspects of Hilbert’s 17<sup>th</sup> problem*, Contemp. Math., American Mathematical Society, 253 (2000), pp. 251–272.
- [35] J. F. STURM. *SeDuMi 1.02: a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11&12 (1999), pp. 625–653. <http://sedumi.ie.lehigh.edu/>.
- [36] H. WOLKOWICZ, R. SAIGAL, AND L. VANDENBERGHE (EDS.). *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, International Series in Operations Research & Management Science, Springer, 2000.
- [37] J. XIE AND A. CHANG. *On the Z-eigenvalues of the signless Laplacian tensor for an even uniform hypergraph*, Numerical Linear Algebra with Applications, 20 (2013), pp. 1030–1045.
- [38] X. ZHANG, C. LING, AND L. QI. *The best rank-1 approximation of a symmetric tensor and related spherical optimization problems*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), no. 3, pp. 806–821.

STATE KEY LABORATORY OF SCIENTIFIC AND ENGINEERING COMPUTING, INSTITUTE OF COMPUTATIONAL MATHEMATICS AND SCIENTIFIC/ENGINEERING COMPUTING, ACADEMY OF MATHEMATICS AND SYSTEMS SCIENCE, CHINESE ACADEMY OF SCIENCES, P.O. Box 2719, BEIJING 100190, P.R. CHINA.

*E-mail address:* `cuichf@lsec.cc.ac.cn`

STATE KEY LABORATORY OF SCIENTIFIC AND ENGINEERING COMPUTING, INSTITUTE OF COMPUTATIONAL MATHEMATICS AND SCIENTIFIC/ENGINEERING COMPUTING, ACADEMY OF MATHEMATICS AND SYSTEMS SCIENCE, CHINESE ACADEMY OF SCIENCES, P.O. Box 2719, BEIJING 100190, P.R. CHINA.

*E-mail address:* `dyh@lsec.cc.ac.cn`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA SAN DIEGO, 9500 GILMAN DRIVE, LA JOLLA, CALIFORNIA 92093, USA.

*E-mail address:* `njw@math.ucsd.edu`