Electronic Transactions on Numerical Analysis.
Volume 62, pp. 72–94, 2024.

ISSN 1068–9613.
DOI: 10.1553/etna_vol62s72

ETNA
Kent State University and
Johann Radon Institute (RICAM)

CP DECOMPOSITION AND LOW-RANK APPROXIMATION OF ANTISYMMETRIC TENSORS*

ERNA BEGOVIĆ KOVAČ† AND LANA PERIŠA‡

Abstract. For antisymmetric tensors, the paper examines a low-rank approximation that is represented via only three vectors. We describe a suitable low-rank format and propose an alternating least-squares structure-preserving algorithm for finding such an approximation. Moreover, we show that this approximation problem is equivalent to the problem of finding the best multilinear low-rank antisymmetric approximation and, consequently, equivalent to the problem of finding the best unstructured rank-1 approximation. The case of partial antisymmetry is also discussed. The algorithms are implemented in the Julia programming language and their numerical performance is discussed.

Key words. CP decomposition, antisymmetric tensors, low-rank approximation, structure-preserving algorithm, Julia

AMS subject classifications. 15A69

1. Introduction.

Tensor decompositions have been extensively studied in recent years [2, 9, 10, 19, 22]. However, the research has mostly been focused on either unstructured or symmetric [7, 21] tensors. In this paper we explore antisymmetric tensors, their CP decomposition, and algorithms for the low-rank approximation.

The idea of the CP decomposition is to write a tensor as a sum of its rank-1 components. It was first introduced by Hitchcock [17, 18] in 1927, but it only became popular in the 1970s as CANDECOMP (canonical decomposition) [5] and PARAFAC (parallel factors) [16]. This decomposition is closely related to the tensor rank R, which is defined as the minimal number of rank-1 summands in the exact CP decomposition. Contrary to the matrix case, the rank of a tensor can exceed its dimension, and it can be different over $\mathbb{R}$ and over $\mathbb{C}$. It is known that the problem of finding the rank of a given tensor is NP-hard.

When computing the CP approximation, the main question is the choice of the number of rank-1 components. Given the antisymmetric structure of our tensors in question, we impose an additional constraint on the CP decomposition. This constraint assures that the resulting tensor is, indeed, antisymmetric, and it gives a bound on the minimal number of rank-1 components.

We focus on tensors of order 3. For a given antisymmetric tensor $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ our goal is to find its low-rank antisymmetric approximation that is represented via only three vectors. In particular, we are looking for the approximation $\widetilde{\mathcal{A}}$ of $\mathcal{A}$ such that $\operatorname{rank}(\widetilde{\mathcal{A}}) \leq 6$ for any n, and

$$\widetilde{\mathcal{A}} = \frac{1}{6}(x\circ y\circ z + y\circ z\circ x + z\circ x\circ y - x\circ z\circ y - y\circ x\circ z - z\circ y\circ x),$$

where $x, y, z \in \mathbb{R}^n$. We propose an alternating least-squares structure-preserving algorithm for solving this problem. The algorithm is based on solving a minimization problem in each tensor mode. We compare our algorithm with a “naive” idea which uses a posteriori antisymmetrization. Further on, we show that our approximation problem is equivalent to the problem of the best multilinear rank-3 structure-preserving antisymmetric tensor approximation from [3]

*Received December 6, 2023. Accepted July 28, 2024. Published online on September 6, 2024. Recommended by Daniel Kressner. Work supported by the Croatian Science Foundation under the project UIP-2019-04-5200.
†University of Zagreb, Faculty of Chemical Engineering and Technology, Marulićev trg 19, 10000 Zagreb, Croatia (ebegovic@fkit.hr).
‡Visage Technologies, Ivana Lučića 2a, 10000 Zagreb, Croatia (lana.perisa@visagetechnologies.com).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

and, consequently, equivalent to the problem of the best unstructured rank-1 approximation. This establishes the equivalence between our algorithm and the higher-order power method (HOPM). Therefore, the corresponding convergence result for the HOPM from [30] can be applied.

Additionally, we study tensors with partial antisymmetry, that is, antisymmetry in only two modes. Similarly to what we do for the tensors that are antisymmetric in all modes, we first determine a suitable format of the CP decomposition that is going to be simpler for the partial antisymmetry. Based on this format, for a given tensor $\mathcal{C} \in \mathbb{R}^{n\times n\times m}$ antisymmetric in two modes, we look for its approximation $\widetilde{\mathcal{C}}$ of the same structure such that $\widetilde{\mathcal{C}}$ is represented by three vectors and $\operatorname{rank}(\widetilde{\mathcal{C}}) = 2$.

In Section 2 we introduce the notation and preliminaries. Our problem of antisymmetric tensor approximation is described in Section 3. In Section 4 we describe the approach with a posteriori antisymmetrization, while in Section 5 we propose the algorithm for solving the minimization problem from Section 3. Section 6 deals with the case of partial antisymmetry. In Section 7 we discuss our numerical results obtained in the Julia programming language; finally, the conclusion is given in Section 8.

2. Notation and preliminaries. Throughout the paper we denote tensors by calligraphic letters, e.g., $\mathcal{A}$. We refer to the tensor dimension as its *order*. Then, for $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ we say that $\mathcal{A}$ is a tensor of order d. Tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ is *cubical* if $n_1 = n_2 = \cdots = n_d$. Vectors obtained from a tensor by fixing all indices except the mth one are called *mode-m fibers*. The fibers of an order-3 tensor are columns (mode-1 fibers), rows (mode-2 fibers), and tubes (mode-3 fibers). Matrices obtained from a tensor by fixing all indices except two are called *slices*. The matrix representation of a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ is called mode-m *matricization* and is denoted by $A_{(m)}$. It is obtained by arranging the mode-m fibers of $\mathcal{A}$ as columns of $A_{(m)}$.

The *mode-m product* of a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ with a matrix $M \in \mathbb{R}^{p\times n_m}$ is a tensor $\mathcal{B} \in \mathbb{R}^{n_1\times\cdots\times n_{m-1}\times p\times n_{m+1}\times\cdots\times n_d}$, i.e.,

$$\mathcal{B} = \mathcal{A} \times_m M, \quad \text{such that} \quad B_{(m)} = M A_{(m)}.$$

The tensor *norm* is a generalization of the Frobenius norm. For $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ we have

$$\|\mathcal{A}\| = \sqrt{\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\cdots\sum_{i_d=1}^{n_d} a_{i_1 i_2\ldots i_d}^2}.$$

The *inner product* of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ is given by

$$\langle \mathcal{A}, \mathcal{B}\rangle = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\cdots\sum_{i_d=1}^{n_d} a_{i_1 i_2\ldots i_d} b_{i_1 i_2\ldots i_d}.$$

The vector *outer product* is denoted by $\circ$. A tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ is a rank-1 tensor if it can be written as the outer product of d vectors,

$$\mathcal{A} = v^{(1)} \circ v^{(2)} \circ \cdots \circ v^{(d)}.$$

Then

$$a_{i_1 i_2\ldots i_d} = v_{i_1}^{(1)} v_{i_2}^{(2)} \cdots v_{i_d}^{(d)}, \quad 1 \le i_k \le n_k,\ 1 \le k \le d.$$

The *Khatri–Rao product* of two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$ is defined as

$$A \odot B = \begin{bmatrix} a_1 \otimes b_1 & a_2 \otimes b_2 & \cdots & a_n \otimes b_n \end{bmatrix} \in \mathbb{R}^{(mp)\times n},$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where a_k and b_k denote the kth columns of A and B, respectively. The *Hadamard (element-wise) product* of two matrices $A, B \in \mathbb{R}^{m\times n}$ is defined as

$$A * B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1n}b_{1n} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2n}b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & a_{m2}b_{m2} & \cdots & a_{mn}b_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

The *Moore–Penrose inverse* of A is denoted by A^+.

For a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, its *CP approximation* takes the form

$$\mathcal{A} \approx \sum_{i=1}^{r} (x_i \circ y_i \circ z_i), \tag{2.1}$$

where $x_i \in \mathbb{R}^{n_1}$, $y_i \in \mathbb{R}^{n_2}$, and $z_i \in \mathbb{R}^{n_3}$. If we arrange vectors x_i, y_i, z_i $(i = 1, \dots, r)$ into matrices

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_r \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_r \end{bmatrix}, \quad Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_r \end{bmatrix},$$

then relation (2.1) can be written as

$$\mathcal{A} \approx [[X, Y, Z]] = \sum_{i=1}^{r} (x_i \circ y_i \circ z_i). \tag{2.2}$$

The smallest number r in the exact CP decomposition (2.2) is called the *tensor rank*. We write $\operatorname{rank}(\mathcal{A}) = r$.

The most commonly used algorithm for computing the CP approximation is the alternating least-squares (ALS) algorithm (see, e.g., [22]). In Algorithm 1 we give the CP-ALS algorithm for order-3 tensors.

Algorithm 1 CP-ALS.

Input: $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$, $r \in \mathbb{N}$
Output: $X, Y, Z \in \mathbb{R}^{n\times r}$
Initialize X, Y, Z as leading r left singular vectors of $A_{(i)}$, $i = 1, 2, 3$, respectively.
repeat
  $X = A_{(1)}(Z \odot Y)(Y^TY * Z^TZ)^+$
  $Y = A_{(2)}(Z \odot X)(X^TX * Z^TZ)^+$
  $Z = A_{(3)}(Y \odot X)(X^TX * Y^TY)^+$
until convergence or maximum number of iterations

3. Problem description. A cubical tensor is *symmetric* (sometimes also called super-symmetric) if its elements are invariant to any permutation of indices. On the contrary, a cubical tensor is *antisymmetric* if its elements change sign when permuting pairs of indices. In particular, an order-3 tensor $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ is antisymmetric if

$$a_{ijk} = a_{jki} = a_{kij} = -a_{ikj} = -a_{jik} = -a_{kji}, \quad 1 \le i, j, k \le n. \tag{3.1}$$

Such tensors are also called alternating 3-tensors $\Lambda^3(\mathbb{R}_n)$ [23], or 3-vectors [14]. The antisymmetric tensors appear in applications such as quantum chemistry [27] and electromagnetism [26]. Besides, they are interesting from the mathematical point of view [3, 15]. From the definition of the antisymmetric tensor $\mathcal{A}$, it obviously follows that:

(i) In all modes, all slices of $\mathcal{A}$ are antisymmetric matrices.
(ii) In all modes, all slices have one null column and one null row.
(iii) An antisymmetric tensor is data-sparse in the sense that many of its non-zero elements are the same, up to the sign.

These facts are useful when it comes to the implementation of specific algorithms.

We can define the *antisymmetrizer* “anti” as the orthogonal projection of a general cubical order-d tensor $\mathcal{B}$ to the subspace of antisymmetric tensors. Then, $\mathcal{A} = \text{anti}(\mathcal{B})$ is an order-d tensor given by

$$\mathcal{A}(i_1, i_2, \dots, i_d) \coloneqq \frac{1}{d!} \sum_{p \in \pi(d)} \text{sign}(p) \mathcal{B}(p(i_1), p(i_2), \dots, p(i_d)),$$

where $\pi(d)$ denotes the set of all permutations of length d. Hence, for $d = 3$, $\mathcal{B} \in \mathbb{R}^{n\times n\times n}$, and $\mathcal{A} = \text{anti}(\mathcal{B})$ we have

$$a_{ijk} = \frac{1}{6}(b_{ijk} + b_{jki} + b_{kij} - b_{ikj} - b_{jik} - b_{kji}). \tag{3.2}$$

Let $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ be an antisymmetric tensor of order 3. Take a triplet of indices (i, j, k), $1 \le i < j < k \le n$. It follows from (3.1) that a subtensor $\widehat{\mathcal{A}}$ of $\mathcal{A}$ obtained at the intersection of the ith, jth, and kth column, row, and tube is of the form

$$\widehat{\mathcal{A}} = \alpha \mathcal{E},$$

where $\alpha \in \mathbb{R}$ and $\mathcal{E}$ is a $3 \times 3 \times 3$ tensor such that

$$\mathcal{E}(i_1, i_2, i_3) = \begin{cases} 1, & \text{if the indices make an even permutation of } (1, 2, 3), \\ -1, & \text{if the indices make an odd permutation of } (1, 2, 3), \\ 0, & \text{if two or more indices are equal.} \end{cases}$$

The tensor $\mathcal{E}$ is called the *Levi-Civita tensor* [12]. We can also write $\mathcal{E}$ using its matricization

$$E_{(1)} = \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{3.3}$$

Obviously, $\mathcal{E}$ is the simplest possible antisymmetric non-zero order-3 tensor.

For three given vectors $x, y, z \in \mathbb{R}^n$ we define an $n \times n \times n$ antisymmetric tensor associated to these vectors as

$$\mathcal{A}_6(x, y, z) \coloneqq \frac{1}{6}(x \circ y \circ z + y \circ z \circ x + z \circ x \circ y - x \circ z \circ y - y \circ x \circ z - z \circ y \circ x). \tag{3.4}$$

Note that tensor $\mathcal{E}$ is a special case of the antisymmetric tensor $\mathcal{A}_6(x, y, z)$. For $x = [6, 0, 0]^T$, $y = [0, 1, 0]^T$, and $z = [0, 0, 1]^T$, we get $\mathcal{A}_6(x, y, z) = \mathcal{E}$. Moreover, for a rank-1 tensor $\mathcal{T} = [[x, y, z]]$, we have $\mathcal{A}_6(x, y, z) = \text{anti}(\mathcal{T})$. The tensor format (3.4) can be favorable because it represents an antisymmetric tensor via only three vectors, that is, $3n$ entries. On the other hand, the standard form of an $n \times n \times n$ antisymmetric tensor contains $\binom{n}{3}$ different entries. Besides, tensor $\mathcal{A}_6(x, y, z)$ is a low-rank tensor. For any size n, we have $\text{rank}(\mathcal{A}_6(x, y, z)) \le 6$.

Our goal is to approximate a given antisymmetric tensor $\mathcal{A}$ with a low-rank antisymmetric tensor of the form (3.4). We demonstrate two approaches. The “naive” one is given in Section 4. Then, in Section 5 we formulate this problem as a minimization problem. For a given non-zero antisymmetric tensor $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$, we are looking for a tensor $\widetilde{\mathcal{A}} = \mathcal{A}_6(x, y, z)$, i.e., vectors $x, y, z \in \mathbb{R}^n$, such that

$$\|\mathcal{A} - \widetilde{\mathcal{A}}\|^2 \to \min.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

4. CP-ALS with a posteriori antisymmetrization. First we describe the naive approach. The process is made up of two steps.

Step 1: Using the CP-ALS algorithm (Algorithm 1), which ignores the tensor structure, find a rank-1 approximation $\bar{\mathcal{A}}$ of $\mathcal{A}$,

$$\bar{\mathcal{A}} = [[x, y, z]], \quad \operatorname{rank}(\bar{\mathcal{A}}) = 1.$$

Step 2: Apply the antisymmetrizer (3.2) on $\bar{\mathcal{A}}$ to obtain $\widetilde{\mathcal{A}}$ in the form (3.4),

$$\widetilde{\mathcal{A}} = \operatorname{anti}(\bar{\mathcal{A}}),$$

that is,

$$\widetilde{\mathcal{A}} = \mathcal{A}_6(x, y, z).$$

This procedure is given in Algorithm 2. We do not need to form the tensor $\bar{\mathcal{A}}$ explicitly.

Algorithm 2 CP with a posteriori antisymmetrization.

Input: $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ antisymmetric
Output: $\widetilde{\mathcal{A}} = \mathcal{A}_6(x, y, z)$
Apply Algorithm 1 on $\mathcal{A}$ with $r = 1$ to obtain $x, y, z \in \mathbb{R}^n$
$\widetilde{\mathcal{A}} = \mathcal{A}_6(x, y, z)$

Obviously, using a rank-1 intermediate tensor produces an unnecessarily large approximation error. However, it can be easily shown that, if the error of the rank-1 approximation is bounded by some $\epsilon > 0$, the resulting error will also be bounded by ϵ.

5. Antisymmetry-preserving CP algorithm. For a given antisymmetric tensor $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ we are looking for vectors $x, y, z \in \mathbb{R}^n$ such that

$$\|\mathcal{A} - \mathcal{A}_6(x, y, z)\|^2 \to \min. \tag{5.1}$$

Contrary to Algorithm 1, here we develop a new structure-preserving low-rank approximation algorithm. Our algorithm uses the ALS approach, that is, we are solving an optimization problem in each mode. It results in a tensor of the form (3.4) and there is no need to apply the antisymmetrizer. ALS algorithms are widely used to address different multilinear minimization problems [8, 29, 11, 13], including the ones regarding the CP approximation [1, 24, 20]. There is also a very recent extension to the antisymmetric case [28], but both the problem and the algorithm are different from ours.

Set

$$a = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^{3n}.$$

Then, similarly to what was done in [1], we define the objective function $f\colon \mathbb{R}^{3n} \to \mathbb{R}$ as

$$f(a) = 6\|\mathcal{A} - \mathcal{A}_6(x, y, z)\|^2. \tag{5.2}$$

We consider three partial minimization problems:

$$\min_x f(a), \quad \min_y f(a), \quad \min_z f(a). \tag{5.3}$$

Before we formulate the algorithm, we need to prove Theorem 5.1 below. It gives three reformulations of the objective function f that we are going to use in order to find the solutions of the problems (5.3).

Observe that, since $\mathcal{A}_6(x,y,z)$ is linear in x, y, and z, the objective function is quadratic in x, y, and z. The approximation problem becomes a quadratic optimization problem. Here we derive the quadratic forms explicitly. However, it is worth mentioning that the underlying linearity opens the possibilities of extension to more general settings.

In order to simplify the statement of the theorem, we define the following objects: the matrices $Q^{(1)} = Q^{(1)}(y,z)$, $Q^{(2)} = Q^{(2)}(x,z)$, $Q^{(3)} = Q^{(3)}(x,y) \in \mathbb{R}^{n\times n}$,

$$Q^{(1)} = 2\left((\|y\|_2^2\|z\|_2^2 - \langle y,z\rangle^2)I_n + (yz^T - zy^T)^2\right), \tag{5.4}$$

$$Q^{(2)} = 2\left((\|z\|_2^2\|x\|_2^2 - \langle z,x\rangle^2)I_n + (zx^T - xz^T)^2\right), \tag{5.5}$$

$$Q^{(3)} = 2\left((\|x\|_2^2\|y\|_2^2 - \langle x,y\rangle^2)I_n + (xy^T - yx^T)^2\right), \tag{5.6}$$

the vectors $c^{(1)} = c^{(1)}(y,z)$, $c^{(2)} = c^{(2)}(x,z)$, $c^{(3)} = c^{(3)}(x,y) \in \mathbb{R}^n$,

$$c^{(1)} = -12\mathcal{A} \times_2 y^T \times_3 z^T, \tag{5.7}$$

$$c_i^{(2)} = -12\mathcal{A} \times_2 z^T \times_3 x^T, \tag{5.8}$$

$$c_i^{(3)} = -12\mathcal{A} \times_2 x^T \times_3 y^T, \tag{5.9}$$

and the real number

$$d = 6\|\mathcal{A}\|^2. \tag{5.10}$$

THEOREM 5.1. *The function f defined by* (5.2) *can be written as*

$$f(a) = d + (c^{(1)})^T x + \frac{1}{2} x^T Q^{(1)} x, \tag{5.11}$$

$$= d + (c^{(2)})^T y + \frac{1}{2} y^T Q^{(2)} y, \tag{5.12}$$

$$= d + (c^{(3)})^T z + \frac{1}{2} z^T Q^{(3)} z, \tag{5.13}$$

for $Q^{(1)}, Q^{(2)}, Q^{(3)} \in \mathbb{R}^{n\times n}$, $c^{(1)}, c^{(2)}, c^{(3)} \in \mathbb{R}^n$, and $d \in \mathbb{R}$ defined by the relations (5.4)–(5.10).

Proof. First, we can write the function f from (5.2) as

$$\begin{aligned} f(a) &= 6\|\mathcal{A}\|^2 - 2\langle \mathcal{A}, 6\mathcal{A}_6(x,y,z)\rangle + \frac{1}{6}\|6\mathcal{A}_6(x,y,z)\|^2 \\ &= 6f_1(a) - 2f_2(a) + \frac{1}{6}f_3(a), \end{aligned} \tag{5.14}$$

where

$$f_1(a) = \|\mathcal{A}\|^2,$$

$$f_2(a) = \langle \mathcal{A}, x\circ y\circ z + y\circ z\circ x + z\circ x\circ y - x\circ z\circ y - y\circ x\circ z - z\circ y\circ x\rangle, \tag{5.15}$$

$$f_3(a) = \|x\circ y\circ z + y\circ z\circ x + z\circ x\circ y - x\circ z\circ y - y\circ x\circ z - z\circ y\circ x\|^2. \tag{5.16}$$

For the function f_2 we have

$$f_2(a) = \sum_{i,j,k=1}^{n} a_{ijk}(x_iy_jz_k + y_iz_jx_k + z_ix_jy_k - x_iz_jy_k - y_ix_jz_k - z_iy_jx_k)$$

$$
\begin{aligned}
&= \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} a_{ijk} y_j z_k + \sum_{k=1}^{n} x_k \sum_{i,j=1}^{n} a_{ijk} y_i z_j \\
&\quad + \sum_{j=1}^{n} x_j \sum_{i,k=1}^{n} a_{ijk} z_i y_k + \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} (-a_{ijk}) z_j y_k \\
&\quad + \sum_{k=1}^{n} x_k \sum_{i,j=1}^{n} (-a_{ijk}) z_i y_j + \sum_{j=1}^{n} x_j \sum_{i,k=1}^{n} (-a_{ijk}) y_i z_k.
\end{aligned}
$$

We rename the indices in the upper expression and use the fact that $\mathcal{A}$ is antisymmetric. We get

$$
\begin{aligned}
f_2(a) &= \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} a_{ijk} y_j z_k + \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} a_{jki} y_j z_k \\
&\quad + \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} a_{kij} y_j z_k + \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} (-a_{ikj}) y_j z_k \\
&\quad + \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} (-a_{kji}) y_j z_k + \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} (-a_{jik}) y_j z_k \\
&= 6 \sum_{i=1}^{n} x_i \sum_{j,k=1}^{n} a_{ijk} y_j z_k. \qquad (5.17)
\end{aligned}
$$

Next, we write the function f_3 as

$$
f_3(a) = \sum_{i,j,k=1}^{n} (x_i y_j z_k + y_i z_j x_k + z_i x_j y_k - x_i z_j y_k - y_i x_j z_k - z_i y_j x_k)^2. \qquad (5.18)
$$

After regrouping the summands and renaming the indices, as we did for f_2, it follows from (5.18) that

$$
\begin{aligned}
f_3(a) &= 6 \sum_{i=1}^{n} x_i^2 \left(\sum_{j,k=1}^{n} y_j^2 z_k^2 \right) - 6 \sum_{i=1}^{n} x_i^2 \left(\sum_{j,k=1}^{n} y_j y_k z_j z_k \right) \\
&\quad + 12 \sum_{i,j=1}^{n} x_i x_j \left(\sum_{k=1}^{n} y_i y_k z_j z_k \right) - 6 \sum_{i,j=1}^{n} x_i x_j \left(\sum_{k=1}^{n} y_i y_j z_k^2 \right) \\
&\quad - 6 \sum_{i,j=1}^{n} x_i x_j \left(\sum_{k=1}^{n} y_k^2 z_i z_j \right) \\
&= \sum_{i=1}^{n} x_i^2 \left(6 \sum_{j=1}^{n} y_j^2 \sum_{k=1}^{n} z_k^2 - 6 \left(\sum_{j=1}^{n} y_j z_j \right)^2 \right) \\
&\quad + \sum_{i,j=1}^{n} x_i x_j \left(12 \sum_{k=1}^{n} y_i y_k z_j z_k - 6 \sum_{k=1}^{n} y_i y_j z_k^2 - 6 \sum_{k=1}^{n} y_k^2 z_i z_j \right) \\
&= \sum_{i=1}^{n} x_i^2 \left(6 \sum_{j=1}^{n} y_j^2 \sum_{k=1}^{n} z_k^2 - 6 \left(\sum_{j=1}^{n} y_j z_j \right)^2 \right.
\end{aligned}
$$

$$
\begin{aligned}
&+ 12y_iz_i \sum_{k=1}^{n} y_kz_k - 6y_i^2 \sum_{k=1}^{n} z_k^2 - 6z_i^2 \sum_{k=1}^{n} y_k^2\Bigg) \\
&+ \sum_{\substack{i,j=1\\ i<j}}^{n} x_ix_j \Bigg(12y_iz_j \sum_{k=1}^{n} y_kz_k + 12z_iy_j \sum_{k=1}^{n} y_kz_k \\
&\quad - 12y_iy_j \sum_{k=1}^{n} z_k^2 - 12z_iz_j \sum_{k=1}^{n} y_k^2 \Bigg).
\end{aligned}
$$

That is,

$$
\begin{aligned}
f_3(a) &= \sum_{i=1}^{n} x_i^2 \left(6\|y\|_2^2\|z\|_2^2 - 6\langle y, z\rangle^2 + 12y_iz_i\langle y, z\rangle - 6y_i^2\|z\|_2^2 - 6z_i^2\|y\|_2^2\right) \\
&\quad + \sum_{\substack{i,j=1\\ i<j}}^{n} x_ix_j \left(12(y_iz_j + z_iy_j)\langle y, z\rangle - 12y_iy_j\|z\|_2^2 - 12z_iz_j\|y\|_2^2\right).
\end{aligned} \tag{5.19}
$$

Then, we can set

$$
\begin{aligned}
d &= 6f_1(a), \\
(c^{(1)})^T x &= -2f_2(a), \\
\frac{1}{2}x^TQ^{(1)}x &= \frac{1}{6}f_3(a).
\end{aligned}
$$

From the relations (5.14), (5.17), and (5.19) we get the assertion (5.11) where

$$
c_i^{(1)} = -12 \sum_{j,k=1}^{n} a_{ijk}y_jz_k, \quad 1 \le i \le n, \tag{5.20}
$$

$$
\begin{aligned}
q_{ii}^{(1)} &= 2\|y\|_2^2\|z\|_2^2 - 2\langle y, z\rangle^2 + 4y_iz_i\langle y, z\rangle - 2y_i^2\|z\|_2^2 - 2z_i^2\|y\|_2^2, \\
q_{ij}^{(1)} &= 2(y_iz_j + z_iy_j)\langle y, z\rangle - 2y_iy_j\|z\|_2^2 - 2z_iz_j\|y\|_2^2, \quad 1 \le i, j \le n,\ i \ne j,
\end{aligned} \tag{5.21}
$$

and d is as given in (5.10). It follows from the expressions in (5.21) that

$$
\begin{aligned}
Q^{(1)} &= 2\left(\|y\|_2^2\|z\|_2^2 - \langle y, z\rangle^2\right)I_n + 2\left((yz^T + zy^T)\langle y, z\rangle - yy^T\|z\|_2^2 - zz^T\|y\|_2^2\right) \\
&= 2\left((\|y\|_2^2\|z\|_2^2 - \langle y, z\rangle^2)I_n + (yz^T - zy^T)^2\right),
\end{aligned}
$$

while the vector given element-wise by (5.20) is equal to that from relation (5.7).

Similarly, using a different regrouping of the summands in equations (5.15) and (5.16), we obtain the assertions (5.12) and (5.13). We get $Q^{(2)}$ and $c^{(2)}$, as in the relations (5.5) and (5.8), respectively, as well as $Q^{(3)}$ and $c^{(3)}$, as in (5.6) and (5.9), respectively. □

The minimization problem of the form

$$
\min_v \left\{ d + c^Tv + \frac{1}{2}v^TQv \right\}
$$

is a problem of quadratic programming with no constraints. Its solution v is given by the linear system

$$
Qv = -c.
$$

Therefore, in order to find the solutions of the minimization problems

$$
\begin{cases} \min_x d + (c^{(1)})^T x + \dfrac{1}{2} x^T Q^{(1)} x, \\ \min_y d + (c^{(2)})^T y + \dfrac{1}{2} y^T Q^{(2)} y, \\ \min_z d + (c^{(3)})^T z + \dfrac{1}{2} z^T Q^{(3)} z, \end{cases} \tag{5.22}
$$

we need to solve the linear systems

$$
\begin{cases} Q^{(1)} x = -c^{(1)}, \\ Q^{(2)} y = -c^{(2)}, \\ Q^{(3)} z = -c^{(3)}, \end{cases}
$$

respectively.

Here we come to an obstacle because the matrices $Q^{(1)}$, $Q^{(2)}$, and $Q^{(3)}$ are singular. Take $Q^{(1)}$. From the relation (5.4) we see that $Q^{(1)}$ is defined by two vectors y and z and we have

$$
Q^{(1)} y = 0, \quad Q^{(1)} z = 0. \tag{5.23}
$$

Precisely,

$$
\begin{aligned} Q^{(1)} y &= \|y\|_2^2 \|z\|_2^2 y - \langle y, z\rangle^2 y + yz^T yz^T y + zy^T zy^T y - yz^T zy^T y - zy^T yz^T y \\ &= \|y\|_2^2 \|z\|_2^2 y - \langle y, z\rangle^2 y + y\langle y, z\rangle^2 + z\langle y, z\rangle \|y\|_2^2 - y\|z\|_2^2 \|y\|_2^2 - z\|y\|_2^2 \langle y, z\rangle = 0, \end{aligned}
$$

and similarly for z. Assuming that y and z are linearly independent vectors, this means that $\operatorname{rank}(Q^{(1)}) \leq n-2$. On the other hand, $Q^{(1)}$ is defined as an identity matrix minus a rank-2 matrix. This implies that $\operatorname{rank}(Q^{(1)}) = n-2$. However, the linear system $Q^{(1)} x = -c^{(1)}$ is consistent because $\operatorname{rank}([Q^{(1)} c^{(1)}]) = \operatorname{rank}(Q^{(1)})$, which can be seen from the relations (5.4) and (5.7). Hence, the linear system $Q^{(1)} x = -c^{(1)}$ can be solved using the Moore–Penrose inverse,

$$
x = -\big(Q^{(1)}\big)^+ c^{(1)}.
$$

The vector x obtained in this way will be orthogonal to the vectors y and z, because of the form of the matrix $Q^{(1)}$ given in (5.4). The next proposition clarifies this.

PROPOSITION 5.2. *Let $y, z \in \mathbb{R}^n$ be linearly independent vectors and let $Q^{(1)} = Q^{(1)}(y, z)$ be as in relation (5.4). The vector $x = \big(Q^{(1)}\big)^+ c$ is orthogonal to the vectors y and z, for any $c \in \mathbb{R}^n$.*

Proof. First, we show that

$$
Q^{(1)} = \alpha P, \tag{5.24}
$$

where P is an orthogonal projector onto $\{y, z\}^\perp$ and $\alpha = 2(\|y\|_2^2 \|z\|_2^2 - \langle y, z\rangle^2) \neq 0$. Take $u \in \{y, z\}^\perp$. We have $u \perp y$ and $u \perp z$, that is, $y^T u = z^T u = 0$. Then,

$$
Q^{(1)} u = \alpha u + 2(yz^T yz^T u + zy^T zy^T u - yz^T zy^T u - zy^T yz^T u) = \alpha u.
$$

This, together with (5.23), implies (5.24).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Hence, $\left(Q^{(1)}\right)^{+} = (1/\alpha)P$. Using the fact that P is a projector, along with the relations (5.24) and (5.23), it follows that

$$\langle x, y\rangle = -\frac{1}{\alpha}\langle Pc^{(1)}, y\rangle = -\frac{1}{\alpha}\langle c^{(1)}, Py\rangle = -\frac{1}{\alpha^2}\langle c^{(1)}, Q^{(1)}y\rangle = 0,$$

that is, $x \perp y$. In the same way we get $x \perp z$. $\square$

Analogous reasoning holds for the linear systems for y and z.

Now, we can write the algorithm for solving the minimization problem (5.1). The algorithm is based on solving three minimization problems (5.22).

Algorithm 3 Antisymmetry-preserving CP.

Input: $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ antisymmetric
Output: $\widetilde{\mathcal{A}} = \mathcal{A}_6(x, y, z)$
Initialize $x, y, z \in \mathbb{R}^n$ as random vectors.
repeat
 For $c^{(1)}$ as in (5.7) and $Q^{(1)}$ as in (5.4), $x = -\left(Q^{(1)}\right)^{+}c^{(1)}$.
 For $c^{(2)}$ as in (5.8) and $Q^{(2)}$ as in (5.5), $y = -\left(Q^{(2)}\right)^{+}c^{(2)}$.
 For $c^{(3)}$ as in (5.9) and $Q^{(3)}$ as in (5.6), $z = -\left(Q^{(3)}\right)^{+}c^{(3)}$.
until convergence or maximum number of iterations
$\widetilde{\mathcal{A}} = \mathcal{A}_6(x, y, z)$

Note that, as shown in Proposition 5.2, Algorithm 3 results in mutually orthogonal vectors x, y, and z, as a consequence of how the vectors are computed. Since the minimization problem (5.1) does not require orthogonal vectors, this may seem restrictive. Proposition 5.3 justifies the choice of orthogonal vectors.

PROPOSITION 5.3. *For the minimization problem* (5.1)*, the equality*

$$\min_{x,y,z} \|\mathcal{A} - \mathcal{A}_6(x, y, z)\| = \min_{\tilde{x},\tilde{y},\tilde{z} \text{ orthogonal}} \|\mathcal{A} - \mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z})\| \tag{5.25}$$

holds.

Proof. First, note that, if x, y, and z are linearly dependent, then $\mathcal{A}_6(x, y, z) = 0$. That is easily seen from the definition (3.4). If we take a linearly dependent triplet of vectors, e.g., $(\alpha y+\beta z, y, z)$, we have $\mathcal{A}_6(\alpha y+\beta z, y, z) = \alpha\mathcal{A}_6(y, y, z)+\beta\mathcal{A}_6(z, y, z)$ and all summands on the right-hand side will be canceled. Thus, any linearly independent triplet of vectors will give a smaller value of the objective function f defined by the relation (5.2).

The objective function f is invariant under very general transformations. Due to multilinearity, for $\alpha, \beta \in \mathbb{R}$, we have

$$\mathcal{A}_6(x + \alpha y + \beta z, y, z) = \mathcal{A}_6(\alpha y + \beta z, y, z) + \mathcal{A}_6(x, y, z) = \mathcal{A}_6(x, y, z),$$

where $\mathcal{A}_6(\alpha y + \beta z, y, z) = 0$. Let us consider an arbitrary matrix $B = (b_{ij}) \in \mathbb{R}^{3\times 3}$. Using the arguments of multilinearity and antisymmetry as above, we have

$$\begin{aligned}
&\mathcal{A}_6(b_{11}x + b_{12}y + b_{13}z, b_{21}x + b_{22}y + b_{23}z, b_{31}x + b_{32}y + b_{33}z)\\
&\quad= b_{11}b_{22}b_{33}\mathcal{A}_6(x, y, z) + b_{11}b_{23}b_{32}\mathcal{A}_6(x, z, y) + b_{12}b_{21}b_{33}\mathcal{A}_6(y, x, z)\\
&\qquad+ b_{12}b_{23}b_{31}\mathcal{A}_6(y, z, x) + b_{13}b_{21}b_{32}\mathcal{A}_6(z, x, y) + b_{13}b_{22}b_{31}\mathcal{A}_6(z, y, x)\\
&\quad= (b_{11}b_{22}b_{33} - b_{11}b_{23}b_{32} - b_{12}b_{21}b_{33} + b_{12}b_{23}b_{31} + b_{13}b_{21}b_{32} - b_{13}b_{22}b_{31})\mathcal{A}_6(x, y, z)
\end{aligned}$$

$$= \det(B)\mathcal{A}_6(x, y, z).$$

Therefore, if $\det(B) = 1$, we have

$$\|\mathcal{A} - \mathcal{A}_6(b_{11}x + b_{12}y + b_{13}z, b_{21}x + b_{22}y + b_{23}z, b_{31}x + b_{32}y + b_{33}z)\|$$
$$= \|\mathcal{A} - \mathcal{A}_6(x, y, z)\|,$$

and the value of the objective function f stays the same.

Set $V = [x, y, z] \in \mathbb{R}^{n\times 3}$. Take the thin QR decomposition $V = \widetilde{V}R$, such that $\det(R) = 1$ and $\widetilde{V} = [\tilde{x}, \tilde{y}, \tilde{z}] \in \mathbb{R}^{n\times 3}$ has orthogonal columns. Then, following the same reasoning, we have

$$\mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z}) = \mathcal{A}_6(x, y, z),$$

which implies (5.25). $\square$

5.1. Equivalence of Algorithm 3 and the HOPM. Here we are going to show that Algorithm 3 is equivalent to the higher-order power method (HOPM) for unstructured rank-1 approximation and see what implications it has.

Due to multilinearity, the minimization problem (5.25) can be modified into a minimization problem on unitary vectors, $\|x\|_2 = \|y\|_2 = \|z\|_2 = 1$, i.e.,

$$\min_{\substack{\tilde{x},\tilde{y},\tilde{z} \text{ orthonormal,} \\ \lambda\in\mathbb{R}}} \|\mathcal{A} - \lambda\mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z})\|^2, \tag{5.26}$$

so it becomes a minimization problem on the Stiefel manifold. Since the expression in (5.26) does not depend on the basis, it is a minimization problem on the Grassmann manifold, which makes sense as the antisymmetric tensors are connected to the Grassmannians; see, e.g., [23].

We can rewrite (5.26) as

$$\min_{\substack{\tilde{x},\tilde{y},\tilde{z} \text{ orthonormal,} \\ \lambda\in\mathbb{R}}} \left\{\|\mathcal{A}\|^2 - 2\lambda\langle\mathcal{A}, \mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z})\rangle + \lambda^2\|\mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z})\|^2\right\}. \tag{5.27}$$

Set

$$V = \begin{bmatrix}\tilde{x} & \tilde{y} & \tilde{z}\end{bmatrix}. \tag{5.28}$$

Observe that

$$\mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z}) = \mathcal{E} \times_1 V \times_2 V \times_3 V,$$

where $\mathcal{E}$ is given by the relation (3.3). Then,

$$\|\mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z})\|^2 = \|\mathcal{E}\|^2 = 6,$$

because $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ are orthonormal and the Frobenius norm is unitary invariant. In this way, the minimization problem (5.27) is simplified to

$$\min_{\substack{\tilde{x},\tilde{y},\tilde{z} \text{ orthonormal,} \\ \lambda\in\mathbb{R}}} \left\{\|\mathcal{A}\|^2 - 2\lambda\langle\mathcal{A}, \mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z})\rangle + 6\lambda^2\right\}.$$

Take the objective function

$$\tilde{f}(\lambda, \tilde{x}, \tilde{y}, \tilde{z}) = \|\mathcal{A}\|^2 - 2\lambda\langle\mathcal{A}, \mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z})\rangle + 6\lambda^2. \tag{5.29}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

In order to find the optimal λ_* for $\tilde{f}$, we set the partial derivative of $\tilde{f}$ to zero. We have

$$\frac{\partial}{\partial\lambda}\tilde{f}(\lambda,\tilde{x},\tilde{y},\tilde{z}) = 12\lambda - 2\langle\mathcal{A},\mathcal{A}_6(\tilde{x},\tilde{y},\tilde{z})\rangle = 0,$$

that is,

$$\lambda_* = \frac{\langle\mathcal{A},\mathcal{A}_6(\tilde{x},\tilde{y},\tilde{z})\rangle}{6}.$$

It follows from (5.29) that

$$\tilde{f}(\lambda_*,\tilde{x},\tilde{y},\tilde{z}) = \|\mathcal{A}\|^2 - \frac{1}{6}\langle\mathcal{A},\mathcal{A}_6(\tilde{x},\tilde{y},\tilde{z})\rangle^2.$$

Thus, minimizing $\tilde{f}(\lambda_*,\tilde{x},\tilde{y},\tilde{z})$ is equivalent to maximizing $|\langle\mathcal{A},\mathcal{A}_6(\tilde{x},\tilde{y},\tilde{z})\rangle|$ over the Stiefel manifold.

Define the compressed tensor

$$\mathcal{A}_c(V) \coloneqq \mathcal{A}\times_1 V^T \times_2 V^T \times_3 V^T, \tag{5.30}$$

where V is as in relation (5.28). This is a $3\times 3\times 3$ tensor. It is very similar to tensor $\mathcal{E}$, except that in place of 1 and -1 it has $(\mathcal{A}_c(V))_{123}$ and $-(\mathcal{A}_c(V))_{123}$, respectively. Using this tensor we obtain

$$\begin{aligned}|\langle\mathcal{A},\mathcal{A}_6(\tilde{x},\tilde{y},\tilde{z})\rangle| &= |\langle\mathcal{A},\mathcal{E}\times_1 V\times_2 V\times_3 V\rangle| = |\langle\mathcal{A}_c(V),\mathcal{E}\rangle| \\ &= 6|(\mathcal{A}_c(V))_{123}| = \sqrt{6}\|\mathcal{A}_c(V)\|.\end{aligned}$$

In the last equation we used the norm of the compressed tensor, $\|\mathcal{A}_c(V)\|^2 = 6((\mathcal{A}_c(V))_{123})^2$. Therefore, maximization of $|\langle\mathcal{A},\mathcal{A}_6(\tilde{x},\tilde{y},\tilde{z})\rangle|$ is equivalent to maximization of $\|\mathcal{A}_c(V)\|$. This corresponds to the best structure-preserving multilinear rank-r approximation from [3] for $r = 3$.

The problem of finding the best antisymmetric multilinear rank-r approximation is equivalent to the problem of finding the best unstructured rank-1 approximation of an antisymmetric tensor; see [3, Theorem 4.2]. This implies the equivalence between our Algorithm 3 and the HOPM used for finding the best unstructured rank-1 approximation. Finally, the global convergence result for the HOPM given in [30] – namely, the iterates of the ALS algorithm for the HOPM converge to the stationary point of the corresponding objective function – applies to our algorithm as well.

6. Partial antisymmetry. Regarding the antisymmetric tensors, we can ask what happens if a tensor has only partial antisymmetry. We observe order-3 tensors. Note that partially antisymmetric tensors do not need to be cubical.

The tensor $\mathcal{C}\in\mathbb{R}^{n\times n\times m}$ is antisymmetric in modes 1 and 2 if all its frontal slices are antisymmetric. Without loss of generality, we assume that tensor $\mathcal{C}$ is antisymmetric in the first two modes. That is,

$$c_{ijk} = -c_{jik}, \quad 1\le i,j\le n,\ 1\le k\le m. \tag{6.1}$$

Tensors that are antisymmetric in modes 2 and 3, or in modes 1 and 3, are defined correspondingly. The partial antisymmetrizer that results in the antisymmetry in modes 1 and 2 can be defined as the operator $\mathrm{anti}_{1,2}$ such that, for $\mathcal{B}\in\mathbb{R}^{n\times n\times m}$ and $\mathcal{C} = \mathrm{anti}_{1,2}(\mathcal{B})$, we have

$$c_{ijk} = \frac{1}{2}(b_{ijk} - b_{jik}).$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

For a pair of indices (i,j), with $1 \le i < j \le n$, the subtensor $\mathcal{G}$ of $\mathcal{C}$ obtained at the intersection of the ith and jth column, row, and tube is a $2\times 2\times 2$ tensor of the form

$$\mathcal{G}(i_1,i_2,i_3) = \begin{cases} \alpha, & \text{if } (i_1,i_2,i_3)=(1,2,1),\\ -\alpha, & \text{if } (i_1,i_2,i_3)=(2,1,1),\\ \beta, & \text{if } (i_1,i_2,i_3)=(1,2,2),\\ -\beta, & \text{if } (i_1,i_2,i_3)=(2,1,2),\\ 0, & \text{if } i_1=i_2, \end{cases}$$

for $\alpha,\beta\in\mathbb{R}$. Its mode-1 matricization is given by

$$G_{(1)} = \left[\begin{array}{cc|cc} 0 & \alpha & 0 & \beta \\ -\alpha & 0 & -\beta & 0 \end{array}\right].$$

Here, the tensor $\mathcal{G}$ plays the role analogous to the Levi-Civita tensor (3.3) in Section 3.

Analogously to the tensor format (3.4), for three vectors $x,y\in\mathbb{R}^n$ and $z\in\mathbb{R}^m$, we can define an $n\times n\times m$ tensor

$$\mathcal{C}_2(x,y,z) := \frac{1}{2}(x\circ y\circ z - y\circ x\circ z). \tag{6.2}$$

If we take $x=[1,0]^T$, $y=[0,1]^T$, and $z=[\alpha,\beta]^T$, then $\mathcal{C}_2(x,y,z)=\mathcal{G}$. Besides, if $\mathcal{T} = [[x,y,z]]$ is a rank-1 tensor, then $\mathcal{C}_2(x,y,z)=\text{anti}_{1,2}(\mathcal{T})$. Obviously, $\text{rank}(\mathcal{C}_2(x,y,z))\le 2$. For the fixed third index, each slice of $\mathcal{C}_2(x,y,z)$ is a skew-symmetric matrix and, therefore, has an even rank. Hence,

$$\text{rank}(\mathcal{C}_2(x,y,z)) = 2.$$

Considering all this, for a given non-zero tensor $\mathcal{C}\in\mathbb{R}^{n\times n\times m}$ that is antisymmetric in the first two modes, we are looking for its rank-2 approximation $\widetilde{\mathcal{C}}$ of the same structure. Again, we examine two approaches. The first one is analogous to Section 4. In the second approach we find a tensor $\widetilde{\mathcal{C}}=\mathcal{C}_2(x,y,z)$ defined by the vectors $x,y\in\mathbb{R}^n$ and $z\in\mathbb{R}^m$, such that

$$\|\mathcal{C}-\widetilde{\mathcal{C}}\|^2 \to \min. \tag{6.3}$$

6.1. Ignoring the structure. Let $\mathcal{C}\in\mathbb{R}^{n\times n\times m}$ be a tensor with partial antisymmetry. We first approximate $\mathcal{C}$ with a rank-1 tensor $\bar{\mathcal{C}}$ by using the CP-ALS algorithm (Algorithm 1) with $r=1$. Then, we apply the operator $\text{anti}_{1,2}$ on $\bar{\mathcal{C}}$ to get a rank-2 tensor $\widetilde{\mathcal{C}}$ that is antisymmetric in modes 1 and 2. We have

$$\begin{aligned} \bar{\mathcal{C}} &= [[x,y,z]], \quad x,y\in\mathbb{R}^n,\ z\in\mathbb{R}^m,\\ \widetilde{\mathcal{C}} &= \text{anti}_{1,2}(\bar{\mathcal{C}}), \end{aligned}$$

or, equivalently, $\widetilde{\mathcal{C}}=\mathcal{C}_2(x,y,z)$. The algorithm with partial a posteriori antisymmetrization is a simple modification of Algorithm 2.

6.2. Preserving the structure. Now we are going to construct an iterative structure-preserving minimization algorithm. Again, let $\mathcal{C}\in\mathbb{R}^{n\times n\times m}$ be a tensor with partial antisymmetry. We are looking for tensor $\widetilde{\mathcal{C}}\in\mathbb{R}^{n\times n\times m}$ that is a solution of the minimization problem (6.3). In particular, we are looking for vectors $x,y\in\mathbb{R}^n$ and $z\in\mathbb{R}^m$ such that

$$\|\mathcal{C}-\mathcal{C}_2(x,y,z)\|^2 \to \min. \tag{6.4}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 4 CP with partial a posteriori antisymmetrization.

Input: $\mathcal{C} \in \mathbb{R}^{n\times n\times m}$ antisymmetric in modes 1 and 2
Output: $\widetilde{\mathcal{C}} = \mathcal{C}_2(x, y, z)$
Apply Algorithm 1 on $\mathcal{A}$ with $r = 1$ to obtain $x, y \in \mathbb{R}^n$, $z \in \mathbb{R}^m$
$\widetilde{\mathcal{C}} = \mathcal{C}_2(x, y, z)$

We set

$$v = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^{2n+m},$$

and define the objective function $g\colon \mathbb{R}^{2n+m} \to \mathbb{R}$ as

$$g(v) = 2\|\mathcal{C} - \mathcal{C}_2(x, y, z)\|^2. \tag{6.5}$$

We formulate the ALS algorithm based on three minimization problems:

$$\min_x g(v), \quad \min_y g(v), \quad \min_z g(v).$$

To this end, we need Theorem 6.1. Before the statement of the theorem, we define the appropriate objects: matrices $Q^{(1)} = Q^{(1)}(y, z), Q^{(2)} = Q^{(2)}(x, z) \in \mathbb{R}^{n\times n}$,

$$Q^{(1)} = 2\|y\|_2^2\|z\|_2^2 I_n - 2yy^T\|z\|_2^2, \tag{6.6}$$

$$Q^{(2)} = 2\|x\|_2^2\|z\|_2^2 I_n - 2xx^T\|z\|_2^2; \tag{6.7}$$

vectors $b^{(1)} = b^{(1)}(y, z), b^{(2)} = b^{(2)}(x, z) \in \mathbb{R}^n$, and $b^{(3)} = b^{(3)}(x, y) \in \mathbb{R}^m$,

$$b^{(1)} = -4\mathcal{C} \times_2 y^T \times_3 z^T, \tag{6.8}$$

$$b^{(2)} = -4\mathcal{C} \times_2 x^T \times_3 z^T, \tag{6.9}$$

$$b^{(3)} = -2(\mathcal{C} \times_1 x^T \times_2 y^T - \mathcal{C} \times_1 y^T \times_2 x^T), \tag{6.10}$$

and numbers $q^{(3)} = q^{(3)}(x, y)$ and $d \in \mathbb{R}$,

$$q^{(3)} = \|xy^T - yx^T\|_2^2, \tag{6.11}$$

$$d = 2\|\mathcal{C}\|^2. \tag{6.12}$$

THEOREM 6.1. *The function g defined by (6.5) can be written as*

$$g(v) = d + (b^{(1)})^T x + \frac{1}{2} x^T Q^{(1)} x \tag{6.13}$$

$$= d + (b^{(2)})^T y + \frac{1}{2} y^T Q^{(2)} y \tag{6.14}$$

$$= d + (b^{(3)})^T z + \frac{1}{2} q^{(3)} z^T z, \tag{6.15}$$

for $Q^{(1)}, Q^{(2)} \in \mathbb{R}^{n\times n}$, $b^{(1)}, b^{(2)} \in \mathbb{R}^n$, $b^{(3)} \in \mathbb{R}^m$, and $q^{(3)} \in \mathbb{R}$ defined by the relations (6.6)–(6.12).

Proof. We start by writing the function g as

$$g(v) = 2\|\mathcal{C}\|^2 - 2\langle \mathcal{C}, x \circ y \circ z - y \circ x \circ z\rangle + \frac{1}{2}\|x \circ y \circ z - y \circ x \circ z\|^2$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$$= 2g_1(v) - 2g_2(v) + \frac{1}{2}g_3(v),$$

for

$$g_1(v) = \|\mathcal{C}\|^2,$$
$$g_2(v) = \langle \mathcal{C}, x \circ y \circ z - y \circ x \circ z \rangle, \tag{6.16}$$
$$g_3(v) = \|x \circ y \circ z - y \circ x \circ z\|^2. \tag{6.17}$$

Function g_2 can be written as

$$\begin{aligned} g_2(v) &= \sum_{i,j=1}^{n} \sum_{k=1}^{m} c_{ijk}(x_i y_j z_k - y_i x_j z_k) \\ &= \sum_{i=1}^{n} x_i \left(\sum_{j=1}^{n} \sum_{k=1}^{m} c_{ijk} y_j z_k \right) + \sum_{j=1}^{n} x_j \left(\sum_{i=1}^{n} \sum_{k=1}^{m} (-c_{ijk}) y_i z_k \right). \end{aligned}$$

Using the partial antisymmetry property (6.1), after renaming the indices we get

$$g_2(v) = 2 \sum_{i=1}^{n} x_i \left(\sum_{j=1}^{n} \sum_{k=1}^{m} c_{ijk} y_j z_k \right).$$

For the function g_3 we have

$$\begin{aligned} g_3(v) &= \sum_{i,j=1}^{n} \sum_{k=1}^{m} (x_i y_j z_k - x_j y_i z_k)^2 \\ &= \sum_{i=1}^{n} x_i^2 \left(\sum_{j=1}^{n} \sum_{k=1}^{m} y_j^2 z_k^2 \right) - 2 \sum_{i,j=1}^{n} x_i x_j y_i y_j \left(\sum_{k=1}^{m} z_k^2 \right) + \sum_{j=1}^{n} x_j^2 \left(\sum_{i=1}^{n} \sum_{k=1}^{m} y_i^2 z_k^2 \right) \\ &= 2 \sum_{i=1}^{n} x_i^2 \|y\|_2^2 \|z\|_2^2 - 2 \sum_{i,j=1}^{n} x_i x_j y_i y_j \|z\|_2^2 \\ &= \sum_{i=1}^{n} x_i^2 (2\|y\|_2^2 \|z\|_2^2 - 2y_i^2 \|z\|_2^2) + \sum_{\substack{i,j=1 \\ i \neq j}}^{n} x_i x_j (-2 y_i y_j \|z\|_2^2). \end{aligned}$$

In the same way as in the proof of Theorem 5.1, we set

$$\begin{aligned} d &= 2g_1(v), \\ (b^{(1)})^T x &= -2g_2(v), \\ \frac{1}{2} x^T Q^{(1)} x &= \frac{1}{2} g_3(v), \end{aligned}$$

where

$$b_i^{(1)} = -4 \sum_{j=1}^{n} \sum_{k=1}^{m} c_{ijk} y_j z_k, \quad 1 \le i \le n, \tag{6.18}$$

and

$$q_{ii}^{(1)} = 2\|y\|^2 \|z\|^2 - 2y_i^2 \|z\|^2,$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$$q_{ij}^{(1)} = -2y_iy_j\|z\|^2, \quad 1 \leq i,j \leq n,\ i \neq j. \tag{6.19}$$

The vector $b^{(1)}$ from (6.18) can be written in the more compact form (6.8) and matrix $Q^{(1)}$ from (6.19) is equivalent to (6.6), while d is like in the relation (6.12). This is how we get the assertion (6.13).

With different regrouping of the summands in the relations (6.16) and (6.17) we get equation (6.14) with $b^{(2)}$ and $Q^{(2)}$ as in (6.9) and (6.7), respectively.

To get equation (6.15) we write

$$g_2(v) = \sum_{k=1}^{m} z_k \left(\sum_{i,j=1}^{n} c_{ijk}(x_iy_j - y_ix_j) \right)$$

and

$$g_3(v) = \sum_{k=1}^{m} z_k^2 \left(\sum_{i,j=1}^{n} (x_iy_j - x_jy_i)^2 \right).$$

Then, we set

$$b_k^{(3)} = -2 \sum_{i,j=1}^{n} c_{ijk}(x_iy_j - y_ix_j), \quad 1 \leq k \leq m,$$

$$q^{(3)} = \sum_{i,j=1}^{n} (x_iy_j - x_jy_i)^2 = \|xy^T - yx^T\|_2^2.$$

The compact form of the vector $b^{(3)}$ corresponds to (6.10). $\square$

Therefore, as in Section 5, our algorithm is based on finding the solutions of the minimization problems

$$\begin{cases} \min\limits_x d + (b^{(1)})^T x + \dfrac{1}{2} x^T Q^{(1)} x, \\ \min\limits_y d + (b^{(2)})^T y + \dfrac{1}{2} y^T Q^{(2)} y, \\ \min\limits_z d + (b^{(3)})^T z + \dfrac{1}{2} q^{(3)} z^T z. \end{cases}$$

Those solutions are obtained, respectively, from the following equations:

$$\begin{cases} Q^{(1)}x = -b^{(1)}, \\ Q^{(2)}y = -b^{(2)}, \\ \quad z = -\dfrac{1}{q^{(3)}} b^{(3)}. \end{cases}$$

The situation regarding these linear systems is similar to that for the fully antisymmetric case. Matrices $Q^{(1)}$ and $Q^{(2)}$ are not of full rank. From their definitions (6.6) and (6.7) we see that both are given as the identity minus a rank-1 matrix and

$$Q^{(1)}x = 0, \quad Q^{(2)}y = 0.$$

Thus, $\text{rank}(Q^{(1)}) = \text{rank}(Q^{(2)}) = n - 1$. Still, we have $\text{rank}([Q^{(1)} b^{(1)}]) = \text{rank}(Q^{(1)})$ and $\text{rank}([Q^{(2)} b^{(2)}]) = \text{rank}(Q^{(2)})$, so the linear systems are consistent and can be solved using

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 5 CP preserving partial antisymmetry.

Input: $\mathcal{C} \in \mathbb{R}^{n\times n\times m}$ antisymmetric in modes 1 and 2
Output: $\widetilde{\mathcal{C}} = \mathcal{C}_2(x, y, z)$
Initialize $x, y \in \mathbb{R}^n$, $z \in \mathbb{R}^m$ as random vectors.
repeat
  For $b^{(1)}$ as in (6.8) and $Q^{(1)}$ as in (6.6), $x = -(Q^{(1)})^+ b^{(1)}$.
  For $b^{(2)}$ as in (6.9) and $Q^{(2)}$ as in (6.7), $y = -(Q^{(2)})^+ b^{(2)}$.
  For $b^{(3)}$ as in (6.10) and $q^{(3)}$ as in (6.11), $z = -b^{(3)}/q^{(3)}$.
until convergence or maximum number of iterations
$\widetilde{\mathcal{C}} = \mathcal{C}_2(x, y, z)$

the Moore–Penrose inverse. Additionally, we get that the vectors x and y must be orthogonal. Note that, for $x \neq y$, we have $q^{(3)} \neq 0$ and z is well defined.

The algorithm for solving the minimization problem (6.4) is very similar to Algorithm 3.

As in the fully antisymmetric case, we can additionally observe that $\mathcal{C}_2(x, y, z) = 0$ if x and y are linearly dependent and

$$\mathcal{C}_2(b_{11}x + b_{12}y, b_{21}x + b_{22}y, z) = \det B\mathcal{C}_2(x, y, z), \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

Then we can rescale our optimization problem such that we are looking for

$$\min_{\substack{\|\tilde{x}\|=\|\tilde{y}\|=\|\tilde{z}\|=1,\\ x\perp y,\ \lambda\in\mathbb{R}}} \{\|\mathcal{C} - \lambda\mathcal{C}_2(\tilde{x}, \tilde{y}, \tilde{z})\|^2\}.$$

Set

$$\tilde{g}(\lambda, \tilde{x}, \tilde{y}, \tilde{z}) = \|\mathcal{C}\|^2 - 2\lambda\langle \mathcal{C}, \mathcal{C}_2(\tilde{x}, \tilde{y}, \tilde{z})\rangle + \lambda^2\|\mathcal{C}_2(\tilde{x}, \tilde{y}, \tilde{z})\|^2.$$

From the shape of $\mathcal{C}_2$ and the fact that $\|\tilde{x}\| = \|\tilde{y}\| = \|\tilde{z}\| = 1$ and $x \perp y$, after a short calculation we get $\|\mathcal{C}_2(\tilde{x}, \tilde{y}, \tilde{z})\|^2 = \frac{1}{2}$. Thus,

$$\tilde{g}(\lambda, \tilde{x}, \tilde{y}, \tilde{z}) = \|\mathcal{C}\|^2 - 2\lambda\langle \mathcal{C}, \mathcal{C}_2(\tilde{x}, \tilde{y}, \tilde{z})\rangle + \frac{1}{2}\lambda^2.$$

The optimal λ for $\tilde{g}$ is

$$\lambda_* = 2\langle \mathcal{C}, \mathcal{C}_2(\tilde{x}, \tilde{y}, \tilde{z})\rangle$$

and

$$\tilde{g}(\lambda_*, \tilde{x}, \tilde{y}, \tilde{z})) = \|\mathcal{C}\|^2 - 2\langle \mathcal{C}, \mathcal{C}_2(\tilde{x}, \tilde{y}, \tilde{z})\rangle^2.$$

Therefore, minimizing $\tilde{g}(\lambda_*, \tilde{x}, \tilde{y}, \tilde{z}))$ is equivalent to maximizing $|\langle \mathcal{C}, \mathcal{C}_2(\tilde{x}, \tilde{y}, \tilde{z})\rangle|$.

Now we can set

$$W = \begin{bmatrix} \tilde{x} & \tilde{y} \end{bmatrix}$$

and define the compressed matrix

$$C_c(W, \tilde{z}) \coloneqq \mathcal{C} \times_1 W^T \times_2 W^T \times_3 \tilde{z}^T, \tag{6.20}$$

which is an analogue of the compressed tensor (5.30). The matrix $C_c(W,\tilde{z})$ is a 2×2 skew-symmetric matrix

$$\begin{bmatrix} 0 & (C_c(W,\tilde{z}))_{12} \\ -(C_c(W,\tilde{z}))_{12} & 0 \end{bmatrix},$$

where

$$|(C_c(W,\tilde{z}))_{12}| = |\mathcal{C} \times_1 \tilde{x}^T \times_2 \tilde{y}^T \times_3 \tilde{z}^T|. \tag{6.21}$$

Moreover, we can write

$$\mathcal{C}_2(\tilde{x},\tilde{y},\tilde{z}) = M \times_1 W \times_2 W \times_3 \tilde{z},$$

for $M = \begin{bmatrix} 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix}$. It follows that

$$|\langle \mathcal{C}, \mathcal{C}_2(\tilde{x},\tilde{y},\tilde{z})\rangle| = |\langle C_c(W,\tilde{z}), M\rangle| = \frac{\sqrt{2}}{2}\|C_c(W,\tilde{z})\|_F$$

and we conclude that maximization of $|\langle \mathcal{C}, \mathcal{C}_2(\tilde{x},\tilde{y},\tilde{z})\rangle|$ is equivalent to maximization of $\|C_c(W,\tilde{z})\|_F$.

Maximization of $\|C_c(W,\tilde{z})\|_F$ corresponds to the multilinear rank-$(2,2,m)$ structure-preserving approximation of $\mathcal{C}$. Similarly as in [3] for the best antisymmetric multilinear rank-r approximation, we can establish an equivalence between the best partially antisymmetric multilinear rank-$(2,2,m)$ approximation and the best unstructured rank-1 approximation of a partially antisymmetric tensor.

PROPOSITION 6.2. *Let $\mathcal{C} \in \mathbb{R}^{n\times n\times m}$ be a partially antisymmetric tensor. Then*

$$\begin{aligned} &\max\{\|\mathcal{C} \times_1 U^T \times_2 U^T \times_3 z^T\| \,:\, U \in \mathbb{R}^{n\times 2},\ U^TU = I_2,\ \|z\|_2 = 1\} \\ &= \sqrt{2}\max\{|\mathcal{C} \times_1 u_1^T \times_2 u_2^T \times_3 z^T| \,:\, \\ &\qquad \|u_1\|_2 = \|u_2\|_2 = \|z\|_2 = 1,\ [u_1u_2]^T[u_1u_2] = I_2\} \end{aligned} \tag{6.22}$$

$$= \sqrt{2}\max\{|\mathcal{C} \times_1 v_1^T \times_2 v_2^T \times_3 z^T| \,:\, \|v_1\|_2 = \|v_2\|_2 = \|z\|_2 = 1\}. \tag{6.23}$$

Proof. Take $\alpha = \mathcal{C} \times_1 u_1^T \times_2 u_2^T \times_3 z^T$. From the relations (6.20) and (6.21) we see that, for every partially antisymmetric tensor $\mathcal{C}$ and for $U = [u_1u_2]$,

$$\|\mathcal{C} \times_1 U^T \times_2 U^T \times_3 z^T\|^2 = \left\|\begin{bmatrix} 0 & \alpha \\ -\alpha & 0 \end{bmatrix}\right\|_F^2 = 2\alpha^2,$$

which proves (6.22).

Obviously, expression (6.22) is less than or equal to (6.23). Take the vectors v_1, v_2, and z that maximize (6.23). There is an upper-triangular 2×2 matrix R such that $|r_{11}| \le 1$, $|r_{22}| \le 1$, and

$$\begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} R$$

is the thin QR decomposition of $[v_1v_2]$. Using the antisymmetry in two modes, we have

$$\begin{aligned} |\mathcal{C} \times_1 v_1^T \times_2 v_2^T \times_3 z^T| &= |\mathcal{C} \times_1 r_{11}u_1^T \times_2 (r_{12}u_1^T + r_{22}u_2^T) \times_3 z^T| \\ &= |\mathcal{C} \times_1 r_{11}u_1^T \times_2 r_{22}u_2^T \times_3 z^T| \end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$$= |r_{11}r_{22}|\,|\mathcal{C} \times_1 u_1^T \times_2 u_2^T \times_3 z^T| \leq |\mathcal{C} \times_1 u_1^T \times_2 u_2^T \times_3 z^T|$$

This proves that the value of (6.22) is equal to the value of (6.23). ◻

Therefore, following the previous discussion and the result of Proposition 6.2, we have obtained the equivalence between our Algorithm 5 and the best unstructured rank-1 approximation. Then, the same as in the fully antisymmetric case, the convergence result for the HOPM from [30] holds.

7. Numerical experiments. We provide numerical examples for the comparison of the CP rank-1 approximation with a posteriori antisymmetrization (Algorithm 2) and the antisymmetry-preserving CP (Algorithm 3). Additionally, for the sake of completeness, we compare these algorithms with the CP-ALS algorithm (Algorithm 1) with $r = 6$, the algorithm that does not preserve antisymmetry. As we will show, antisymmetry-preserving CP outperforms CP with a posteriori antisymmetrization in terms of accuracy, which was expected, but also in execution time, while CP-ALS has been shown to be much slower than the other two algorithms, and it also completely destroys the antisymmetric property.

All the algorithms are implemented and tested in the Julia programming language [4], version 1.8.1, on a personal computer, with the `BenchmarkTools` [6] package, used for determining the execution times of the algorithms (function `@btime`) and the `TensorToolbox` [25] package for tensor calculations.

For a given tensor $\mathcal{A}$ and an approximation $\widetilde{\mathcal{A}}$, we are looking at the relative error $\|\mathcal{A} - \widetilde{\mathcal{A}}\|/\|\mathcal{A}\|$. We run the CP-ALS algorithm, both on its own and within CP with a posteriori antisymmetrization with tolerance 10^{-8}, and we stop the antisymmetry-preserving CP algorithm when either the relative error or the difference between relative errors in two consecutive iterations falls below 10^{-8}.

7.1. Example 1. First we generate an antisymmetric tensor $\mathcal{A}$ of size $n \times n \times n$ and rank 6, by randomly selecting three vectors x, y, z of size n and defining $\mathcal{A} = 6\mathcal{A}_6(x, y, z)$, where $\mathcal{A}_6(x, y, z)$ is defined in (3.4). In this example we know that $\mathcal{A}$ has the proposed structure. We evaluate and compare the accuracy and the speed of our algorithms. The results for different n are presented in Table 7.1. The best result in each column is shown in bold.

TABLE 7.1
Evaluation of the CP algorithm with a posteriori antisymmetrization (CP+antisym – Algorithm 2), antisymmetry-preserving CP (antisymCP – Algorithm 3), and CP-ALS with $r = 6$ (Algorithm 1) in terms of the relative error $\|\mathcal{A} - \widetilde{\mathcal{A}}\|/\|\mathcal{A}\|$ and execution times obtained by function `@btime`.

	$n = 10$		$n = 25$		$n = 50$	
	error	time	error	time	error	time
CP+antisym	0.8333	224 µs	0.8333	905.9 µs	0.8333	**3.983 ms**
antisymCP	$\mathbf{8.21{\times}10^{-16}}$	**69.9 µs**	$\mathbf{1.34{\times}10^{-15}}$	**502.5 µs**	$\mathbf{1.66{\times}10^{-15}}$	8.283 ms
CP-ALS	$5.27{\times}10^{-6}$	8.472 ms	$1.998{\times}10^{-7}$	26.282 ms	$8.43{\times}10^{8}$	187.625 ms

Even though the execution time of CP with a posteriori antisymmetrization is comparable to that for antisymmetry-preserving CP, by first approximating with a non-antisymmetric tensor of rank 1, CP with a posteriori antisymmetrization loses the underlying structure, and results in an approximation with large error. CP-ALS manages to find a good non-antisymmetric approximation, but it requires much more time, so disregarding the antisymmetric property did not help either with accuracy or with execution times. Overall, antisymmetry-preserving CP achieves best results.

Here, as in the following examples, the initial vectors in Algorithm 3 are taken as random vectors. If we initialize the algorithm using a higher-order singular value decomposition

ETNA
Kent State University and
Johann Radon Institute (RICAM)

(HOSVD), the number of iterations decreases, but the execution time increases because of the additional time needed to perform the HOSVD.

7.2. Example 2. Now we construct an antisymmetric tensor element-wise as

$$\begin{aligned}\mathcal{A}(i,j,k) &= \sin(x_i)\sin(y_j)\sin(z_k) + \sin(y_i)\sin(z_j)\sin(x_k) + \sin(z_i)\sin(x_j)\sin(y_k)\\ &\quad - \sin(y_i)\sin(x_j)\sin(z_k) - \sin(x_i)\sin(z_j)\sin(y_k) + \sin(z_i)\sin(y_j)\sin(x_k),\end{aligned}$$

where x_i, y_j, and z_k are sets of n equidistant points on the intervals $[0,1]$, $[2,10]$, and $[1,3]$, respectively. This type of tensor appears in signal processing applications. The accuracy and speed of our algorithms for different n are presented in Table 7.2. The best result in each column is shown in bold.

TABLE 7.2

Evaluation of the CP algorithm with a posteriori antisymmetrization (CP+antisym – Algorithm 2), antisymmetry-preserving CP (antisymCP – Algorithm 3), and CP-ALS with $r = 6$ (Algorithm 1) in terms of the relative error $\|\mathcal{A} - \widetilde{\mathcal{A}}\|/\|\mathcal{A}\|$ and execution times obtained by function `@btime`.

	$n = 10$		$n = 25$		$n = 50$	
	error	time	error	time	error	time
CP+antisym	0.8333	220.9 µs	0.8333	912.7 µs	0.8333	**3.95 ms**
antisymCP	$\mathbf{7.55{\times}10^{-16}}$	**111.5 µs**	$\mathbf{9.2{\times}10^{-16}}$	**3.439 µs**	$\mathbf{1.575{\times}10^{-15}}$	26.898 ms
CP-ALS	$4.02{\times}10^{-7}$	7.659 ms	$3.91{\times}10^{-9}$	29.453 ms	$8.492{\times}10^{-7}$	86.045 ms

Similarly as in Example 7.1, antisymmetry-preserving CP beats the other two methods in terms of accuracy and speed of getting an accurate solution.

In the next two examples we use tensors of smaller size, because the ranks of those tensors increase with size, and, since we are approximating by a rank-6 tensor, we want to use tensors for which it makes sense to do this type of approximation.

7.3. Example 3. Now we generate an antisymmetric tensor that does not necessarily have the structure (3.4), by discretizing the function $f(x,y,z) = \exp(x^2 + 2y^2 + 3z^2)$ on a grid $\xi_i = (i-1)/(n-1)$, with $i = 1,\dots,n$, and then applying the antisymmetrizer (3.2). We test for different values of n and show the results in Table 7.3. Again, the best result in each column is shown in bold.

TABLE 7.3

Evaluation of the CP algorithm with a posteriori antisymmetrization (CP+antisym – Algorithm 2), antisymmetry-preserving CP (antisymCP – Algorithm 3), and CP-ALS with $r = 6$ (Algorithm 1) in terms of the relative error $\|\mathcal{A} - \widetilde{\mathcal{A}}\|/\|\mathcal{A}\|$ and execution times obtained by function `@btime`.

	$n = 3$		$n = 5$		$n = 7$	
	error	time	error	time	error	time
CP+antisym	0.8333	185 µs	0.8339	260.2 µs	0.8345	276.6 µs
antisymCP	$\mathbf{1.61{\times}10^{-14}}$	**17.1 µs**	**0.0557**	**87.4 µs**	**0.0802**	**130.4 µs**
CP-ALS	$2.55{\times}10^{-5}$	9.875 ms	**0.0557**	26.282 ms	**0.0802**	4.538 ms

Antisymmetry-preserving CP achieves the best execution times. When the tensor can be well approximated by the CP approximation of the form (3.4), it also achieves the best accuracy ($n = 3$). Otherwise, it results in the same error as CP-ALS, but much lower execution times ($n = 5, 7$).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

7.4. Example 4. We generate a random tensor of size $n \times n \times n$ and antisymmetrize it with (3.2). We compare the three algorithms and present the results in Table 7.4. The best result in each column is shown in bold.

TABLE 7.4
Evaluation of the CP algorithm with a posteriori antisymmetrization (CP+antisym – Algorithm 2), antisymmetry-preserving CP (antisymCP – Algorithm 3), and CP-ALS with $r = 6$ (Algorithm 1) in terms of the relative error $\|\mathcal{A} - \widetilde{\mathcal{A}}\|/\|\mathcal{A}\|$ and execution times obtained by function `@btime`.

	$n=3$		$n=5$		$n=7$	
	error	time	error	time	error	time
CP+antisym	0.8333	184.6 µs	0.8546	336.5 µs	0.9242	743.3 µs
antisymCP	$\mathbf{3.616{\times}10^{-16}}$	**17 µs**	0.3432	**139.9 µs**	0.723	**493.2 µs**
CP-ALS	$8.11{\times}10^{-8}$	14.364 ms	**0.2716**	20.172 ms	**0.6393**	421.051 ms

Similarly as in the previous example, when a tensor can be well approximated by CP decomposition with six summands (here for $n = 3$), antisymmetry-preserving CP achieves the best results. For $n = 5, 7$, antisymmetry-preserving CP gives somewhat worse results than CP-ALS in terms of accuracy, but still gives the approximation in much shorter times, and CP-ALS does not preserve the antisymmetry. Note that the rank of a random antisymmetric tensor is much higher than six. This is the reason why all approximations produce high relative error.

7.5. Example 5. Partial antisymmetry. For the partial antisymmetry, we compare Algorithm 4, CP with partial a posteriori antisymmetrization, and Algorithm 5, CP preserving partial antisymmetry, with standard CP-ALS (Algorithm 1) with $r = 2$, which ignores the structure.

Here, regardless of how we construct the tensor $\mathcal{A}$, all methods give approximately the same error. Again, the CP preserving partial antisymmetry stands out in terms of execution times. We present results in Table 7.5, with tensors $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ defined as follows:

- $\mathcal{A}_1$ is an $8 \times 8 \times 10$ tensor constructed by randomly selecting vectors x, y, z of sizes $8, 8, 10$, respectively, and setting $\mathcal{A} = 2\mathcal{C}_2$, where $\mathcal{C}_2$ is defined in (6.2).
- $\mathcal{A}_2$ is a $5\times5\times7$ tensor constructed from the function the same way as in Example 7.3.
- $\mathcal{A}_3$ is a $5 \times 5 \times 4$ tensor generated by partially antisymmetrizing a tensor with randomly selected elements, using the $\mathrm{anti}_{1,2}$ operator.

TABLE 7.5
Evaluation of the CP algorithm with partial a posteriori antisymmetrization (CP+pantisym – Algorithm 5), antisymmetry-preserving partial CP (pantisymCP – Algorithm 5), and CP-ALS with $r = 2$ (Algorithm 1) in terms of the relative error $\|\mathcal{A} - \widetilde{\mathcal{A}}\|/\|\mathcal{A}\|$ and execution times obtained by function `@btime`.

	$\mathcal{A}_1$		$\mathcal{A}_2$		$\mathcal{A}_3$	
	error	time	error	time	error	time
CP+pantisym	$1.88{\times}10^{-16}$	202.7 µs	0.1001	269.4 µs	0.7175	569.2 µs
pantisymCP	$5.832{\times}10^{-16}$	**30.2 µs**	0.1001	**53.70 µs**	0.7175	**106.6 µs**
CP-ALS	$6.774{\times}10^{-16}$	1.051 ms	0.1001	1.282 ms	0.7175	3.026 ms

8. Conclusion. We have described an antisymmetric tensor format $\mathcal{A}_6(x, y, z)$ determined by only three vectors, $x, y, z \in \mathbb{R}^n$. For any n, tensors of the form $\mathcal{A}_6(x, y, z)$ have rank at most six. We developed an ALS algorithm for structure-preserving low-rank approximation of an antisymmetric tensor $\mathcal{A}$ by a tensor of the form $\widetilde{\mathcal{A}} = \mathcal{A}_6(x, y, z)$. In order to

ETNA
Kent State University and
Johann Radon Institute (RICAM)

obtain our algorithm, we wrote the objective function as three different quadratic forms, given explicitly, one for each mode. The algorithm works in such a way that, in each micro-iteration, a quadratic optimization problem for the corresponding tensor mode is solved.

We showed that our minimization problem

$$\|\mathcal{A} - \mathcal{A}_6(x, y, z)\| \to \min$$

for $x, y, z \in \mathbb{R}^n$ can be viewed as a minimization problem for orthonormal vectors $\tilde{x}, \tilde{y}, \tilde{z} \in \mathbb{R}^n$,

$$\|\mathcal{A} - \mathcal{A}_6(\tilde{x}, \tilde{y}, \tilde{z})\| \to \min .$$

Further, we demonstrated that this minimization problem is equivalent to the maximization problem

$$\|\mathcal{A} \times_1 V^T \times_2 V^T \times_3 V^T\| \to \max,$$

where $V \in \mathbb{R}^{n\times 3}$ is a matrix with orthonormal columns. The prior maximization problem corresponds to the problem of the best multilinear low-rank approximation of antisymmetric tensors. Using the result from [3] stating that antisymmetric multilinear low-rank approximation is equivalent to the best unstructured rank-1 approximation, we were able relate our algorithm to the HOPM. Therefore, the global convergence results for the HOPM from [30] apply here.

For tensors with partial antisymmetry, we established a partially antisymmetric tensor format $\mathcal{C}_2(x, y, z)$ determined by three vectors, $x, y \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$. Tensors of the form $\mathcal{C}_2(x, y, z)$ have rank 2. We created a similar ALS algorithm for structure-preserving rank-2 approximation of a partially antisymmetric tensor $\mathcal{C}$ by a tensor of the form $\widetilde{C} = \mathcal{C}_2(x, y, z)$. Analogously to the fully antisymmetric case, we verified that the algorithm in question is equivalent to the HOPM.

The method described in this paper can be generalized to solve the approximation problem for different antisymmetric structures. Given that the target format can be written as a sum of multilinear terms, the underlying linearity in each mode would lead to quadratic optimization problems which would be handled in the same way, with different coefficient matrices and vectors. For example, instead of antisymmetric rank-6 approximation, this way, one could find antisymmetric rank-$6r$ approximation represented by $3r$ vectors. The paper limited its scope to order-3 tensors. For antisymmetric order-d tensors, analogous rank-$d!r$ approximation would be represented by dr vectors.

Acknowledgments. The authors would like to thank the anonymous referees, whose insightful remarks improved the paper, especially for the comments regarding orthogonality of the resulting vectors and the equivalence to the HOPM.

REFERENCES

[1] E. Acar, D. M. Dunlavy, and T. G. Kolda, *A scalable optimization approach for fitting canonical tensor decompositions*, J. Chemometrics, 25 (2011), pp. 67–86.

[2] E. Acar and B. Yener, *Unsupervised multiway data analysis: a literature survey*, IEEE Trans. Knowledge Date Engrg., 21 (2009), pp. 6–20.

[3] E. Begović Kovač and D. Kressner, *Structure-preserving low multilinear rank approximation of antisymmetric tensors*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 967–983.

[4] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, *Julia: a fresh approach to numerical computing*, SIAM Rev., 59 (2017), pp. 65–98.

[5] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart–Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[6] J. Chen and J. Revels, *Robust benchmarking in noisy environments*, Preprint on arXiv, 2016. https://arxiv.org/abs/1608.04295
[7] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.
[8] P. Comon, X. Luciani, and A. L. F. de Almeida, *Tensor decompositions, alternating least squares and other tales*, J. Chemometrics, 23 (2009), pp. 393–405.
[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
[10] ———, *Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl., 26 (2004/05), pp. 295–327.
[11] M. Espig, W. Hackbusch, and A. Khachatryan, *On the convergence of alternating least squares optimisation in tensor format representations*, Preprint on arXiv, 2015. https://arxiv.org/abs/1506.00062
[12] H. Goldstein, C. P. Poole, and J. L. Safko, *Classical Mechanics*, 3rd ed., Pearson, London, 2001.
[13] L. Grasedyck, M. Kluge, and S. Krämer, *Variants of alternating least squares tensor completion in the tensor train format*, SIAM J. Sci. Comput., 37 (2015), pp. A2424–A2450.
[14] W. H. Greub, *Multilinear Algebra*, Springer, New York, 1967.
[15] W. Hackbusch, *On the representation of symmetric and antisymmetric tensors*, in Contemporary Computational Mathematics—A Celebration of the 80th birthday of Ian Sloan. Vol. 1, 2, J. Dick, F. Y. Kuo, and H. Woźniakowski, eds., Springer, Cham, 2018, pp. 483–515.
[16] R. A. Harshman, *Foundations of the Parafac procedure: models and conditions for an "explanatory" multi-modal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.
[17] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.
[18] ———, *Multilple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., 7 (1927), pp. 39–79.
[19] D. Hong, T. G. Kolda, and J. A. Duersch, *Generalized canonical polyadic tensor decomposition*, SIAM Rev., 62 (2020), pp. 133–163.
[20] F. Hummel, T. Tsatsoulis, and A. Grüneis, *Low rank factorization of the Coulomb integrals for periodic coupled cluster theory*, J. Chem. Phys., 146 (2017), Paper No. 124105 (11 pages).
[21] T. G. Kolda, *Numerical optimization for symmetric tensor decomposition*, Math. Program., 151 (2015), pp. 225–248.
[22] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
[23] J. M. Landsberg, *Tensors: Geometry and Applications*, American Mathematical Society, Providence, 2012.
[24] R. Minster, I. Viviano, X. Liu, and G. Ballard, *CP decomposition for tensors via alternating least squares with QR decomposition*, Numer. Linear Algebra Appl., 30 (2023), Paper No. e2511 (17 pages).
[25] L. Periša, *Julia tensor toolbox*, Software, 2019.
[26] K. F. Riley, M. P. Hobson, and S. J. Bence, *Mathematical Methods for Physics and Engineering*, 3rd ed., Cambridge University Press, Cambridge, 2006.
[27] S. Szalay, M. Pfeffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and O. Legeza, *Tensor product methods and entanglement optimization for ab initio quantum chemistry*, Int. J. Quantum Chem., 115 (2015), pp. 1342–1391.
[28] P.-H. Tsai, P. Fischer, and E. Solomonik, *Accelerating the Galerkin reduced-order model with the tensor decomposition for turbulent flows*, Preprint on arXiv, 2023. https://arxiv.org/abs/2311.03694
[29] A. Uschmajew, *Local convergence of the alternating least squares algorithm for canonical tensor approximation*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 639–652.
[30] ———, *A new convergence proof for the higher-order power method and generalizations*, Pac. J. Optim., 11 (2015), pp. 309–321.