

SHIFTED AND EXTRAPOLATED POWER METHODS FOR TENSOR ℓ^p -EIGENPAIRS*

STEFANO CIPOLLA[†], MICHELA REDIVO-ZAGLIA[†], AND FRANCESCO TUDISCO[‡]

Abstract. This work is concerned with the computation of ℓ^p -eigenvalues and eigenvectors of square tensors with d modes. In the first part we propose two possible shifted variants of the popular (higher-order) power method, and, when the tensor is entry-wise nonnegative with a possibly reducible pattern and p is strictly larger than the number of modes, we prove convergence of both schemes to the Perron ℓ^p -eigenvector and to the maximal corresponding ℓ^p -eigenvalue of the tensor. Then, in the second part, motivated by the slow rate of convergence that the proposed methods achieve for certain real-world tensors when $p \approx d$, the number of modes, we introduce an extrapolation framework based on the simplified topological ε -algorithm to efficiently accelerate the shifted power sequences. Numerical results for synthetic and real world problems show the improvements gained by the introduction of the shifting parameter and the efficiency of the acceleration technique.

Key words. ℓ^p -eigenvalues, tensors, shifted higher-order power method, extrapolation methods, Shanks transformations, ε -algorithms

AMS subject classifications. 15A69, 15A18, 65B05, 65B10, 65B99, 65F15

1. Introduction. A tensor, or hypermatrix, is a multi-dimensional array: a set of numbers t_{i_1, \dots, i_d} indexed along d modes. When $d = 1$ the tensor is a vector, whereas for $d = 2$ it reduces to a matrix.

Tensor eigenvalue problems have gained considerable attention in recent years, and a number of contributions have addressed relevant issues both from the theoretical and the numerical points of view. The multi-dimensional nature of tensors naturally gives rise to a variety of eigenvalue problems. In fact, the classic eigenvalue and singular value problems for a matrix can be generalized to the tensor setting following different constructions which lead to different notions of eigenvalues and singular values for tensors, all of them reducing to the standard matrix case when the tensor is assumed to have 2 modes. The best known methods for computing eigenvalues and eigenvectors of real tensors are based on extensions of the power method. To our knowledge, the first occurrence of a power method for tensors, often called *higher-order* power method, is given in [32, 47], whereas shifted versions of this method, such as the SS-HOPM and LZI, were considered, e.g., in [34, 41]. When the considered tensor is real with nonnegative entries, a natural question arises whether and to what extent the Perron-Frobenius theorem for matrices can be transferred to the multi-dimensional setting. The answer turns out to be non-trivial and many authors have worked in this direction in recent years; see, for instance, [14, 24, 25, 41, 42]. A summary of these results and a unifying Perron–Frobenius theorem for tensors is presented in [28].

In this work we focus on the ℓ^p -eigenvector problem for square real tensors, being p any real number larger than 1. In Section 2 we review the definition and relevant properties of such eigenvalue problem. Then, in Section 3, we introduce two new shifted power methods for this tensor eigenproblem that include as special cases the SS-HOPM and the LZI algorithms mentioned above. In the case of real nonnegative tensors with possibly reducible patterns, we prove that the proposed shifted power sequences converge to the unique Perron ℓ^p -eigenvector and to the corresponding positive ℓ^p -eigenvalue, showing that in this setting the methods inherit the desirable convergence guarantees of their matrix counterpart. In Section 4 we study

*Received September 25, 2019. Accepted December 2, 2019. Published online on January 29, 2020. Recommended by L. Reichel.

[†]Department of Mathematics “Tullio Levi-Civita”, University of Padua, 35121 Padua, Italy (cipolla@math.unipd.it, michela.redivozaglia@unipd.it).

[‡]GSSI—Gran Sasso Science Institute, 67100 L’Aquila, Italy (francesco.tudisco@gssi.it).

experimentally the numerical behavior of the newly introduced schemes. In Section 5 we discuss how the sequences produced by the new algorithms can be accelerated by means of extrapolation methods that require only few additional computations. Finally, in Section 6, we show via numerical experiments that the extrapolated sequences converge significantly faster than the original power sequences for a number of different test problems from real world applications.

2. Tensor ℓ^p -eigenvalues and eigenvectors. In this section we review the notion of ℓ^p -eigenvalues for a square real tensor and discuss some of their properties.

Let $\mathbf{T} = (t_{i_1, \dots, i_d}) \in \mathbb{R}^{n \times \dots \times n}$ be a real tensor with d modes, each of dimension n . Since each mode of the tensor has the same dimension, we say that the tensor is square, and we briefly write $\mathbf{T} \in \mathbb{R}^{[d, n]}$. Furthermore, we say that \mathbf{T} is symmetric if the entries t_{i_1, \dots, i_d} are invariant with respect to any permutation of the indices i_1, \dots, i_d , as introduced in [17]. To any square tensor $\mathbf{T} \in \mathbb{R}^{[d, n]}$ and any vector \mathbf{x} of size n , we can associate a new vector \mathbf{y} by “multiplying” \mathbf{T} times \mathbf{x} . This operation extends the matrix-vector product and is defined via a polynomial map which we denote with the same capital letter denoting the corresponding tensor, but in italic normal font, more precisely:

DEFINITION 2.1. Given a tensor $\mathbf{T} = (t_{i_1, \dots, i_d}) \in \mathbb{R}^{n \times \dots \times n}$, define the map

$$(2.1) \quad T : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \mathbf{x} \mapsto T(\mathbf{x})_{i_1} = \sum_{i_2, \dots, i_d} t_{i_1, i_2, \dots, i_d} x_{i_2} \cdots x_{i_d}$$

for $i_1 = 1, \dots, n$.

For the sake of completeness, let us point out that the vector $T(\mathbf{x})$ is sometimes denoted by $\mathbf{T}\mathbf{x}^{d-1}$; see, e.g., [34, 47]. This is because, when $d = 2$, that is, when \mathbf{T} is a $n \times n$ matrix, then $T(\mathbf{x})$ coincides with the matrix-vector product $\mathbf{T}\mathbf{x}$. However, in this work we prefer to distinguish \mathbf{T} and T explicitly. This choice is made for the sake of generality, having in mind a future extension of the analysis here presented to tensor singular values and the case of rectangular tensors.

REMARK 2.2. Clearly, other multiplicative maps can be associated to a square tensor $\mathbf{T} \in \mathbb{R}^{[d, n]}$. Indeed, for any $1 \leq k \leq d$ one can define a map $T_k : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by replacing the summation on the right-hand side of (2.1) with the sum over all the indices i_1, \dots, i_d except for i_k . This defines d different operations T_1, \dots, T_d where $T_1 = T$ but $T_k \neq T$ in general if $k \neq 1$. Precisely, one easily realizes that all the mappings T_k coincide if and only if the tensor \mathbf{T} is symmetric. Note, in particular, that for the matrix case $d = 2$, the two maps T_1 and T_2 coincide with the matrix-vector products $\mathbf{T}\mathbf{x}$ and $\mathbf{T}^T\mathbf{x}$, respectively.

As T maps \mathbb{R}^n into itself, we can associate a concept of eigenvalues and eigenvectors to \mathbf{T} via the mapping T .

DEFINITION 2.3. Let $1 < p < +\infty$. A real number $\lambda \in \mathbb{R}$ is said to be an ℓ^p -eigenvalue of \mathbf{T} , corresponding to the ℓ^p -eigenvector $\mathbf{x} \in \mathbb{R}^n$, if the following holds

$$(2.2) \quad T(\mathbf{x}) = \lambda \Phi_p(\mathbf{x}), \quad \|\mathbf{x}\|_p = 1,$$

where $\|\mathbf{x}\|_p$ denotes the usual ℓ^p -norm $\|\mathbf{x}\|_p = (|x_1|^p + \dots + |x_n|^p)^{1/p}$ and $\Phi_p : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the map entry-wise defined by $\Phi_p(\mathbf{x})_i = |x_i|^{p-2} x_i = \text{sign}(x_i) |x_i|^{p-1}$ for $i = 1, \dots, n$.

The notion of ℓ^p -eigenvalues was probably introduced by Lim in [40]. Note that, unlike the matrix case, if $p \neq d$, then the system of equations $T(\mathbf{x}) = \lambda \Phi_p(\mathbf{x})$ is not homogeneous and thus, in general, ℓ^p -eigenvectors are not defined up to scalar multiples. This is the reason why we require the normalization $\|\mathbf{x}\|_p = 1$.

The map Φ_p is also known as the duality map of ℓ^p with gauge function $\mu(\alpha) = \alpha^{p-1}$ (see, e.g., [35]) and enjoys several useful properties. We recall two of them below:

- If q is the Hölder conjugate of p , i.e. $p^{-1} + q^{-1} = 1$, then $\Phi_q = \Phi_p^{-1}$ is the inverse of Φ_p , that is, we have $\Phi_p(\Phi_q(\mathbf{x})) = \Phi_q(\Phi_p(\mathbf{x})) = \mathbf{x}$ for any \mathbf{x} .
- When $p = 2$ we have $\Phi_p(\mathbf{x}) = \Phi_q(\mathbf{x}) = \mathbf{x}$, i.e., Φ_p is the identity map.

We say that a solution $(\lambda, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^n$ of equation (2.2) is an ℓ^p -eigenpair of \mathbf{T} . A special name is usually reserved to the cases $p = 2$ and $p = d$, the former being known as Z -eigenpairs and the latter as H -eigenpairs. Actually, when d is odd, the usual definition of H -eigenvectors given in the literature slightly differs from (2.2). In fact, an H -eigenpair is usually defined as the solution of the system $T(\mathbf{x}) = \lambda \mathbf{x}^{d-1}$ [45], which coincides with (2.2) only if d is even or if \mathbf{x} is entry-wise nonnegative. In this work we will call H -eigenvector any vector \mathbf{x} satisfying (2.2) with $p = d$. Of course one could define ℓ^p -eigenvalues by replacing the duality map Φ_p with a standard entry-wise power of the vector. However, as we notice below, using Φ_p has the advantage that, just like for symmetric matrices, ℓ^p -eigenvectors of symmetric tensors \mathbf{T} defined as in (2.2) have a variational characterization as critical points of the Rayleigh quotient

$$(2.3) \quad \mathbf{x} \mapsto f_p(\mathbf{x}) = \frac{\mathbf{x}^T T(\mathbf{x})}{\|\mathbf{x}\|_p^d}, \quad \mathbf{x}^T T(\mathbf{x}) = \sum_{i_1, \dots, i_d} t_{i_1, i_2, \dots, i_d} x_{i_1} \cdots x_{i_d}.$$

PROPOSITION 2.4. *Let \mathbf{T} be a symmetric square tensor, and let $p > 1$. Then $\mathbf{x} \in \mathbb{R}^n$ is an ℓ^p -eigenvector of \mathbf{T} if and only if \mathbf{x} is a critical point of f_p and the associated ℓ^p -eigenvalue coincides with $\mathbf{x}^T T(\mathbf{x}) / \|\mathbf{x}\|_p^d$.*

Proof. As both p and d are larger than one, the Rayleigh quotient f_p is differentiable and thus \mathbf{x} is a critical point of f_p if and only if $\nabla f_p(\mathbf{x}) = \mathbf{0}$, where ∇ denotes the gradient. By the symmetry of \mathbf{T} , the gradient of $\mathbf{x}^T T(\mathbf{x})$ is given by

$$\nabla \{\mathbf{x}^T T(\mathbf{x})\} = T_1(\mathbf{x}) + T_2(\mathbf{x}) + \cdots + T_d(\mathbf{x}) = dT(\mathbf{x}),$$

where $T_k : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are the maps considered in Remark 2.2. Similarly, we compute the gradient of the p -norm: $\nabla \|\mathbf{x}\|_p = \Phi_p(\mathbf{x}) / \|\mathbf{x}\|_p^{p-1}$.

Therefore, by the chain rule, we get

$$\nabla f_p(\mathbf{x}) = \frac{\|\mathbf{x}\|_p^d \nabla \{\mathbf{x}^T T(\mathbf{x})\} - \mathbf{x}^T T(\mathbf{x}) \nabla \|\mathbf{x}\|_p^d}{\|\mathbf{x}\|_p^{2d}} = \frac{d}{\|\mathbf{x}\|_p^d} \left\{ T(\mathbf{x}) - \left(\frac{\mathbf{x}^T T(\mathbf{x})}{\|\mathbf{x}\|_p^d} \right) \Phi_p(\mathbf{x}) \right\},$$

which concludes the proof. \square

REMARK 2.5. Note that with $\mathbf{y} = \mathbf{x} / \|\mathbf{x}\|_p$, we get $f_p(\mathbf{x}) = \mathbf{y}^T T(\mathbf{y})$. Hence, critical points of f_p coincide with critical points of $\mathbf{x}^T T(\mathbf{x})$, constrained by $\|\mathbf{x}\|_p = 1$. With this constrained optimization formulation one can obtain an equally simple proof of Proposition 2.4 as a direct consequence of the Karush-Kuhn-Tucker conditions. This is the approach used for example in [40] and [34, Theorem 3.2].

The notion of ℓ^p -eigenvectors and eigenvalues for nonnegative tensors arises in many contexts and applications. The cases $p = 2$ and $p = d$ are the natural generalization of eigenvalues and eigenvectors for matrices, as in the case $d = 2$ these two notions coincide. The case $p = d$ has been used, for example, to characterize the positive definiteness of homogeneous polynomial forms in [46] or to compute the importance of nodes in a network [2]. The case $p = 2$ is strictly related with the best rank-1 approximation of a tensor with respect to the Frobenius norm [20, 23, 33]. It arises in certain constraint satisfaction problems [19], as well as in problems involving higher-order statistics [50], signal processing [16, 44], quantum geometry [31], and data analysis [15, 29, 43, 51].

If \mathbf{T} is symmetric, then it has a finite set of Z -eigenvalues, one of which must be real if d is odd [12]. However, computing a prescribed Z -eigenpair is in general NP-hard [30]. When

\mathbf{T} is entry-wise nonnegative, instead, conditions can be given on the nonzero pattern of \mathbf{T} to ensure that tensor versions of the power method converge globally to the largest Z -eigenvalue [22, 26, 38]. These types of results belong to the developing Perron–Frobenius theory for nonnegative tensors, which naturally involves ℓ^p -eigenvalues and eigenvectors, with p not necessarily equal to 2 nor to d [14, 25, 27]. We review the relevant results below.

We say that $\mathbf{T} = (t_{i_1, \dots, i_d})$ is nonnegative when $t_{i_1, \dots, i_d} \geq 0$ for all $i_j = 1, \dots, n$ and $j = 1, \dots, d$. We briefly write $\mathbf{T} \geq 0$ and, likewise, $\mathbf{T} > 0$ when all the entries of \mathbf{T} are strictly positive. Several notions of spectral radius can be then employed to generalize this concept from the matrix to the higher-order case; see, e.g., [27]. Here we adopt the following

$$r_p(\mathbf{T}) = \sup\{|\lambda| : \lambda \text{ is an } \ell^p\text{-eigenvalue of } \mathbf{T}\}.$$

Note that, when \mathbf{T} is symmetric, using Proposition 2.4 and Remark 2.5, we have

$$r_p(\mathbf{T}) := \max_{\|\mathbf{x}\|_p=1} |\mathbf{x}^T T(\mathbf{x})|.$$

Unlike the matrix case, the Perron-Frobenius theorem for tensors relies on the choice of p and the number of modes d . To the best of our knowledge, if the tensor is not entry-wise positive or stochastic (settings that are studied for example in [26] and [22], respectively) $p \geq d$ is the wider range of values p for which existence, uniqueness, and maximality of a positive ℓ^p -eigenpair for $\mathbf{T} \geq 0$ has been proved; cf. [27], for instance. Within this range, a distinction must be made: While the case $p = d$ requires assumptions on the irreducibility of \mathbf{T} (e.g., strong and weak irreducibility or primitivity, see Definition 3.7 and [13, 24]), it was observed in [27] that $p > d$ is associated with a Lipschitz contractive map, and the Perron-Frobenius theorem holds without any special requirement on the non-zero pattern of \mathbf{T} . In this work we shall mostly focus on this second case, thus we recall below the corresponding Perron-Frobenius theorem. We refer to [27] for more details and for a thorough bibliography review on the subject.

THEOREM 2.6 ([27]). *Let $\mathbf{T} \in \mathbb{R}^{[d,n]}$ be such that $\mathbf{T} \geq 0$ and such that $T(\mathbf{1})$ is entry-wise positive, where $\mathbf{1}$ is the vector of all ones. If $p > d$, then $r_p(\mathbf{T}) > 0$, and there exists a unique entry-wise positive $\mathbf{u} \in \mathbb{R}^n$ such that $\|\mathbf{u}\|_p = 1$ and $T(\mathbf{u}) = r_p(\mathbf{T})\Phi_p(\mathbf{u})$.*

3. Shifted power method for ℓ^p -eigenvalues. The power method is arguably the best known method to address the computation of the maximal ℓ^p -eigenpair of \mathbf{T} . Shifted variants of this method have been proposed for Z -eigenpairs and for H -eigenpairs in [33] and [41], respectively. Here we propose two novel possible extensions of those techniques to the case of a general ℓ^p -eigenvalue problem for nonnegative tensors. The pseudocode of these new methods is shown in Algorithms 1 and 2. Note that the symbol \circ at line 4 of Algorithm 1 and at line 4 of Algorithm 2 denotes the entry-wise product and is required in order to ensure that, when $\mathbf{T} \geq 0$, both $T(\mathbf{x}_k)$ and the shifting vectors \mathbf{x}_k and $\Phi_p(\mathbf{x}_k)$ have the same zero pattern. In particular, when \mathbf{T} is irreducible, $\mathbf{x}_0 > 0$ implies that $T(\mathbf{x}_k) > 0$ for all k , thus $\text{sign}(\mathbf{z}) = \mathbf{1}$ and we retrieve the shifted methods for both Z and H eigenpairs as special cases of the new schemes.

The proposed shifted methods share several interesting convergence properties with the original unshifted power method. In particular, for nonnegative tensors and under the assumption $p > d$, we are guaranteed that if the initial \mathbf{x}_0 is chosen entry-wise positive, then for large enough k the whole sequence of eigenvector approximations \mathbf{x}_k will stay within a certain cone $C_+(\mathbf{T})$ and it will converge to the unique eigenvector of the tensor \mathbf{T} in that cone; at the same time, the sequence of eigenvalue approximations λ_k will converge to the corresponding eigenvalue and, if \mathbf{T} is symmetric, we are guaranteed that such eigenvalue corresponds to the ℓ^p -spectral radius of \mathbf{T} . We state this convergence result in our main Theorem 3.3 below.

Algorithm 1: Shifted Power Method 1.

Input: $\mathbf{x}_0 > 0, \sigma \geq 0, p > 1$, tolerance $\varepsilon > 0$

- 1 $q := p/(p-1)$ (*Conjugate exponent*)
- 2 **For** $k = 0, 1, 2, 3, \dots$ **repeat**
- 3 $\mathbf{z} = T(\mathbf{x}_k)$
- 4 $\mathbf{y} = \Phi_q(\mathbf{z}) + \sigma \text{sign}(\mathbf{z}) \circ \mathbf{x}_k$
- 5 $\mathbf{x}_{k+1} = \mathbf{y} / \|\mathbf{y}\|_p$
- 6 $\lambda_{k+1} = f_p(\mathbf{x}_{k+1})$
- 7 **until** $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_p < \varepsilon$

Algorithm 2: Shifted Power Method 2.

Input: $\mathbf{x}_0 > 0, \sigma \geq 0, p > 1$, tolerance $\varepsilon > 0$

- 1 $q := p/(p-1)$ (*Conjugate exponent*)
- 2 **For** $k = 0, 1, 2, 3, \dots$ **repeat**
- 3 $\mathbf{z} = T(\mathbf{x}_k)$
- 4 $\mathbf{y} = \Phi_q(\mathbf{z} + \sigma \text{sign}(\mathbf{z}) \circ \Phi_p(\mathbf{x}_k))$
- 5 $\mathbf{x}_{k+1} = \mathbf{y} / \|\mathbf{y}\|_p$
- 6 $\lambda_{k+1} = f_p(\mathbf{x}_{k+1})$
- 7 **until** $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_p < \varepsilon$

3.1. Convergence analysis. Let \mathbb{R}_+^n denote the nonnegative orthant in \mathbb{R}^n , i.e., the cone of vectors with nonnegative entries, and let \mathbb{R}_{++}^n be its interior, i.e., the open cone of vectors with strictly positive entries. Consider the cone of nonnegative vectors whose zero pattern is preserved by \mathbf{T} :

DEFINITION 3.1. For a nonzero tensor $\mathbf{T} \geq 0$ define $C_+(\mathbf{T})$ as the cone

$$C_+(\mathbf{T}) = \bigcap_{k \geq k_0} C_+^k \quad \text{with} \quad C_+^k = \left\{ \mathbf{x} \geq 0 : c_1 \mathbf{x} \leq T^k(\mathbf{1}) \leq c_2 \mathbf{x}, \text{ for some } c_1, c_2 > 0 \right\},$$

where T^k denote k compositions of the map T and $k_0 = k_0(\mathbf{T})$ is the smallest integer $k_0 \geq 1$ such that $C_+(\mathbf{T})$ is not empty.

Note that, for each k , the set C_+^k is the cone of nonnegative vectors having the same zero pattern as $T^k(\mathbf{1})$. Since $T^{k+1}(\mathbf{1}) = T(T^k(\mathbf{1}))$ and $\mathbf{1} \geq T(\mathbf{1})$ entry-wise, one easily realizes that $T^k(\mathbf{1})_i = 0$ implies $T^{k+1}(\mathbf{1})_i = 0$. Thus, there exists a finite integer k_0 such that $\bigcap_{k \geq k_0} C_+^k$ is nonempty, i.e., $C_+(\mathbf{T})$ is well defined. Although $C_+(\mathbf{T})$ may contain only the zero vector when \mathbf{T} is too sparse, this situation rarely occurs in practice. Indeed, for example, when $T(\mathbf{1}) > 0$, then $C_+(\mathbf{T}) = \mathbb{R}_{++}^n$ is the whole cone of positive vectors.

From its definition we notice that $C_+(\mathbf{T})$ is the cone of nonnegative vectors with the least number of zero entries that is preserved by T and that $C_+(\mathbf{T})$ is also preserved by the iterates of the shifted power methods. Precisely, if \mathbf{x}_{k+1} is computed from \mathbf{x}_k using the updating formulas of either Algorithm 1 or Algorithm 2 and $\mathbf{x}_k \in C_+(\mathbf{T})$, then also $\mathbf{x}_{k+1} \in C_+(\mathbf{T})$. Moreover, we will show in the theorem below that \mathbf{T} has a unique ℓ^p -eigenvector in $C_+(\mathbf{T})$ and that the two algorithms converge to it.

At the same time, we do not need to compute the cone $C_+(\mathbf{T})$ beforehand, as we are guaranteed to converge within $C_+(\mathbf{T})$ if we start with any positive vector $\mathbf{x}_0 > 0$. More precisely, if $\mathbf{x}_0 > 0$ and (\mathbf{x}_k) is the sequence generated by either Algorithm 1 or Algorithm 2, then all the vectors \mathbf{x}_k belong to $C_+(\mathbf{T})$ when k is large enough. Indeed, if we start with \mathbf{x}_0

entry-wise positive, then \mathbf{x}_1 has the same zero pattern as $T(\mathbf{1})$, \mathbf{x}_2 has the same zero pattern as $T^2(\mathbf{1})$, and so forth. Thus, $\mathbf{x}_k \in C_+(\mathbf{T})$ for all $k \geq k_0$.

In what follows, absolute values are intended entry-wise, that is, for $\mathbf{x} \in \mathbb{R}^n$, $|\mathbf{x}|$ denotes the vector with entries $|x_i|$, for $i = 1, \dots, n$.

LEMMA 3.2. *Let $\mathbf{T} \in \mathbb{R}^{[d,n]}$ be entry-wise nonnegative, and let f_p be defined as in (2.3). Then the maximum of f_p is attained at a nonnegative vector.*

Proof. Let $\mathbf{x} \in \mathbb{R}^n$. Since \mathbf{T} has nonnegative entries, we have $T(\mathbf{x}) \leq T(|\mathbf{x}|)$. Thus, if $\mathbf{x} \neq 0$, we have $|f_p(\mathbf{x})| = \frac{|\mathbf{x}^T T(\mathbf{x})|}{\|\mathbf{x}\|_p^d} \leq \frac{|\mathbf{x}^T T(|\mathbf{x}|)|}{\|\mathbf{x}\|_p^d} = f_p(|\mathbf{x}|)$. Hence, if $\mathbf{y} \in \mathbb{R}^n \setminus \{0\}$ is a global maximizer of f_p , then also $|\mathbf{y}|$ is a global maximizer. \square

The following main result holds:

THEOREM 3.3. *Let $\mathbf{T} \geq 0$, $k_0 = k_0(\mathbf{T})$ be as in Definition 3.1, and let $p > d$. Then there exists a unique solution to (2.2) in $C_+(\mathbf{T})$ that is a unique ℓ^p -eigenvector $\mathbf{u} \in C_+(\mathbf{T})$. We will define this as the Perron ℓ^p -eigenvector. In particular, for any $\sigma \geq 0$ and any $\mathbf{x}_0 > 0$, the sequences (λ_k) and (\mathbf{x}_k) defined by either Algorithm 1 or Algorithm 2 are such that:*

1. $\mathbf{x}_k \in C_+(\mathbf{T})$ for all $k \geq k_0$;
2. (\mathbf{x}_k) converges to the Perron ℓ^p -eigenvector \mathbf{u} of \mathbf{T} in $C_+(\mathbf{T})$ with $\|\mathbf{u}\|_p = 1$;
3. (λ_k) converges to the ℓ^p -eigenvalue $\lambda > 0$ of \mathbf{T} corresponding to \mathbf{u} .

Proof. Point 1 holds by definition of $C_+(\mathbf{T})$. So, for any $k \geq k_0$ all the points of the sequence (\mathbf{x}_k) are in $C_+(\mathbf{T})$. We show below that the iterators are contractions on $C_+(\mathbf{T})$.

Given two vectors $\mathbf{x}, \mathbf{y} \in C_+(\mathbf{T})$, consider the following quantities

$$M(\mathbf{x}/\mathbf{y}) = \inf\{\mu > 0 : \mathbf{x} \leq \mu \mathbf{y}\} \quad m(\mathbf{x}/\mathbf{y}) = \sup\{\mu > 0 : \mathbf{x} \geq \mu \mathbf{y}\}.$$

Since $C_+(\mathbf{T})$ is a simplicial cone (see, e.g., [37]), for $\mathbf{x}, \mathbf{y} \in C_+(\mathbf{T})$ it holds that $M(\mathbf{x}/\mathbf{y}) = \max_{y_i \neq 0} x_i/y_i$ and $m(\mathbf{x}/\mathbf{y}) = \min_{y_i \neq 0} x_i/y_i$. For any $\mathbf{x}, \mathbf{y} \in C_+(\mathbf{T})$, by definition we have $m(\mathbf{x}/\mathbf{y})\mathbf{y} \leq \mathbf{x} \leq M(\mathbf{x}/\mathbf{y})\mathbf{y}$, thus

$$m(\mathbf{x}/\mathbf{y})^{d-1}T(\mathbf{y}) \leq T(\mathbf{x}) \leq M(\mathbf{x}/\mathbf{y})^{d-1}T(\mathbf{y}).$$

For $\sigma \geq 0$ and $p > 1$, consider the following two mappings:

$$H_\sigma(\mathbf{x}) = \Phi_p^{-1}(T(\mathbf{x})) + \sigma \mathbf{x} \quad \text{and} \quad K_\sigma(\mathbf{x}) = \Phi_p^{-1}(T(\mathbf{x}) + \sigma \Phi_p(\mathbf{x})).$$

Note that both H_σ and K_σ preserve $C_+(\mathbf{T})$, that is, for any $\mathbf{x} \in C_+(\mathbf{T})$ we have $H_\sigma(\mathbf{x}) \in C_+(\mathbf{T})$ and $K_\sigma(\mathbf{x}) \in C_+(\mathbf{T})$. Also note that when $\mathbf{x} \in C_+(\mathbf{T})$ then $\mathbf{z} = T(\mathbf{x}) \in C_+(\mathbf{T})$. Thus, $\text{sign}(\mathbf{z}) \circ \mathbf{x} = \mathbf{x}$ and $\text{sign}(\mathbf{z}) \circ \Phi_p(\mathbf{x}) = \Phi_p(\mathbf{x})$. Hence, as $q = p/(p-1)$, we have $\Phi_q = \Phi_p^{-1}$ and $H_\sigma(\mathbf{x}), K_\sigma(\mathbf{x})$ are the iteration maps in Algorithms 1 and 2. Observe, moreover, that for any two distinct points \mathbf{x} and \mathbf{y} on the ℓ^p -sphere $\mathcal{S}_p = \{\mathbf{x} : \|\mathbf{x}\|_p = 1\}$, we have $M(\mathbf{x}/\mathbf{y}) > 1$ and $m(\mathbf{x}/\mathbf{y}) < 1$. In fact, $M(\mathbf{x}/\mathbf{y}) \leq 1$ implies $x_i \leq y_i$ for any i and $x_i < y_i$ for at least one i (as $\mathbf{x} \neq \mathbf{y}$), which contradicts the fact that \mathbf{x} and \mathbf{y} have the same ℓ^p -norm. An analogous contradiction follows by assuming $m(\mathbf{x}/\mathbf{y}) \geq 1$.

Therefore, as $p > d$, for any two $\mathbf{x}, \mathbf{y} \in C_+(\mathbf{T}) \cap \mathcal{S}_p$, we have $M(\mathbf{x}/\mathbf{y})^{\frac{d-1}{p-1}} < M(\mathbf{x}/\mathbf{y})$, $m(\mathbf{x}/\mathbf{y})^{\frac{d-1}{p-1}} > m(\mathbf{x}/\mathbf{y})$, and

$$(3.1) \quad H_\sigma(\mathbf{x}) \leq M(\mathbf{x}/\mathbf{y})^{\frac{d-1}{p-1}} \Phi_p^{-1}(T(\mathbf{y})) + \sigma M(\mathbf{x}/\mathbf{y})\mathbf{y} < M(\mathbf{x}/\mathbf{y})H_\sigma(\mathbf{y}),$$

$$(3.2) \quad H_\sigma(\mathbf{x}) \geq m(\mathbf{x}/\mathbf{y})^{\frac{d-1}{p-1}} \Phi_p^{-1}(T(\mathbf{y})) + \sigma m(\mathbf{x}/\mathbf{y})\mathbf{y} > m(\mathbf{x}/\mathbf{y})H_\sigma(\mathbf{y}),$$

$$(3.3) \quad K_\sigma(\mathbf{x}) \leq \Phi_p^{-1}(M(\mathbf{x}/\mathbf{y})^{d-1}T(\mathbf{y}) + \sigma M(\mathbf{x}/\mathbf{y})^{p-1}\Phi_p(\mathbf{y})) < M(\mathbf{x}/\mathbf{y})K_\sigma(\mathbf{y}),$$

$$(3.4) \quad K_\sigma(\mathbf{x}) \geq \Phi_p^{-1}(m(\mathbf{x}/\mathbf{y})^{d-1}T(\mathbf{y}) + \sigma m(\mathbf{x}/\mathbf{y})^{p-1}\Phi_p(\mathbf{y})) > m(\mathbf{x}/\mathbf{y})K_\sigma(\mathbf{y}).$$

Let us now consider the normalized maps

$$\tilde{H}_\sigma(\mathbf{x}) = H_\sigma(\mathbf{x})/\|H_\sigma(\mathbf{x})\|_p \quad \text{and} \quad \tilde{K}_\sigma(\mathbf{x}) = K_\sigma(\mathbf{x})/\|K_\sigma(\mathbf{x})\|_p,$$

and the Hilbert projective distance between \mathbf{x} and \mathbf{y} in $C_+(\mathbf{T})$, defined by

$$(3.5) \quad d_H(\mathbf{x}, \mathbf{y}) = \log \left(\frac{M(\mathbf{x}/\mathbf{y})}{m(\mathbf{x}/\mathbf{y})} \right).$$

Note that both \tilde{H}_σ and \tilde{K}_σ preserve the set $C_+(\mathbf{T}) \cap \mathcal{S}_p$. Moreover, as d_H is scale invariant, combining inequalities (3.1)–(3.4) we see that both \tilde{H}_σ and \tilde{K}_σ are contractions on $C_+(\mathbf{T}) \cap \mathcal{S}_p$ with respect to d_H . Indeed, we get

$$d_H(\tilde{H}_\sigma(\mathbf{x}), \tilde{H}_\sigma(\mathbf{y})) = \log \left(\frac{M(H_\sigma(\mathbf{x})/H_\sigma(\mathbf{y}))}{m(H_\sigma(\mathbf{x})/H_\sigma(\mathbf{y}))} \right) < \log \left(\frac{M(\mathbf{x}/\mathbf{y})}{m(\mathbf{x}/\mathbf{y})} \right) = d_H(\mathbf{x}, \mathbf{y})$$

and similarly for \tilde{K}_σ .

As $C_+(\mathbf{T}) \cap \mathcal{S}_p$ is complete (see, e.g., [26]), the Banach fixed point theorem (see [21]) implies that there exists a unique $\mathbf{u} \in C_+(\mathbf{T}) \cap \mathcal{S}_p$ such that, for any initial choice $\mathbf{x}_0 > 0$, we have $\tilde{H}_\sigma(\mathbf{x}_k) \rightarrow \mathbf{u} = \tilde{H}_\sigma(\mathbf{u})$ as $k \rightarrow \infty$. Similarly, $\tilde{K}_\sigma(\mathbf{x}_k) \rightarrow \mathbf{v} = \tilde{K}_\sigma(\mathbf{v})$ for a unique $\mathbf{v} \in C_+(\mathbf{T}) \cap \mathcal{S}_p$. Thus $H_\sigma(\mathbf{u}) = \lambda_1 \mathbf{u}$ and $K_\sigma(\mathbf{v}) = \lambda_2 \mathbf{v}$ with $\lambda_1 = \|H_\sigma(\mathbf{u})\|_p$ and $\lambda_2 = \|K_\sigma(\mathbf{v})\|_p$. We now show that \mathbf{u} and \mathbf{v} coincide. Indeed, from $K_\sigma(\mathbf{v}) = \lambda_2 \mathbf{v}$, we have $\Phi_p^{-1}(T(\mathbf{v})) = \Phi_p^{-1}(\Phi_p(\lambda_2) - \sigma)\mathbf{v} =: \alpha \mathbf{v}$. Therefore $H_\sigma(\mathbf{v}) = (\alpha + \sigma)\mathbf{v}$, that is, \mathbf{v} is the fixed point of \tilde{H}_σ in $C_+(\mathbf{T}) \cap \mathcal{S}_p$, i.e., $\mathbf{v} = \mathbf{u}$.

Finally, note that \mathbf{u} is an eigenvector of \mathbf{T} with eigenvalue $\lambda = \Phi_p(\lambda_2) - \sigma = \Phi_p(\lambda_1 - \sigma)$ and such λ is positive. In fact, as both $\Phi_p(\mathbf{u})$ and $T(\mathbf{u})$ are nonnegative vectors in $C_+(\mathbf{T})$, the identity $T(\mathbf{u}) = \lambda \Phi_p(\mathbf{u})$ implies that $\lambda > 0$. \square

COROLLARY 3.4. *If $\mathbf{T} \geq 0$ is symmetric and $p > d$, then the sequence (λ_k) generated by either Algorithm 1 or Algorithm 2 converges to $r_p(\mathbf{T})$.*

Proof. Let \mathbf{w} be any ℓ^p -eigenvector of \mathbf{T} with $\|\mathbf{w}\|_p = 1$ and eigenvalue μ . We have

$$|\mu| = |\mathbf{w}^T T(\mathbf{w})| \leq |\mathbf{w}|^T T(|\mathbf{w}|) \leq \max_{\|\mathbf{x}\|_p=1} f_p(\mathbf{x}) =: \tilde{r}.$$

Let \mathbf{x}^* be a maximizer of f_p , and let $\mathbf{u}^* = \mathbf{x}^*/\|\mathbf{x}^*\|_p$. As $f_p(\mathbf{x}^*) = f_p(\mathbf{u}^*)$ we deduce that \mathbf{u}^* is a maximizer, too, and by Lemma 3.2 we can assume $\mathbf{u}^* \geq 0$. By Proposition 2.4 we have that \mathbf{u}^* is an ℓ^p -eigenvector of \mathbf{T} , that is, $T(\mathbf{u}^*) = \tilde{r} \Phi_p(\mathbf{u}^*)$, with $\tilde{r} = r_p(\mathbf{T}) = f_p(\mathbf{u}^*)$. In particular, $r_p(\mathbf{T})^{\frac{1}{p-1}} \mathbf{u}^* = \Phi_p^{-1}(T(\mathbf{u}^*))$ implies that for all $k \geq 1$ there exists $\lambda_k > 0$ such that $\lambda_k \mathbf{u}^* = (\Phi_p^{-1} T)^k(\mathbf{u}^*)$. Thus, if $T^k(\mathbf{u}^*)_i = 0$ for some integer k , then also $(\mathbf{u}^*)_i = 0$. In other words, \mathbf{u}^* is zero at least in all the positions where the vectors of $C_+(\mathbf{T})$ are zero.

Now, let \mathbf{u} be the unique eigenvector of \mathbf{T} in $C_+(\mathbf{T})$ with eigenvalue $\lambda > 0$ and the limit of the sequences generated by either Algorithm 1 or Algorithm 2 as in Theorem 3.3. Suppose that $\lambda < r_p(\mathbf{T})$, and let $\alpha, \beta \in (0, 1)$ be such that $\alpha^p + \beta^p = 1$. We have $\mathbf{u} \neq \mathbf{u}^*$ and if $\mathbf{y} = \alpha \mathbf{u}^* + \beta \mathbf{u}$, then $\mathbf{y} \in C_+(\mathbf{T})$ and $\|\mathbf{y}\|_p = 1$. Therefore,

$$f_p(\mathbf{y}) = \mathbf{y}^T T(\mathbf{y}) \geq (\alpha \mathbf{u} + \beta \mathbf{u}^*)^T \left(\alpha^{d-1} T(\mathbf{u}) + \beta^{d-1} T(\mathbf{u}^*) \right) \geq \alpha^d f_p(\mathbf{u}) + \beta^d f_p(\mathbf{u}^*).$$

As $\alpha = (1 - \beta^p)^{1/p}$ and $p > d$, we have $\alpha^d = (1 - \beta^p)^{d/p} > (1 - \beta^p) > (1 - \beta^d)$, and thus, we can choose α and β so that

$$f_p(\mathbf{y}) \geq \alpha^d f_p(\mathbf{u}) + \beta^d f_p(\mathbf{u}^*) = \alpha^d \lambda + \beta^d r_p(\mathbf{T}) > \lambda.$$

On the other hand, the uniqueness of \mathbf{u} implies that $\lambda = \max_{\mathbf{x} \in C_+(\mathbf{T})} f_p(\mathbf{x})$, which yields a contradiction. Thus $\lambda = r_p(\mathbf{T})$, concluding the proof. \square

Before moving forward, let us briefly comment on Theorems 2.6 and 3.3. By considering the general ℓ^p -eigenvalue problem, Theorem 3.3 completes the work of [41] and shows that both Algorithms 1 and 2 converge for any square tensor \mathbf{T} and any choice of $p > d$. Moreover, by using the cone $C_+(\mathbf{T})$ rather than the whole positive orthant in \mathbb{R}^n , we partially extend the Perron-Frobenius theorem of [28] to the case of tensors with some zero unfolding. In fact, Theorem 3.3 shows that Theorem 2.6 holds unchanged without the assumption $T(\mathbf{1}) > 0$ and replacing the positive orthant with $C_+(\mathbf{T})$. However, we point out that when $C_+(\mathbf{T}) = \mathbb{R}_{++}^n$ and $\sigma = 0$, Theorem 2.6 actually tells us more than the above Theorem 3.3. First of all the proof of Theorem 2.6 shows that the power sequence in that case is associated with a Lipschitz contractive map, which is a stronger property than the contractivity shown above in Theorem 3.3. This allows us to show that the sequence for $\sigma = 0$ converges linearly, with an explicit convergence ratio. Moreover, Theorem 2.6 ensures convergence of the power sequence to $r_p(\mathbf{T})$, without assuming any symmetry on \mathbf{T} . We believe that the discussed results of Theorem 2.6 can be transferred to the case of shifted power sequences and to tensors with some zero unfolding, but this analysis goes beyond the scope of this paper and is postponed to future work.

3.2. Dependence on the parameter p . In this section we show that the Perron ℓ^p -eigenvector depends continuously on the parameter p . To this end, let us first define the following maps:

$$H(\mathbf{x}, p) := \Phi_p^{-1}(T(\mathbf{x})) \quad \text{and} \quad \tilde{H}(\mathbf{x}, p) := \frac{H(\mathbf{x}, p)}{\|H(\mathbf{x}, p)\|_p}.$$

Note that $H(\mathbf{x}, p) = H_0(\mathbf{x}) = K_0(\mathbf{x})$ (respectively, $\tilde{H}(\mathbf{x}, p) = \tilde{H}_0(\mathbf{x}) = \tilde{K}_0(\mathbf{x})$) being H_0, K_0 (respectively, \tilde{H}_0, \tilde{K}_0) the maps introduced in the proof of Theorem 3.3 for the choice $\sigma = 0$. With this notation and due to Theorem 3.3, we have that for any $p > d$, there exists a unique \mathbf{x}^* in $C_+(\mathbf{T})$ such that $\tilde{H}(\mathbf{x}^*, p) = \mathbf{x}^*$ and such that \mathbf{x}^* is the unique Perron ℓ^p -eigenvector of \mathbf{T} . Thus, with a slight abuse of notation, consider the following mapping,

$$\mathbf{x}^* : (d, +\infty) \rightarrow C_+(\mathbf{T}), \quad p \mapsto \mathbf{x}^*(p) \quad \text{such that} \quad \tilde{H}(\mathbf{x}^*(p), p) = \mathbf{x}^*(p),$$

which to a given $p \in (d, +\infty)$ assigns the unique Perron ℓ^p -eigenvector of \mathbf{T} .

In order to prove Theorem 3.6, which states the continuous dependence on the parameter p , we need the following additional technical lemma.

LEMMA 3.5. *Let $\mathbf{T} \geq 0$ and $p > d \geq 2$. For $\mathbf{x} \in C_+(\mathbf{T})$ it holds*

$$\Phi_{p+\varepsilon}^{-1}(\mathbf{x}) = \Phi_p^{-1}(\mathbf{x}) + \Psi(\mathbf{x})\varepsilon + o(\varepsilon),$$

where $\Psi(\mathbf{x})_i = -\frac{1}{(p-1)^2} x_i^{\frac{1}{p-1}} \ln(x_i)$ if $x_i > 0$ and $\Psi(\mathbf{x})_i = 0$, otherwise.

Proof. Let $q_\varepsilon = (p + \varepsilon)/(p - 1 + \varepsilon) = 1 + 1/(p - 1 + \varepsilon)$ be the conjugate exponent of $p + \varepsilon$. For ε sufficiently small it holds that

$$q_\varepsilon - 1 = \frac{1}{p - 1 + \varepsilon} = \frac{1}{p - 1} - \frac{1}{(p - 1)^2} \varepsilon + o(\varepsilon),$$

and thus, for a sufficiently small ε we can write

$$(3.6) \quad \Phi_{p+\varepsilon}^{-1}(\mathbf{x}) = \Phi_{q_\varepsilon}(\mathbf{x}) = \mathbf{x}^{\frac{1}{p-1}} \mathbf{x}^{-\frac{1}{(p-1)^2} \varepsilon + o(\varepsilon)}.$$

Moreover, since for $x > 0$ it holds that $x^\varepsilon = 1 + \varepsilon \ln(x) + o(\varepsilon)$, we have

$$(3.7) \quad x_i^{-\frac{1}{(p-1)^2} \varepsilon + o(\varepsilon)} = 1 - \frac{1}{(p-1)^2} \ln(x_i) \varepsilon + o(\varepsilon),$$

and hence the thesis follows by combining (3.7) and (3.6). \square

THEOREM 3.6. *If $\mathbf{T} \geq 0$ and $p > d \geq 2$, then for any small enough $\varepsilon > 0$ we have*

$$d_H(\mathbf{x}^*(p), \mathbf{x}^*(p + \varepsilon)) = O(\varepsilon),$$

where d_H is the Hilbert distance defined in (3.5).

Proof. Using the triangle inequality and since the Hilbert metric is scale invariant, we have

$$(3.8) \quad \begin{aligned} d_H(\mathbf{x}^*(p), \mathbf{x}^*(p + \varepsilon)) &= d_H(\tilde{H}(\mathbf{x}^*(p), p), \tilde{H}(\mathbf{x}^*(p + \varepsilon), p + \varepsilon)) \\ &\leq d_H(\tilde{H}(\mathbf{x}^*(p), p), \tilde{H}(\mathbf{x}^*(p + \varepsilon), p)) + d_H(\tilde{H}(\mathbf{x}^*(p + \varepsilon), p + \varepsilon), \tilde{H}(\mathbf{x}^*(p + \varepsilon), p)) \\ &= \underbrace{d_H(H(\mathbf{x}^*(p), p), H(\mathbf{x}^*(p + \varepsilon), p))}_{(a)} + \underbrace{d_H(H(\mathbf{x}^*(p + \varepsilon), p + \varepsilon), H(\mathbf{x}^*(p + \varepsilon), p))}_{(b)}. \end{aligned}$$

We now upper bound both terms in (a) and in (b). Concerning part (a), note that using (3.1) and (3.2) for $\sigma = 0$, for any two $\mathbf{x}, \mathbf{y} \in C_+(\mathbf{T})$ we have

$$\begin{aligned} d_H(H(\mathbf{x}), H(\mathbf{y})) &= \log \left(\frac{M(H(\mathbf{x})/H(\mathbf{y}))}{m(H(\mathbf{x})/H(\mathbf{y}))} \right) \\ &\leq \log \left(\left\{ \frac{M(\mathbf{x}/\mathbf{y})}{m(\mathbf{x}/\mathbf{y})} \right\}^{\frac{d-1}{p-1}} \right) = \left(\frac{d-1}{p-1} \right) d_H(\mathbf{x}, \mathbf{y}). \end{aligned}$$

Thus,

$$(3.9) \quad d_H(H(\mathbf{x}^*(p), p), H(\mathbf{x}^*(p + \varepsilon), p)) \leq \gamma d_H(\mathbf{x}^*(p), \mathbf{x}^*(p + \varepsilon))$$

with $\gamma = (d-1)/(p-1)$. As for part (b), notice that, for any $\mathbf{x} \in C_+(\mathbf{T})$ and ε sufficiently small, using Lemma 3.5, we have

$$\begin{aligned} H(\mathbf{x}, p + \varepsilon) &= \Phi_{p+\varepsilon}^{-1}(T(\mathbf{x})) = \Phi_p^{-1}(T(\mathbf{x})) + \varepsilon \Psi(T(\mathbf{x})) + o(\varepsilon) \\ &= H(\mathbf{x}, p) + \varepsilon \Psi(T(\mathbf{x})) + o(\varepsilon). \end{aligned}$$

Hence we have,

$$M(H(\mathbf{x}, p + \varepsilon)/H(\mathbf{x}, p)) = \max_{i: x_i \neq 0} \frac{H(\mathbf{x}, p)_i + \varepsilon \Psi(T(\mathbf{x}))_i + o(\varepsilon)}{H(\mathbf{x}, p)_i} \leq 1 + \varepsilon C_1 + o(\varepsilon)$$

and

$$m(H(\mathbf{x}, p + \varepsilon)/H(\mathbf{x}, p)) = \min_{i: x_i \neq 0} \frac{H(\mathbf{x}, p)_i + \varepsilon \Psi(T(\mathbf{x}))_i + o(\varepsilon)}{H(\mathbf{x}, p)_i} \geq 1 + \varepsilon C_2 + o(\varepsilon),$$

where

$$C_1 := \max_{i: x_i \neq 0} \frac{\Psi(T(\mathbf{x}))_i}{H(\mathbf{x}, p)_i} \quad \text{and} \quad C_2 := \min_{i: x_i \neq 0} \frac{\Psi(T(\mathbf{x}))_i}{H(\mathbf{x}, p)_i}.$$

Eventually, we obtain

$$d_H(H(\mathbf{x}, p + \varepsilon), H(\mathbf{x}, p)) \leq \log \left(\frac{1 + \varepsilon C_1 + o(\varepsilon)}{1 + \varepsilon C_2 + o(\varepsilon)} \right) = \varepsilon(C_1 - C_2) + o(\varepsilon).$$

Using the above inequality for $\mathbf{x} = \mathbf{x}^*(p + \varepsilon)$, together with (3.8) and (3.9) we obtain

$$d_H(\mathbf{x}^*(p), \mathbf{x}^*(p + \varepsilon)) \leq \varepsilon \frac{(C_1 - C_2 + 1)}{(1 - \gamma)},$$

which concludes the proof. \square

Theorem 3.6 shows that when $p > d$, the Perron ℓ^p -eigenvector of a nonnegative tensor depends continuously on the parameter p . This theorem can be useful for example in the context of H -eigenvectors because it allows us to argue that, in order to approximate an H -eigenvector of $\mathbf{T} \geq 0$, we can use an ℓ^p -eigenvector with $p \approx d$. Indeed, note that for nonnegative eigenvectors the duality map Φ_p coincides with taking the entry-wise powers of the vector, and thus, ℓ^d -eigenvectors coincide with H -eigenvectors. Note moreover that, if the contractivity property (3.9) holds also for the case $p = d$ (which does happen in practice, as discussed in [22, 26]), then Theorem 3.6 would work also for $p = d$. This is particularly interesting because this argument holds without any irreducibility requirement on \mathbf{T} . Instead, for the case of H -eigenvectors, the convergence of the power method requires the tensor to be weakly irreducible (see Definition 3.7 and [28, Theorem 3.3] or [54, Theorem 3.1]), which is an expensive property to verify, especially for large tensors. So, for example, if $T(\mathbf{1}) > 0$ but $\mathbf{T} \geq 0$ is not weakly irreducible, then we can use the positive Perron ℓ^p -eigenvector of \mathbf{T} with $p \approx d$ to compute an approximation of a positive H -eigenvector. To our knowledge, there exists no other method with this type of guarantee to approximate a positive H -eigenvector of a reducible tensor. We clarify this idea with an example in the remainder of this section.

3.2.1. Computing positive H -eigenvectors (case $p = d$). Given a nonnegative tensor $\mathbf{T} \in \mathbb{R}^{[d, n]}$, consider the graph $G(\mathbf{T}) = (V, E)$ defined as follows: $V = \{1, \dots, n\}$ and there is an edge from node u to node v , i.e., $uv \in E$, if

$$t_{u, j_2, \dots, j_{m-1}, v, j_{m+1}, \dots, j_d} > 0$$

for some $m = 2, 3, \dots, d - 1$.

DEFINITION 3.7 (Weakly irreducible tensor). *The tensor \mathbf{T} is said to be weakly irreducible or, in the more general notation of [28], $\{1, \dots, n\}$ -weakly irreducible if $G(\mathbf{T})$ is strongly connected.*

In [28], points (iii) of Theorems 3.1 and 3.3, it is shown that if a tensor is weakly irreducible then there exists a unique positive H -eigenvector and the power method converges to such H -eigenvector. However, in general, if the tensor is not weakly irreducible, then we have no guarantee of convergence for the power method to an H -eigenpair; see [28]. Instead, using our Theorem 3.3, convergence is still ensured when $p > d$, and we can use values of p slightly larger than d to efficiently compute good approximations to positive H -eigenvectors.

For example, consider the following $3 \times 3 \times 3$ nonnegative tensor

$$(3.10) \quad \mathbf{T}(:, :, 1) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{T}(:, :, 2) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{T}(:, :, 3) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

It is readily seen that this tensor is not weakly irreducible. Indeed, the graph of this tensor is

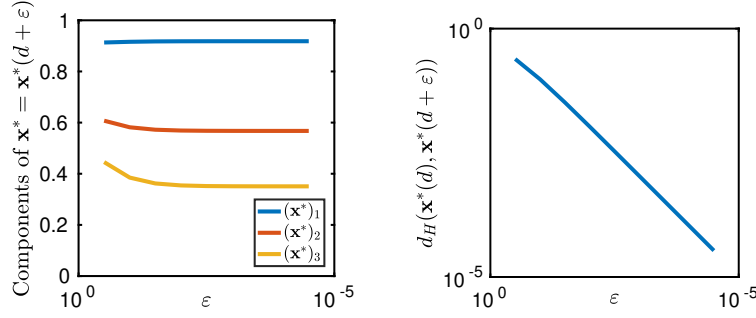
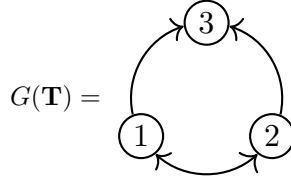


FIG. 3.1. Left: The three entries of the unique positive Perron ℓ^p -eigenvector $\mathbf{x}^*(d + \varepsilon)$, as ε decreases from 10^{-1} to 10^{-5} . Right: LogLog plot of the Hilbert distance between $\mathbf{x}^*(d + \varepsilon)$ and the H -eigenvector $\tilde{\mathbf{u}}_2$, as ε decreases. The plot confirms the behavior of Theorem 3.6.



which is not strongly connected. Thus, uniqueness of a nonnegative H -eigenvector and convergence of the power method are not guaranteed. Indeed, for this particular tensor, the eigenvector equation $T(\mathbf{x}) = \lambda \Phi_p(\mathbf{x})$ for $p = d$ can be expressed as the system of equations

$$\begin{cases} x_3 x_1 + x_2^2 + x_2 x_3 = \lambda x_1 |x_1|, \\ x_3 x_1 = \lambda x_2 |x_2|, \\ x_2^2 = \lambda x_3 |x_3|. \end{cases}$$

A few algebraic manipulations show that all the H -eigenpairs of \mathbf{T} are (up to normalization) of the form

$$\lambda = 1, \mathbf{u} = \begin{bmatrix} \mu |\mu| \\ \mu \\ 1 \end{bmatrix} \quad \text{or} \quad \lambda = -1, \mathbf{v} = \begin{bmatrix} \eta |\eta| \\ \eta \\ -1 \end{bmatrix},$$

where μ and η are real numbers such that

$$\mu(\mu^3 - 2\mu - 1)(-\mu^3 - 1) = 0, \quad \eta(\eta^3 - 1)(-\eta^3 + 2\eta - 1) = 0.$$

Therefore, the following nonnegative H -eigenvectors of (3.10) correspond to the positive eigenvalue $\lambda = 1$

$$\mathbf{u}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{u}_2 = \frac{1}{2} \begin{bmatrix} 3 + \sqrt{5} \\ 1 + \sqrt{5} \\ 2 \end{bmatrix}$$

and any of their nonnegative multiples.

On the other hand, $T(\mathbf{1}) = [3, 1, 1]^T > 0$, and thus $C_+(\mathbf{T}) = \mathbb{R}_{++}^3$. Thus, Theorem 3.3 implies that for any $p > d = 3$, such a tensor has a unique Perron ℓ^p -eigenvector $\mathbf{x}^*(p)$ which is entry-wise positive and which can be computed via either of Algorithms 1 and 2. Moreover, Theorem 3.6 suggests that for $p \approx d$ we have $\mathbf{x}^*(p) \approx \tilde{\mathbf{u}}_2$ with $\tilde{\mathbf{u}}_2 := \mathbf{u}_2 / \|\mathbf{u}_2\|_d$, i.e., the Perron ℓ^p -eigenvector approximates the (normalized) positive H -eigenvector. This phenomenon is shown in Figure 3.1, which plots the values of the entries of $\mathbf{x}^*(d + \varepsilon)$ and the distance $d_H(\mathbf{x}^*(d + \varepsilon), \tilde{\mathbf{u}}_2)$ for the tensor in (3.10), as $\varepsilon > 0$ decreases to zero.

4. Numerical experiments: part 1. In this section we investigate experimentally for different test problems the power method in its two shifted variants described in Algorithms 1 and 2 for different values of p and of the shift σ . For the sake of clarity we use different sections to describe the problem set and present the corresponding numerical results. All the numerical experiments are performed on a laptop running Linux with 16Gb memory and CPU Intel® Core™ i7-4510U with clock 2.00GHz. The code is written and executed in MATLAB R2018b.

Let \mathbf{x}_k be the sequence of vectors generated by any of the methods when applied to address the ℓ^p -eigenvector problem for the tensor \mathbf{T} , and let $\lambda_k = f_p(\mathbf{x}_k)$ be defined by (2.3). All the presented figures in the following sections show the behavior of the residual

$$(4.1) \quad \|T(\mathbf{x}_k) - \lambda_k \Phi_p(\mathbf{x}_k)\|_\infty$$

up to iteration 30.

It is worth pointing out that often (e.g., in [29, 34]) the convergence plots of power sequences for tensors show the relative error between consecutive iterates $\|\mathbf{x}_k - \mathbf{x}_{k+1}\|/\|\mathbf{x}_{k+1}\|$ or the behavior of the eigenvalue sequence λ_k , rather than the point-wise residual (4.1). For the sake of brevity here we do not show plots of the relative error nor of the eigenvalue sequence. However we stress that: (a) the relative error between consecutive iterates $\|\mathbf{x}_k - \mathbf{x}_{k+1}\|_p/\|\mathbf{x}_{k+1}\|_p$, as expected, always decreases at least as fast as the shown point-wise residual (4.1), and (b) the eigenvalue sequence always stabilizes for the test problems considered.

In the following numerical results, as pointed out in Section 3.2.1, we will use $p \approx d$ in order to investigate the numerical properties of Algorithms 1 or 2 when used to approximate H -eigenpairs. Let us add, moreover, that this is the only numerical relevant case since, as shown in the proof of Theorem 3.6, the contractivity of our fixed point map increases accordingly to d . Finally, let us point out that in all the following numerical results we use a random vector \mathbf{x}_0 obtained using MATLAB's function `rand` as initial guess.

In all the figures we denote by Alg1 and Alg2 the sequences computed by Algorithms 1 or 2, respectively, and by PM the power method sequence, i.e., the case where $\sigma = 0$.

4.1. Nonnegative tensors with different irreducibility pattern. As mentioned above, a shifted version of the power method for H -eigenvalue problems for nonnegative tensors has been introduced in [41]. It is simple to note that, in this case, i.e., when $p = d$, the latter method coincides with Algorithm 2 proposed above. In what follows, we refer to that method as the LZI algorithm, following the notation of [53]. The latter paper introduces the notion of weakly positive tensor and proposes a convergence analysis of the LZI algorithm, proving the linear convergence of the method for weakly positive tensors. However, due to Theorem 3.3, this requirement on the structure of the tensor is unnecessary when $p > d$. In this section we analyze the behavior of Algorithms 1 and 2 when applied to the three test problems **A**, **B**, and **C** defined in (4.2) and considered in [53]. All these three tensors are square of size $n \times n \times n$. **A** is irreducible, but not primitive nor weakly positive; **B** is primitive and weakly positive, but not essentially positive; **C** is primitive but not weakly positive. This implies that

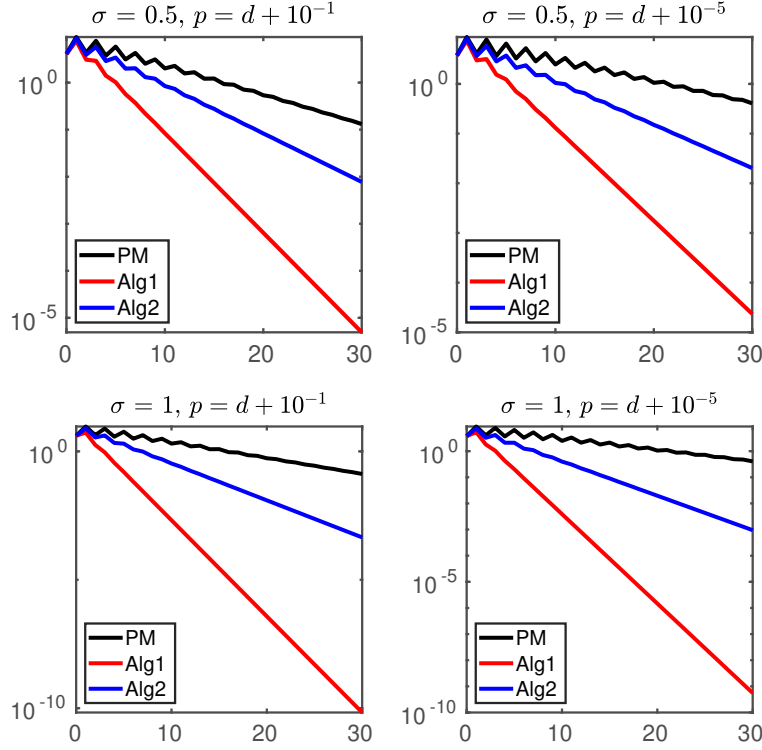


FIG. 4.1. Experiments for the test problem $\mathbf{A} \in \mathbb{R}^{n \times n \times n}$, $n = 100$, considered in [53] and defined in (4.2).

$$C_+(\mathbf{A}) = C_+(\mathbf{B}) = C_+(\mathbf{C}) = \mathbb{R}_{++}^n.$$

$$(4.2) \quad \begin{aligned} \mathbf{A} &= (a_{ijk}) \quad \text{with} \quad a_{ijk} = \begin{cases} 1 & i = 1, j = k, 2 \leq j \leq n \\ 1 & j = k = 1, 2 \leq i \leq n \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{B} &= (b_{ijk}) \quad \text{with} \quad b_{ijk} = \begin{cases} i + j & j = k, i \neq j, 1 \leq i, j \leq n \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{C} &= (c_{ijk}) \quad \text{with} \quad c_{ijk} = \begin{cases} 1 & i = 1, j = k = n \\ 1 & j = k = 1, 2 \leq i \leq n \\ 1 & i = n, 1 \leq j = k \leq n - 1 \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

For this data set, our experiments show that Algorithm 1 often outperforms the LZI algorithm [41, 53] when computing approximated H -eigenpairs, i.e., for $p \approx d$. Indeed, as Figures 4.1, 4.2, and 4.3 show, the point-wise residual of Algorithm 1 always converges to zero faster than the one of Algorithm 2.

4.2. A real symmetric tensor: Kofidis and Regalia example. One of the first appearances of the power method for tensor eigenvalues was related to Z -eigenvalues, that is, the ‘‘Euclidean’’ case $p = 2$. The symmetric higher-order power method for Z -eigenpairs have been introduced in [20]. Afterwards, Kofidis and Regalia [32] noted that the convergence of that method is guaranteed only if certain not necessarily mild conditions are met. Moreover,

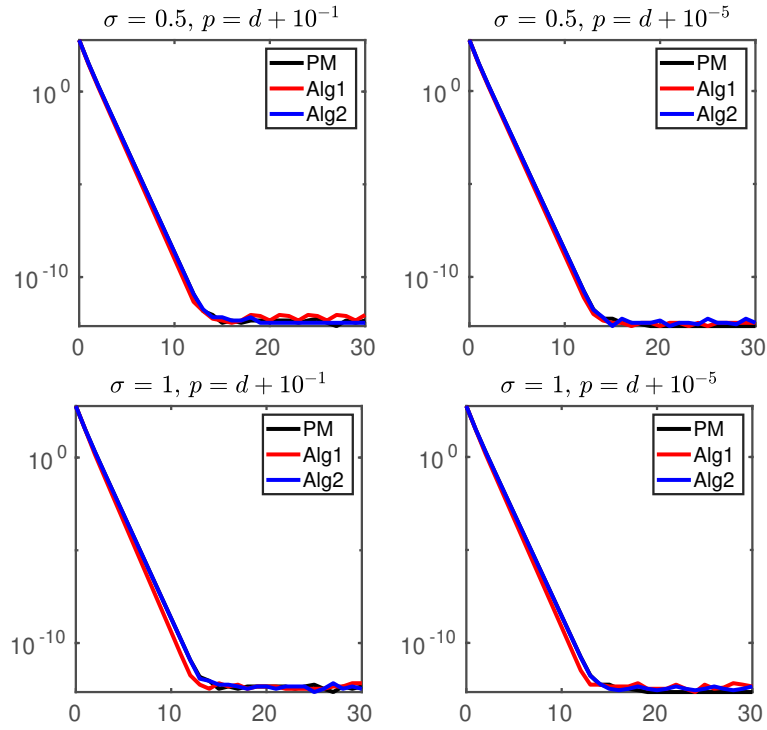


FIG. 4.2. Experiments for the test problem $\mathbf{B} \in \mathbb{R}^{n \times n \times n}$, $n = 100$, considered in [53] and defined in (4.2).

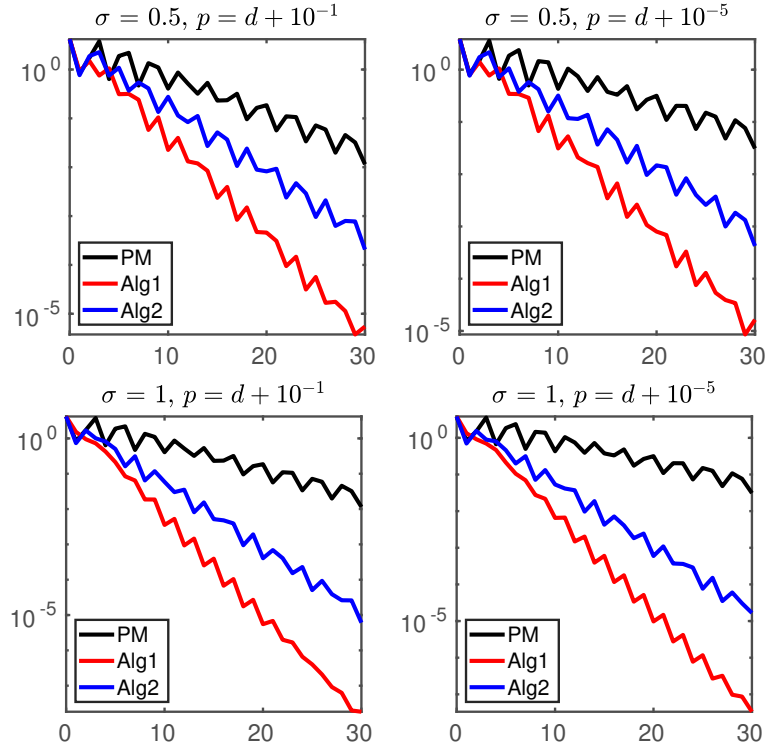


FIG. 4.3. Experiments for the test problem $\mathbf{C} \in \mathbb{R}^{n \times n \times n}$, $n = 100$, considered in [53] and defined in (4.2).

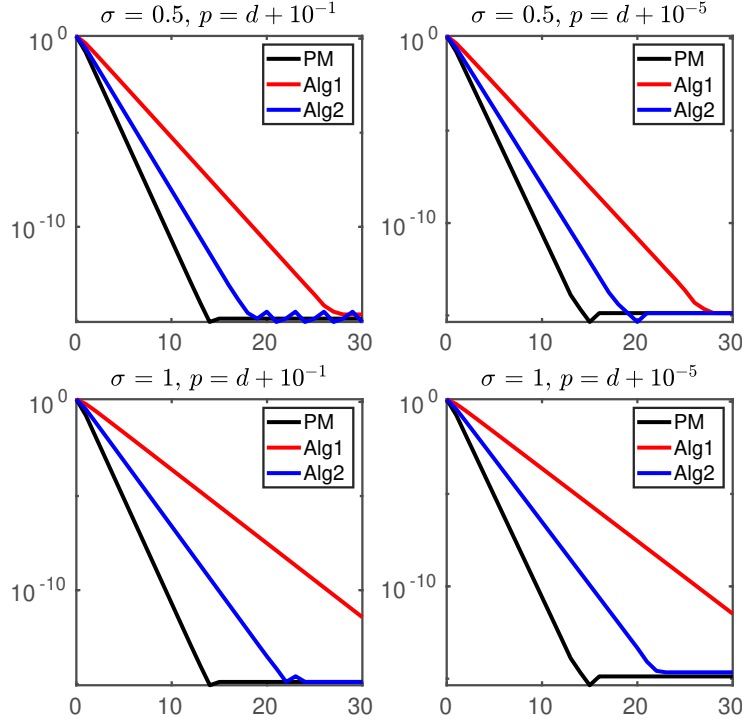


FIG. 4.4. Experiments for the test problem $\mathbf{K} \in \mathbb{R}^{n \times n \times n \times n}$.

they provide a particularly bad-behaving example tensor \mathbf{K} for which the power iteration fails to converge. We recall that example tensor below [32, Example 1]. Consider the tensor $\tilde{\mathbf{K}}$ with nonzero entries defined by

$$\tilde{\mathbf{K}} = \left\{ \begin{array}{cccc} \kappa_{1111} = 0.2883 & \kappa_{1112} = -0.0031 & \kappa_{1113} = 0.1973 & \kappa_{1122} = -0.2485 \\ \kappa_{1123} = -0.2939 & \kappa_{1133} = 0.3847 & \kappa_{1222} = 0.2972 & \kappa_{1223} = 0.1862 \\ \kappa_{1233} = 0.0919 & \kappa_{1333} = -0.3619 & \kappa_{2222} = 0.1241 & \kappa_{2223} = -0.3420 \\ \kappa_{2233} = 0.2127 & \kappa_{2333} = 0.2727 & \kappa_{3333} = -0.3054 & \end{array} \right\}.$$

The tensor \mathbf{K} is obtained from $\tilde{\mathbf{K}}$ by symmetrizing it with respect to any permutation of the indices i, j, k, ℓ .

Later on, a shifted symmetric higher-order power method (SS-HOPM) for Z -eigenpairs was proposed in [34, 47]. Clearly, when $p = 2$, the proposed Algorithms 1 and 2 both coincide with the SS-HOPM. In the following Figure 4.4 we analyze the behavior of Algorithms 1 and 2 and of the power method for the element-wise absolute value of the Kofidis and Regalia tensor, i.e., $\mathbf{T} := |\mathbf{K}| = (|\kappa_{ijkl}|) \in \mathbb{R}^{3 \times 3 \times 3 \times 3}$. In particular, plots in Figure 4.4 show the residual $\|T(\mathbf{x}_k) - \lambda_k \Phi_p(\mathbf{x}_k)\|_\infty$ from the element-wise absolute value of the same starting point $\mathbf{x}_0 = (-0.2695, 0.1972, 0.3270)$ proposed in [32, 34].

4.3. Real world network data tensors. Higher-order network analysis is gaining an increasing amount of attention since a large number of real-world complex networks show higher-order features and a higher-organization [3]. In this context one often describes higher-order network data via a nonnegative tensor and then carries out an analysis based on the eigen- or singular vectors in order to compute, for example, importance scores for network components; see, for example, [1, 2, 39, 51]. To test the performances of our methods in this

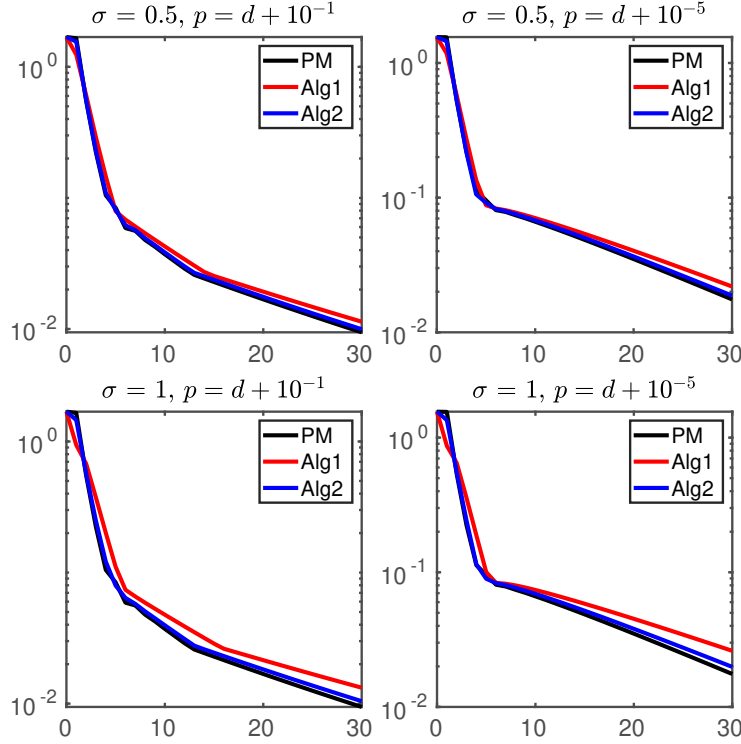
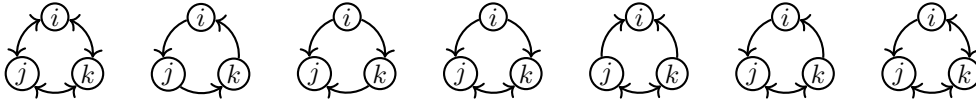


FIG. 4.5. Experiments for $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, $n = 62$, generated using *Dolphins*.

context, here we consider a network modeled by the graph $G = (V, E)$, with $V = \{1, \dots, n\}$ being the set of nodes and E the set of edges between the nodes. Then, we build the third order three-cycle tensor $\mathbf{T} = (t_{ijk})$ with entries

$$t_{ijk} = \begin{cases} 1, & \text{if there is a three-cycle between nodes } i, j, k, \\ 0, & \text{otherwise,} \end{cases}$$

where a three-cycle between i, j, k is any closed sequence of edges involving those three nodes. While there is only one possible type of three-cycles in an undirected network, a triangle between i, j , and k , in the directed case there are up to 7 different types of three-cycles as we show in the illustration below.



We then compute the Perron ℓ^p -eigenvector of \mathbf{T} for different values of $p \approx d$, in order to quantify the hypergraph centrality of the network, following the eigenvector centrality model discussed in [2]. In our tests we used four real-world directed and undirected network data sets of different size coming from [18], as listed in Table 4.1. Results up to iteration 30 are shown in Figures 4.5, 4.6, 4.7, and 4.8, where we plot the residual $\|T(\mathbf{x}_k) - \lambda_k \Phi_p(\mathbf{x}_k)\|_\infty$ for the four different three-cycle tensors obtained by the chosen data sets and for the three sequences obtained with Algorithms 1 and 2, for the choices of $\sigma = 0.5$ and $\sigma = 1$, and the unshifted power method.

TABLE 4.1

Data set information: the column ‘size’ refers to the number of nodes in the network, whereas the two columns on its right show the number of nonzeros in the adjacency matrix (i.e., twice the number of edges) and in the three-cycle tensor, respectively. All the three-cycle tensors generated from these data sets are such that $T(\mathbf{1}) \geq 0$ except for gre1107 for which strict inequality holds. Source [18].

Problem name	Size	$\text{nnz}(\text{adj}(\mathbf{G}))$	$\text{nnz}(\mathbf{T})$
Dolphins (undirected)	62	318	570
yeast (undirected)	2361	13828	35965
gre1107 (directed)	1107	5664	11045
wb-cs-stanford (directed)	9914	36854	101992

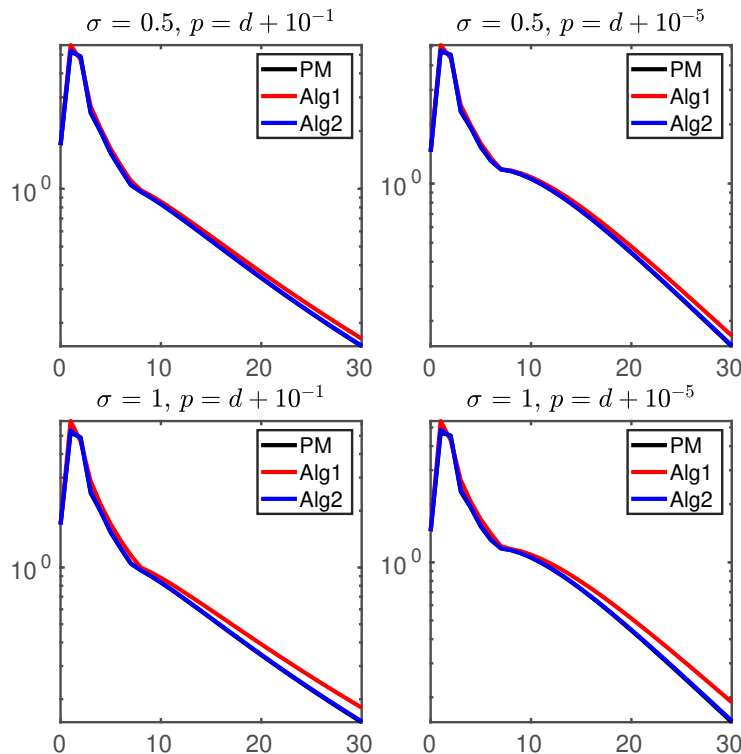


FIG. 4.6. Experiments for $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, $n = 2361$, generated using yeast.

While we observe that the residual decreases, as expected, unlike in previous example tests, none of the three methods perform well for these data sets. In fact, over 50 iterations are often not enough to achieve more than 2 digits of precision. For this reason, we introduce in the next section an extrapolation strategy. With this technique we speed-up the original sequence by using the *Simplified Topological Shanks Transformations* and the corresponding algorithms (called in short STEA’s) [7, 8, 9], obtaining extrapolated methods that achieve competitive performance for the considered network data.

5. Extrapolation for fixed-point iterations. The numerical experiments carried out in Section 4 have shown that for some problems (Section 4.3) the rate of convergence of Algorithms 1, 2 and of the power method can be quite slow. This can affect and limit the applicability of such methods for real world problems. Motivated by this observation, in

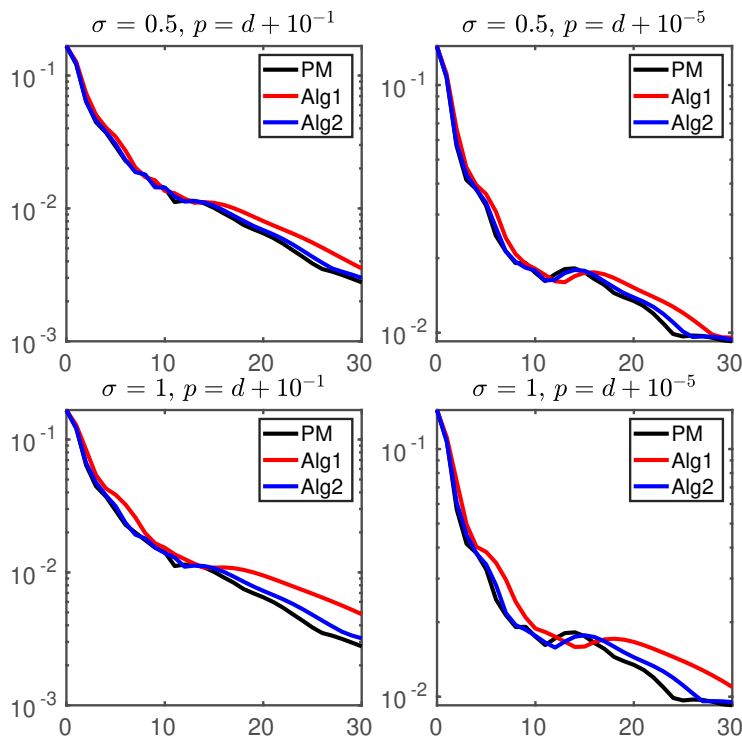


FIG. 4.7. Experiments for $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, $n = 1107$, generated using `gre1107`.

this section, we introduce extrapolation techniques for accelerating the convergence of the sequence (\mathbf{x}_k) generated by Algorithm 1 or 2.

The theory of extrapolation methods has been developed and successfully applied to a variety of problems, such as the solution of linear and nonlinear systems, matrix eigenvalue problems, the computation of matrix functions, the solution of integral equations, and many others [4, 6, 9, 15, 49]. These methods transform the original sequence (\mathbf{x}_k) into a new sequence (\mathbf{y}_k) by means of a *sequence transformation*, which, under some assumptions, converges faster to the limit. The idea behind such transformations is typically to assume that the original sequence (\mathbf{x}_k) behaves like a model sequence whose limit \mathbf{x} can be computed exactly by a finite algebraic process. The set \mathcal{K} of these model sequences is called the *kernel* of the transformation. If the sequence (\mathbf{x}_k) belongs to the kernel \mathcal{K} , then the transformed sequence “converges in one step”, i.e., the sequence is transformed into the constant sequence, where the constant is the limit of the original sequence. If the sequence (\mathbf{x}_k) does not belong to the kernel but it is “close enough” to it, then there is a good chance that the transformed sequence converges, to the same limit, faster than the original sequence.

Among the existing sequence transformations (also called *extrapolation methods*), the Shanks transformation [10, 11, 48] is arguably the best all-purpose method for accelerating the convergence of a sequence. The kernel of the vector Shanks transformation \mathcal{K}_S contains the set of sequences (\mathbf{x}_k) for which there exists an integer h such that for all k we have

$$a_0(\mathbf{x}_k - \mathbf{x}) + \dots + a_h(\mathbf{x}_{k+h} - \mathbf{x}) = 0, \quad k = 0, 1, \dots,$$

for some real coefficients a_i such that $a_0 a_h \neq 0$ and $a_0 + \dots + a_h \neq 0$. If we assume, without

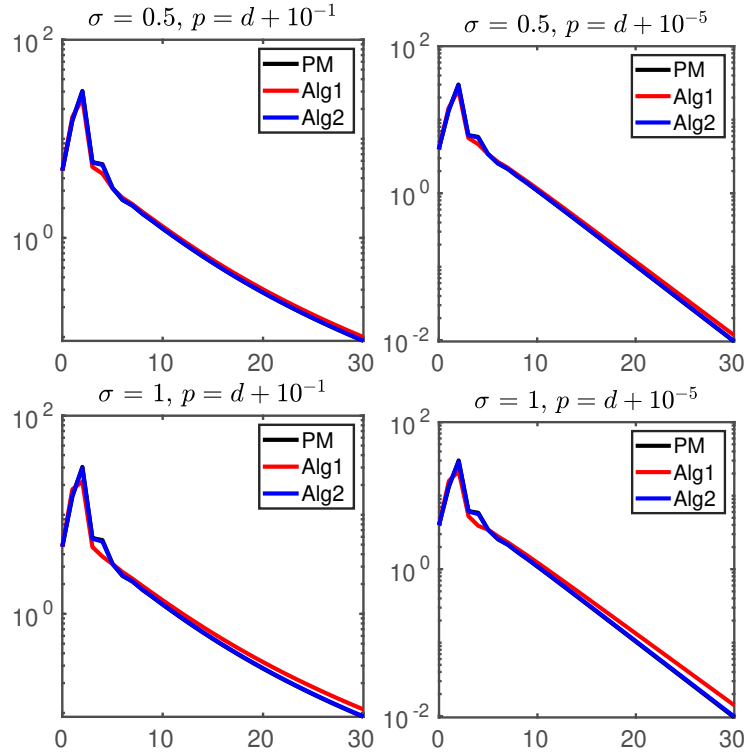


FIG. 4.8. Experiments for $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, $n = 9914$, generated using *wb-cs-stanford*.

loss of generalities, that $a_0 + \dots + a_h = 1$, then for each sequence of the kernel we have

$$\mathbf{x} = a_0 \mathbf{x}_k + \dots + a_h \mathbf{x}_{k+h}.$$

Of course, even if the given sequence (\mathbf{x}_k) does not belong to the Shanks kernel, we may apply the transformation; However, in this case we obtain a set of sequences, usually denoted $(\mathbf{e}_h(\mathbf{x}_k))$, depending on k and h , whose elements are given by

$$(5.1) \quad \mathbf{e}_h(\mathbf{x}_k) = a_0^{(k,h)} \mathbf{x}_k + \dots + a_h^{(k,h)} \mathbf{x}_{k+h},$$

where the coefficients $a_i^{(k,h)}$ are such that if $(\mathbf{x}_k) \in \mathcal{K}_S$, then $\mathbf{e}_h(\mathbf{x}_k) = \mathbf{x}$ for all k . For a recent survey on Shanks transformations, see [11].

Some of these Shanks transformations can be implemented recursively, and, among these, the Topological Shanks Transformations can be implemented by the Topological Epsilon Algorithms, in short TEA's [5]. Recently, simplified versions of these algorithms, called the Simplified Topological Epsilon Algorithms (STEA's), have been introduced. These simplified algorithms have three main advantages with respect to the original ones: the numerical stability can be significantly improved, the rules defining the extrapolated sequence are simpler and the computational cost is reduced both in terms of memory allocation and in terms of operations to be performed.

Finally, let us observe that Shanks transformations can be coupled with a restarting technique which is particularly suited for fixed-point problems (see [9, 15]) and that roughly goes as follows. Assume that we are interested in a fixed-point \mathbf{x} of a mapping $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$. We compute a certain number of basic iterates $\mathbf{x}_{k+1} = \mathcal{F}(\mathbf{x}_k)$ from a given \mathbf{x}_0 . Then we

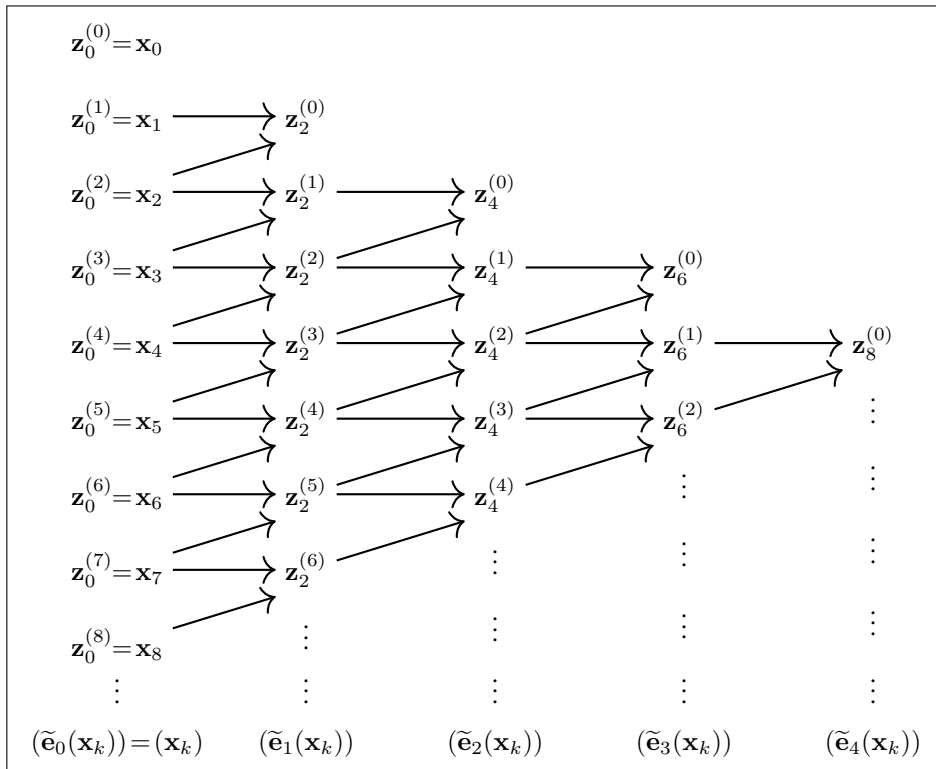


FIG. 5.1. Diagrammatic illustration of the “triangular” recursive rule (5.4) for the computation of the extrapolated sequence $(\tilde{\mathbf{e}}_h(\mathbf{x}_k)) = (\mathbf{z}_{2h}^{(k)})$, for $h = 1, 2, 3, 4$. The arrows emphasize the dependence of $\mathbf{z}_{2j+2}^{(i)}$ upon only $\mathbf{z}_{2j}^{(i+2)}$ and $\mathbf{z}_{2j}^{(i+1)}$.

apply the extrapolation algorithm to them, and we restart the basic iterates from the computed extrapolated term and so on. The advantage of this approach is that, under suitable regularity assumptions on \mathcal{F} and if h is large enough, the sequence generated in this way converges quadratically to the fixed point of \mathcal{F} [36].

In Section 5.1, we briefly review the topological Shanks transformations and their simplified versions (in particular the STEA2 algorithm that is the less expensive in terms of memory requirement). For further details, see [7, 9]. In Section 5.2 we discuss the details of the restarted procedure and how this applies to the specific ℓ^p -eigenvector setting.

5.1. Topological Shanks transformations and the STEA2 algorithm. For $p > d$, let (\mathbf{x}_k) be the sequence generated by either Algorithm 1 or 2, and let \mathbf{x} be the Perron ℓ^p -eigenvector limit of this sequence. The so-called *second Topological Shanks Transformation* starts from the original sequence and, given an arbitrary nonzero $\mathbf{y} \in \mathbb{R}^n$ and a fixed integer h , produces a new sequence $(\tilde{\mathbf{e}}_h(\mathbf{x}_k))$ defined as

$$(5.2) \quad \tilde{\mathbf{e}}_h(\mathbf{x}_k) = a_0^{(k,h)} \mathbf{x}_{k+h} + \cdots + a_h^{(k,h)} \mathbf{x}_{k+2h},$$

Algorithm 3: Restarted extrapolation method.

Input: Choose $2h$, cycles $\in \mathbb{N}$, \mathbf{x}_0 and $\mathbf{y} \in \mathbb{R}^n$

- 1 **for** $i = 0, 1, \dots$, cycles (*outer iterations*) **do**
- 2 Compute $x_0 = \mathbf{y}^T \mathbf{x}_0$
- 3 **for** $k = 1, \dots, 2h$ (*inner iterations*) **do**
- 4 Compute $\mathbf{x}_k = \mathcal{F}(\mathbf{x}_{k-1})$
- 5 Compute $x_k = \mathbf{y}^T \mathbf{x}_k$
- 6 **end**
- 7 Apply STEA2 to $\mathbf{x}_0, \dots, \mathbf{x}_{2h}$ and x_0, \dots, x_{2h} to compute $\mathbf{z}_{2h}^{(0)} = \tilde{\mathbf{e}}_h(\mathbf{x}_0)$
- 8 Set $\mathbf{x}_0 = \mathbf{z}_{2h}^{(0)}$
- 9 Choose $\mathbf{y} \in \mathbb{R}^n$
- 10 **end**

where the coefficients $a_i^{(k,h)}$ are the solutions of the linear system

$$(5.3) \quad \begin{bmatrix} 1 & \dots & 1 \\ b_0 & \dots & b_h \\ \vdots & & \vdots \\ b_{h-1} & \dots & b_{2h-1} \end{bmatrix} \begin{bmatrix} a_0^{(k,h)} \\ \vdots \\ a_h^{(k,h)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and $b_i := \mathbf{y}^T(\mathbf{x}_{k+i+1} - \mathbf{x}_{k+i})$, for $i = 0, \dots, 2h - 1$. Note that for this transformation it holds that if $(\mathbf{x}_k) \in \mathcal{K}_S$, then $\tilde{\mathbf{e}}_h(\mathbf{x}_k) = \mathbf{x}$ for all k , i.e., all the transformed terms coincide with the limit. However, we often do not know whether the original sequence satisfies (5.1) and, if so, we do not know what is the correct value of h . So, in practice, we fix an arbitrary integer h and transform the sequence via (5.2) using such an integer.

The second Simplified Topological ε -Algorithm (STEA2) [7] allows us to compute the terms of the new sequence $\tilde{\mathbf{e}}_h(\mathbf{x}_k)$ via four equivalent recursive formulas without solving explicitly the linear system given by (5.3). Here we focus on the third one.

Define the vectors $\mathbf{z}_{2j}^{(i)}$ as follows: set $\mathbf{z}_0^{(k)} = \mathbf{x}_k$ and, for $i, j = 0, 1, \dots$, compute

$$(5.4) \quad \mathbf{z}_{2j+2}^{(i)} = \mathbf{z}_{2j}^{(i+1)} + \frac{\varepsilon_{2j+2}^{(i)} - \varepsilon_{2j}^{(i+1)}}{\varepsilon_{2j}^{(i+2)} - \varepsilon_{2j}^{(i+1)}} (\mathbf{z}_{2j}^{(i+2)} - \mathbf{z}_{2j}^{(i+1)}),$$

where the scalar quantities $\varepsilon_j^{(i)}$ are given by Wynn's scalar ε -algorithm [52] (an algorithm implementing Shanks transformation for scalar sequences) applied to the sequence $(\mathbf{y}^T \mathbf{x}_0), (\mathbf{y}^T \mathbf{x}_1), (\mathbf{y}^T \mathbf{x}_2), \dots$. The link with the transformation is given by the fact that we have $\mathbf{z}_{2h}^{(k)} = \tilde{\mathbf{e}}_h(\mathbf{x}_k)$ (see, e.g., [5]), and thus (5.4) allows us to compute the desired extrapolated sequence $(\tilde{\mathbf{e}}_h(\mathbf{x}_k))$ with a few simple operations. Indeed, note that (5.4) contains only sums and differences of vectors, and it relies only on two terms of a triangular scheme, as illustrated in Figure 5.1. Of course, there are also the inner products for the initialization of Wynn's scalar ε -algorithm.

5.2. Restarted extrapolation method for ℓ^p -eigenpairs. When dealing with fixed-point problems $\mathcal{F}(\mathbf{x}) = \mathbf{x}$, as previously pointed out, it is often advisable to couple the extrapolation method with a restarting technique [9]. If we consider the STEA2 scheme, the general restarted method is presented in Algorithm 3.

It is important to remark that, when $h = n$, where n is the dimension of the problem, the sequence generated by Algorithm 3 converges quadratically to \mathbf{x} , under suitable regularity assumptions on \mathcal{F} ; see [36].

Since all the algorithms we took into consideration in Section 3 are based on a fixed-point iteration, for all of them we consider the restarted extrapolation method described in Algorithm 3. In our case, \mathcal{F} is either the iteration map of Algorithm 1 or that of Algorithm 2. Concerning the computational complexity and the storage requirements, in our experimental investigations, we used the public available software EPSfun MATLAB toolbox, na44 package in Netlib [8] that contains optimized versions of the STEA algorithms. The $2h + 1$ vectors $\mathbf{x}_0, \dots, \mathbf{x}_{2h}$ are computed together with the extrapolation scheme, and thus only $h + 2$ vectors of dimension n have to be stored in order to compute $\tilde{\mathbf{e}}_h(\mathbf{x}_0)$; see [8, 9] for implementation details. Observe that, in addition to applying the power method map \mathcal{F} , each outer cycle also requires the computation of $2h + 1$ scalar products.

The practical implementation and the performance of the outlined method rely on two key parameter choices: the choice of h and the choice of $\mathbf{y} \in \mathbb{R}^n$. As described above, the choice of h is connected to the memory requirement and influences the rate of convergence. In general, also for relatively small problems, we choose a value of h smaller than n (the dimension of the problem). This is the case, for instance, for the real-world examples of Section 6. Concerning the choice of $\mathbf{y} \in \mathbb{R}^n$, this is a well-known critical point in the topological Shanks transformations, and it is usually addressed by model-dependent heuristics. Indeed, no general theoretical result has been obtained so far concerning the selection of an optimal $\mathbf{y} \in \mathbb{R}^n$. In our examples, in each cycle we take a different \mathbf{y} , chosen as $\mathbf{y} = \tilde{\mathbf{e}}_h(\mathbf{x}_0)$, i.e., the last extrapolated term, except for in the first cycle, where it is set equal to \mathbf{x}_0 . The quality of this choice is supported by the performance we obtained, and by the fact that in all our tests the resulting extrapolated vectors computed with such a choice of \mathbf{y} belong to the desired cone $C_+(\mathbf{T})$.

6. Numerical experiments: part 2. In this section we present several numerical experiments to demonstrate the advantages of the extrapolation framework. We focus only on problems from Section 4.3 that exhibit a *slow* convergence rate (even if we observe similar acceleration performance on all the data sets). For the same reason we focus only on the case $p = d + 10^{-5}$. As before, in all our experiments we consider the point-wise eigenvector residual

$$\|T(\mathbf{x}_k) - \lambda_k \Phi_p(\mathbf{x}_k)\|_\infty$$

evaluated at the current iteration step. Our numerical results are focused on the analysis of the rate of convergence of the accelerated sequence when compared to the original one. To this end, we run Algorithm 3 for a prescribed number of inner and outer iterations (i.e., we fix the value of h and the number of cycles) without any other stopping criterion. Results are shown in Figures 6.1, 6.2, 6.3, and 6.4, where we highlight with a circle each restart of the outer loop, i.e., the residual generated by the iterate defined at Line 8 of Algorithm 3.

The linear functional \mathbf{y} is updated at the end of each outer cycle by choosing $\mathbf{y} = \tilde{\mathbf{e}}_h(\mathbf{x}_0)$ (for the first extrapolation step we choose $\mathbf{y} = \mathbf{x}_0$). For the implementation of the simplified topological ε -algorithm we used the public domain MATLAB toolbox EPSfun [8]. For all the reported numerical results we choose a random starting point.

As Figures 6.1, 6.2, 6.3, and 6.4 show, the introduction of extrapolation techniques for the computation of ℓ^p -eigenpairs greatly improves the rate of convergence of the original fixed point Algorithms 1 and 2 at a cost of one extra scalar product per fixed point iteration; see Section 5.

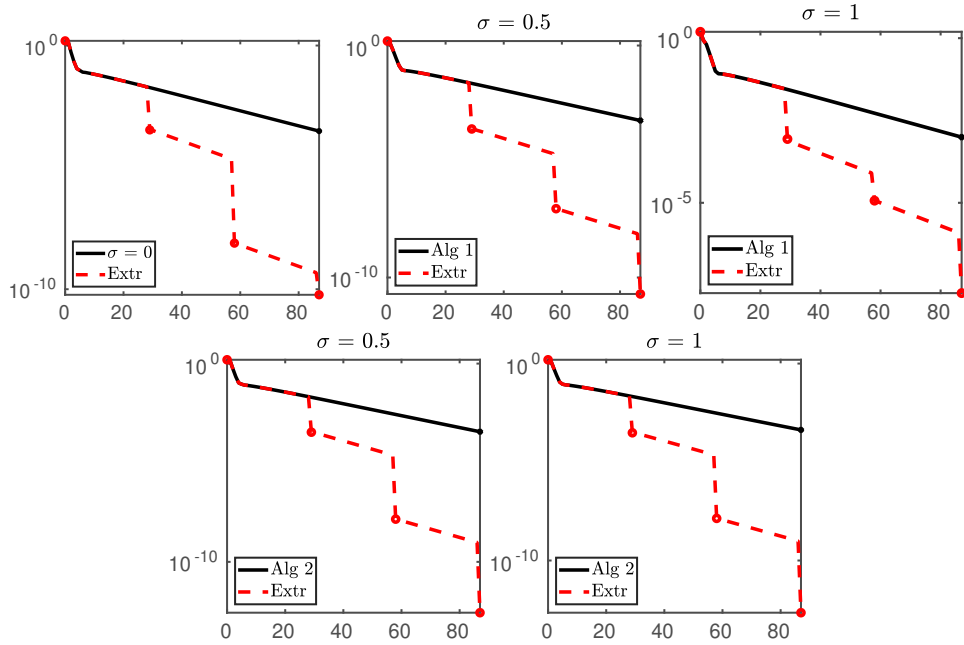


FIG. 6.1. *STEA2* on $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, $n = 62$, generated using *dolphins*. $2h = 28$, cycles = 3, $p = d + 10^{-5}$.

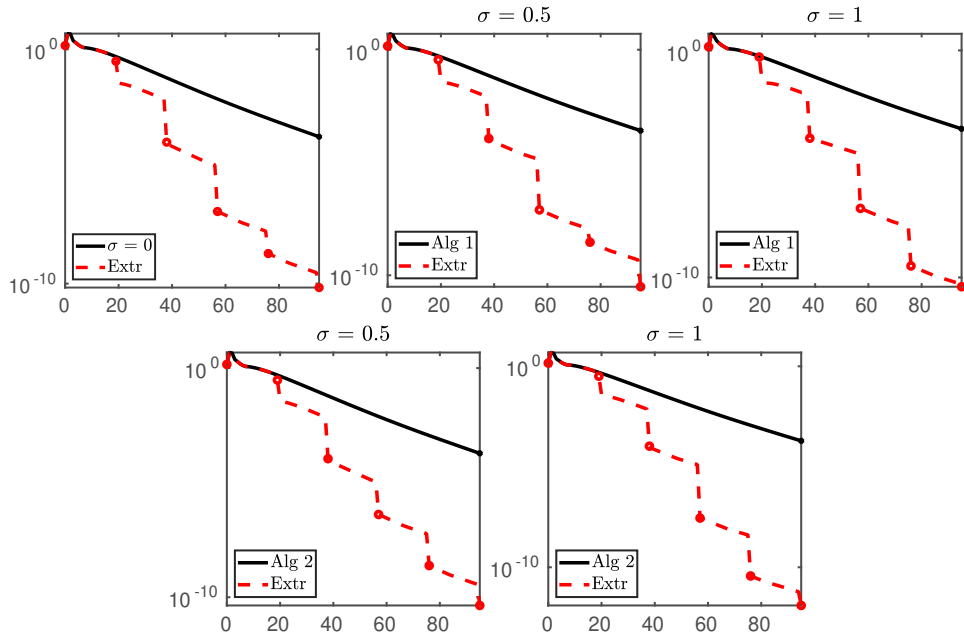


FIG. 6.2. *STEA2* on $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, $n = 2361$, generated using *yeast*. $2h = 12$, cycles = 6, $p = d + 10^{-5}$.

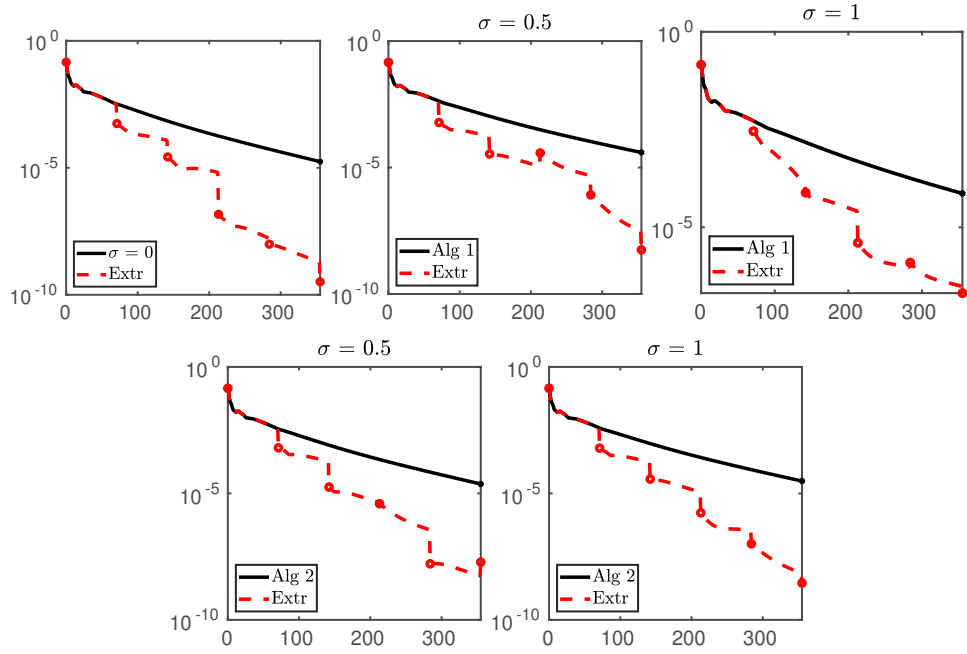


FIG. 6.3. *STEA2* on $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, $n = 1107$, generated using *gre1107*. $2h = 70$, cycles = 5, $p = d + 10^{-5}$.

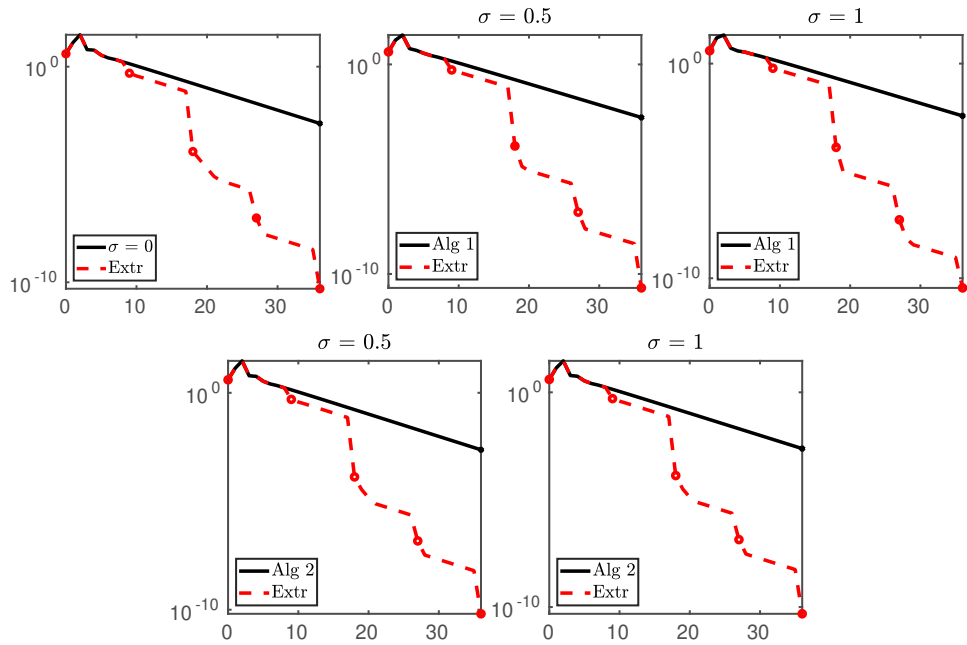


FIG. 6.4. *STEA2* on $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, $n = 9914$, generated using *stanford*. $2h = 8$, cycles = 4, $p = d + 10^{-5}$.

TABLE 6.1
Comparison of execution times to produce residuals smaller than 10^{-9} , $\sigma = 0$.

Problem	Time(s)	
	<i>Extrapolated</i>	<i>Not Extrapolated</i>
Dolphins	0.2373	0.3062
yeast	0.4199	0.7193
grell107	1.4225	1.8462
wb-cs-stanford	0.3817	0.6994

Finally, let us point out that even though the acceleration is quite evident for the shifted versions of Algorithms 1 and 2, acceleration seems to happen more systematically on the unshifted ones. We find this aspect particularly interesting, and we believe it deserves further investigation.

In Table 6.1, we report the comparison of the execution times needed to produce a residual smaller than 10^{-9} for the unshifted extrapolated scheme and the non-extrapolated one (the execution times of the other extrapolated schemes are all very similar).

7. Conclusions. In this work we introduced two new shifted power methods for computing ℓ^p -eigenpairs of tensors. We proved that the introduced algorithms are guaranteed to converge for entry-wise nonnegative tensors with possibly reducible patterns and for $p > d$, where d is the number of modes of the tensor. Moreover, we show that for nonnegative tensors the Perron ℓ^p -eigenvector depends continuously on the parameter p . This result, together with the global convergence guarantees of the shifted power methods, allows us to propose the first method that can provably approximate the positive Perron H -eigenvector of a nonnegative tensor by choosing $p \approx d$. The methods may suffer from a slow rate of convergence, as we observe in the numerical experiments proposed in Section 4. For this reason, we also discuss the use of the simplified ε -algorithm to extrapolate the power sequence and accelerate its convergence. The numerical experiments in Section 6 show that the application of the proposed extrapolation method substantially improves the convergence rate of the power methods for ℓ^p -eigenvectors at the price of a minor additional cost per step.

Acknowledgements. The work of M. R.-Z. was partially supported by the University of Padua, Project no. DOR1903575/19. The work of S. C. was partially supported by the GNCS – INdAM project “Efficient Methods for large scale problems with applications to data analysis and preconditioning”. The work F. T. was funded by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie individual fellowship “MAGNET” No 744014. All the authors are members of the INdAM Research group GNCS.

REFERENCES

- [1] F. ARRIGO AND F. TUDISCO, *Multi-dimensional, multilayer, nonlinear and dynamic HITS*, in Proceedings of the 2019 SIAM International Conference on Data Mining, T. Berger-Wolf and N. Chawla, eds., SIAM, Philadelphia, 2019, pp. 369–377.
- [2] A. R. BENSON, *Three hypergraph eigenvector centralities*, SIAM J. Math. Data Sci., 1 (2019), pp. 293–312.
- [3] A. R. BENSON, D. F. GLEICH, AND J. LESKOVEC, *Higher-order organization of complex networks*, Science, 353 (2016), pp. 163–166.
- [4] A. BOUHAMIDI, K. JBILOU, L. REICHEL, AND H. SADOK, *An extrapolated TSVD method for linear discrete ill-posed problems with Kronecker structure*, Linear Algebra Appl., 434 (2011), pp. 1677–1688.
- [5] C. BREZINSKI, *Généralisations de la transformation de Shanks, de la table de Padé et de l’ ε -algorithme*, Calcolo, 12 (1975), pp. 317–360.
- [6] C. BREZINSKI AND M. REDIVO-ZAGLIA, *Extrapolation methods. Theory and Practice*, North-Holland, Amsterdam, 1991.

- [7] ———, *The simplified topological ε -algorithms for accelerating sequences in a vector space*, SIAM J. Sci. Comput., 36 (2014), pp. A2227–A2247.
- [8] ———, *EPSfun: a Matlab toolbox for the simplified topological ε -algorithm*, Netlib, na44package, 2017. <http://www.netlib.org/numeralgo/>
- [9] ———, *The simplified topological ε -algorithms: software and applications*, Numer. Algorithms, 74 (2017), pp. 1237–1260.
- [10] ———, *The genesis and early developments of Aitken's process, Shanks' transformation, the ε -algorithm, and related fixed point methods*, Numer. Algorithms, 80 (2019), pp. 11–133.
- [11] C. BREZINSKI, M. REDIVO-ZAGLIA, AND Y. SAAD, *Shanks sequence transformations and Anderson acceleration*, SIAM Rev., 60 (2018), pp. 646–669.
- [12] D. CARTWRIGHT AND B. STURMFELS, *The number of eigenvalues of a tensor*, Linear Algebra Appl., 438 (2013), pp. 942–952.
- [13] K. CHANG, L. QI, AND G. ZHOU, *Singular values of a real rectangular tensor*, J. Math. Anal. Appl., 370 (2010), pp. 284–294.
- [14] K. C. CHANG, K. PEARSON, AND T. ZHANG, *Perron-Frobenius theorem for nonnegative tensors*, Commun. Math. Sci., 6 (2008), pp. 507–520.
- [15] S. CIPOLLA, M. REDIVO-ZAGLIA, AND F. TUDISCO, *Extrapolation methods for fixed-point multilinear PageRank computations*, Numer. Linear Algebra Appl., 27 (2020), Art. e2280, 22 pages.
- [16] P. COMON, *Tensor Decompositions: State of the Art and Applications*, in Mathematics in Signal Processing V, J. G. McWhirter and I. K. Proudler, eds., vol. 71 of Inst. Math. Appl. Conf. Ser. New Ser., Oxford Univ. Press, Oxford, 2002, pp. 1–24.
- [17] 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.
- [18] T. A. DAVIS AND Y. HU, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), Art. 1, 25 pages.
- [19] W. F. DE LA VEGA, M. KARPINSKI, R. KANNAN, AND S. VEMPALA, *Tensor decomposition and approximation schemes for constraint satisfaction problems*, in STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, ACM, New York, 2005, pp. 747–754.
- [20] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *On the best rank-1 and rank- (R_1, R_2, \dots, R_N) approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.
- [21] M. EDELSTEIN, *On fixed and periodic points under contractive mappings*, J. London Math. Soc., 37 (1962), pp. 74–79.
- [22] D. FASINO AND F. TUDISCO, *Ergodicity coefficients for higher-order stochastic processes*, Preprint on arXiv, 2019. <https://arxiv.org/abs/1907.04841>
- [23] S. FRIEDLAND, *Best rank one approximation of real symmetric tensors can be chosen symmetric*, Front. Math. China, 8 (2013), pp. 19–40.
- [24] S. FRIEDLAND, S. GAUBERT, AND L. HAN, *Perron-Frobenius theorem for nonnegative multilinear forms and extensions*, Linear Algebra Appl., 438 (2013), pp. 738–749.
- [25] A. GAUTIER AND M. HEIN, *Tensor norm and maximal singular vectors of nonnegative tensors - a Perron-Frobenius theorem, a Collatz-Wielandt characterization and a generalized power method*, Linear Algebra Appl., 505 (2016), pp. 313–343.
- [26] A. GAUTIER AND F. TUDISCO, *The contractivity of cone-preserving multilinear mappings*, Nonlinearity, 32 (2019), pp. 4713–4728.
- [27] A. GAUTIER, F. TUDISCO, AND M. HEIN, *The Perron-Frobenius theorem for multi-homogeneous mappings*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 1179–1205.
- [28] ———, *A unifying Perron-Frobenius theorem for nonnegative tensors via multi-homogeneous maps*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 1206–1231.
- [29] D. F. GLEICH, L.-H. LIM, AND Y. YU, *Multilinear PageRank*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1507–1541.
- [30] C. J. HILLAR AND L.-H. LIM, *Most tensor problems are NP-hard*, J. ACM, 60 (2013), Art. 45, 39 pages.
- [31] S. HU, L. QI, AND G. ZHANG, *Computing the geometric measure of entanglement of multipartite pure states by means of non-negative tensors*, Phys. Rev. A, 93 (2016), Art. 012304, 7 pages.
- [32] E. KOFIDIS AND P. A. REGALIA, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.
- [33] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
- [34] T. G. KOLDA AND J. R. MAYO, *Shifted power method for computing tensor eigenpairs*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1095–1124.
- [35] J. LANG AND D. EDMUNDS, *Eigenvalues, Embeddings and Generalised Trigonometric Functions*, Springer, Heidelberg, 2011.
- [36] H. LE FERRAND, *The quadratic convergence of the topological epsilon algorithm for systems of nonlinear equations*, Numer. Algorithms, 3 (1992), pp. 273–283.

- [37] B. LEMMENS AND R. NUSSBAUM, *Birkhoff's version of Hilbert's metric and its applications in analysis*, in Handbook of Hilbert Geometry, A. Papadopoulos and M. Troyanov, vol. 22 of IRMA Lect. Math. Theor. Phys., Eur. Math. Soc., Zürich, 2014, pp. 275–303.
- [38] W. LI AND M. K. NG, *On the limiting probability distribution of a transition probability tensor*, Linear Multilinear Algebra, 62 (2014), pp. 362–385.
- [39] X. LI, M. K. NG, AND Y. YE, *HAR: hub, authority and relevance scores in multi-relational data for query search*, in Proceedings of the 2012 SIAM International Conference on Data Mining, J. Ghosh, H. Liu, I Davidson, C. Domeniconi, and C. Kamath, eds., SIAM, Philadelphia, 2012, pp. 141–152.
- [40] L.-H. LIM, *Singular values and eigenvalues of tensors: a variational approach*, in 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, IEEE Conference Proceedings, Los Alamitos, 2005, pp. 129–132.
- [41] Y. LIU, G. ZHOU, AND N. F. IBRAHIM, *An always convergent algorithm for the largest eigenvalue of an irreducible nonnegative tensor*, J. Comput. Appl. Math., 235 (2010), pp. 286–292.
- [42] M. NG, L. QI, AND G. ZHOU, *Finding the largest eigenvalue of a nonnegative tensor*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1090–1099.
- [43] Q. NGUYEN, F. TUDISCO, A. GAUTIER, AND M. HEIN, *An efficient multilinear optimization framework for hypergraph matching*, IEEE Trans. Pattern Anal. Mach. Intell., 39 (2017), pp. 1054–1075.
- [44] C. L. NIKIAS AND J. M. MENDEL, *Signal processing with higher-order spectra*, IEEE Signal Process. Mag., 10 (1993), pp. 10–37.
- [45] L. QI AND Z. LUO, *Tensor Analysis: Spectral Theory and Special Tensors*, SIAM, Philadelphia, 2017.
- [46] L. QI AND K. L. TEO, *Multivariate polynomial minimization and its application in signal processing*, J. Global Optim., 26 (2003), pp. 419–433.
- [47] P. A. REGALIA AND E. KOFIDIS, *Monotonic convergence of fixed-point algorithms for ICA*, IEEE Trans. Neural Netw. Learn. Syst., 14 (2003), pp. 943–949.
- [48] D. SHANKS, *Non-linear transformations of divergent and slowly convergent sequences*, J. Math. and Phys., 34 (1955), pp. 1–42.
- [49] A. SIDI, *Vector Extrapolation Methods with Applications*, SIAM, Philadelphia, 2017.
- [50] A. SWAMI, G. B. GIANNAKIS, AND G. ZHOU, *Bibliography on higher-order statistics*, Signal Process., 60 (1997), pp. 65–126.
- [51] F. TUDISCO, F. ARRIGO, AND A. GAUTIER, *Node and layer eigenvector centralities for multiplex networks*, SIAM J. Appl. Math., 78 (2018), pp. 853–876.
- [52] P. WYNN, *On a device for computing the $e_m(S_n)$ transformation*, Math. Tables Aids Comput., 10 (1956), pp. 91–96.
- [53] L. ZHANG, L. QI, AND Y. XU, *Linear convergence of the LZI algorithm for weakly positive tensors*, J. Comput. Math., 30 (2012), pp. 24–33.
- [54] G. ZHOU, L. QI, AND S.-Y. WU, *Efficient algorithms for computing the largest eigenvalue of a nonnegative tensor*, Front. Math. China, 8 (2013), pp. 155–168.