Noname manuscript No.
(will be inserted by the editor)

Accelerating the Sinkhorn-Knopp iteration by Arnoldi-type methods

A. Aristodemo · L. Gemignani

the date of receipt and acceptance should be inserted later

Abstract It is shown that the problem of balancing a nonnegative matrix by positive diagonal matrices can be recast as a nonlinear eigenvalue problem with eigenvector nonlinearity. Based on this equivalent formulation some adaptations of the power method and Arnoldi process are proposed for computing the dominant eigenvector which defines the structure of the diagonal transformations. Numerical results illustrate that our novel methods accelerate significantly the convergence of the customary Sinkhorn-Knopp iteration for matrix balancing in the case of clustered dominant eigenvalues.

Keywords Sinkhorn-Knopp iteration, Nonlinear Eigenvalue Problem, Power method, Arnoldi method

1 Introduction

Many important types of data, like text, sound, event logs, biological sequences, can be viewed as graphs connecting basic data elements. Networks provide a powerful tool for describing the dynamic behavior of systems in biology, computer science, information engineering. Networks and graphs are generally represented as very large nonnegative matrices describing either the network topology, quantifying certain attributes of nodes or exhibiting the correlation between certain node features. Among the challenging theoretical and computational problems with these matrices there are the balancing/scalability issues.

The work of L. Gemignani was partially supported by the GNCS/INdAM project "Tecniche Innovative per Problemi di Algebra Lineare" and by the University of Pisa (grant PRA 2017-05).

A. Aristodemo
Dipartimento di Matematica, Università di Pisa, Largo B. Pontecorvo, 5 - 56127 Pisa, Italy, E-mail: aristodemo@student.dm.unipi.it

L. Gemignani
Dipartimento di Informatica, Università di Pisa, Largo Bruno Pontecorvo, 3 - 56127 Pisa, Italy,E-mail: luca.gemignani@unipi.it

The Sinkhorn-Knopp (SK) balancing problem can be stated as follows: Given a nonnegative matrix $A \in \mathbb{R}^{n\times n}$ $(A \geq 0)$, find if they exist two nonnegative diagonal matrices $D_1, D_2 \in \mathbb{R}^{n\times n}$ such that $S = D_1AD_2$ is doubly stochastic, i.e.,

$$D_2A^TD_1\boldsymbol{e} = \boldsymbol{e}, \quad D_1AD_2\boldsymbol{e} = \boldsymbol{e}, \quad \boldsymbol{e} = [1, \dots, 1]^T. \tag{1}$$

The problem was raised in three different papers [35–37] that contain the well-known iteration for matrix balancing that bears their names. Several equilibration problems exist in which row or column norms are not equal but rather are specified by positive vectors. Variants of the SK problem have attracted attention in various fields of pure and applied sciences including input-output analysis in economics [31], optimal transportation theory and its applications in machine learning [11], complex network analysis [6, 19], probabilistic and statistical modeling [33], optimization of traffic and telecommunication flows [21] and matrix preconditioning [12]. For a general review and summary of these applications one can see [16].

For any admissible vector $\boldsymbol{v} \in \mathbb{R}^n$ and any $\alpha \in \mathbb{Z}$, $\alpha \neq 0$, let $\mathcal{D}^\alpha(\boldsymbol{v})$ be defined as the $n \times n$ diagonal matrix with diagonal entries $d_i = v_i^\alpha$, $1 \leq i \leq n$. Then the computation in (1) amounts to find two vectors $\boldsymbol{r}$ and $\boldsymbol{c}$ such that $D_1 = \mathcal{D}(\boldsymbol{r})$ and $D_2 = \mathcal{D}(\boldsymbol{c})$ satisfy

$$\begin{cases} \mathcal{D}(\boldsymbol{c})A^T\mathcal{D}(\boldsymbol{r})\boldsymbol{e} = \mathcal{D}(\boldsymbol{c})A^T\boldsymbol{r} = \mathcal{D}(A^T\boldsymbol{r})\boldsymbol{c} = \boldsymbol{e}; \\ \mathcal{D}(\boldsymbol{r})A\mathcal{D}(\boldsymbol{c})\boldsymbol{e} = \mathcal{D}(\boldsymbol{r})A\boldsymbol{c} = \mathcal{D}(A\boldsymbol{c})\boldsymbol{r} = \boldsymbol{e}. \end{cases}$$

When A is symmetric we can determine $\boldsymbol{r} = \boldsymbol{c} = \boldsymbol{z}$ to satisfy $\mathcal{D}(A\boldsymbol{z})\boldsymbol{z} = \mathcal{D}(\boldsymbol{z})A\boldsymbol{z} = \boldsymbol{e}$. In [37] the authors proposed the following fixed point iteration –called Sinkhorn-Knopp (SK) iteration– for computing the desired vectors $\boldsymbol{r}$ and $\boldsymbol{c}$:

$$\begin{cases} \boldsymbol{c}_{k+1} = \mathcal{D}^{-1}(A^T\boldsymbol{r}_k)\boldsymbol{e}; \\ \boldsymbol{r}_{k+1} = \mathcal{D}^{-1}(A\boldsymbol{c}_{k+1})\boldsymbol{e}. \end{cases} \tag{2}$$

In the symmetric case the SK iteration reduces to

$$\boldsymbol{z}_{k+1} = \mathcal{D}^{-1}(A\boldsymbol{z}_k)\boldsymbol{e} \tag{3}$$

or, equivalently, by setting $1./\boldsymbol{z}_k := \mathcal{D}^{-1}(\boldsymbol{z}_k)\boldsymbol{e} = \boldsymbol{x}_k$ with the assumption $1/0 = +\infty$,

$$\boldsymbol{x}_{k+1} = 1./\boldsymbol{z}_{k+1} = A\boldsymbol{z}_k = A(1./\boldsymbol{x}_k). \tag{4}$$

The SK iterations (2),(3),(4) have been rediscovered several times in different applicative contexts. Related methods are the RAS method [31] in economics, the iterative proportional fitting procedure (IPFP) in statistics and Kruithof's projection scheme [21] in optimization.

A common drawback of all these iterative algorithms is the slow convergence behavior exhibited even in deceivingly simple cases. To explain this performance gap we observe that the equations in (2) can be combined to get

$$\boldsymbol{c}_{k+1} = \mathcal{D}^{-1}(A^T\mathcal{D}^{-1}(A\boldsymbol{c}_k)\boldsymbol{e})\boldsymbol{e}, \quad k \geq 0, \tag{5}$$

which can be expressed componentwise as

$$(\boldsymbol{c}_{k+1})_s = \left(\sum_{m=1}^n a_{m,s}\left(\sum_{\ell=1}^n a_{m,\ell}(\boldsymbol{c}_k)_\ell\right)^{-1}\right)^{-1}, \quad 1 \leq s \leq n,\ k \geq 0.$$

This means that (5) is equivalent to the fixed point iteration

$$\boldsymbol{c}_{k+1} = T(\boldsymbol{c}_k), \quad T(\boldsymbol{x})_s = \left(\sum_{m=1}^{n} a_{m,s} \left(\sum_{\ell=1}^{n} a_{m,\ell}\boldsymbol{x}_\ell\right)^{-1}\right)^{-1}, \tag{6}$$

for solving

$$\boldsymbol{x} = T(\boldsymbol{x}), \quad \boldsymbol{x} \geq \boldsymbol{0}, \tag{7}$$

where T is the nonlinear operator introduced by Menon in [25,26]. Morishima [29] first dealt with the nonlinear eigenvalue problem (7) by proving its solvability over the closed simplex $\boldsymbol{x} \geq \boldsymbol{0}, \| \boldsymbol{x} \|_1 = 1$ in Euclidean $n-$space.

Our first contribution consists of a novel formulation of the fixed point problem (7) as a nonlinear eigenvalue problem with eigenvector nonlinearity (NEPv) [4,9,17] of the form

$$\boldsymbol{x} = J_T(\boldsymbol{x})\boldsymbol{x}, \quad \boldsymbol{x} \geq \boldsymbol{0}, \tag{8}$$

where $J_T(\boldsymbol{z})$ denotes the Jacobian matrix of T evaluated at the point $\boldsymbol{z}$. Although the proof is quite simple, to our knowledge this property has been completely overlooked in the literature even though it has both theoretical and computational implications.

Theoretically, it follows that the local dynamics of the original SK algorithm (5) can be described as a power method with perturbations [38] applied to the matrix $J_T(\boldsymbol{x})$ evaluated at the fixed point. Therefore the SK iterations inherit the slow convergence of the power process in the case of clustered dominant eigenvalues of $J_T(\boldsymbol{x})$. In particular, under some technical assumptions it is shown that the convergence rate[1] of (6) is $O(1/\Delta)$ where $\Delta = \lambda_1 - \lambda_2$ is the eigen-gap of $J_T(\boldsymbol{x})$ and $\lambda_1 = 1 > \lambda_2 \geq 0$ are the first two largest eigenvalues of $J_T(\boldsymbol{x})$ (compare with [19] for a quite different proof of the same estimate). Moreover, this convergence analysis also extends to certain power-based adaptations of the SK iteration (6).

Relation (8) can also be exploited practically in order to speed up the computation of the Sinkhorn-Knopp vector. Acceleration methods using nonlinear solvers applied to equation (7) have been recently proposed in [20] whereas optimization strategies and descending techniques are considered in [18, 32]. In this paper we pursue a different approach by taking into account the properties of the equivalent NEPv (8). Some adaptations of an iterative procedure referred to as the Self-Consistent Field (SCF) iteration [9] , in which a set of eigenvectors of a matrix that changes at each iteration are to be computed, are devised. Specifically, we propose here to compute an approximation of the fixed point of T by using an SCF iteration of the form

$$\lambda_k \boldsymbol{v}_{k+1} = J_T(\boldsymbol{v}_k)\boldsymbol{v}_{k+1}, \quad k \geq 0, \tag{9}$$

where λ_k is the dominant eigenvalue of $J_T(\boldsymbol{v}_k)$ with corresponding normalized eigenvector $\boldsymbol{v}_{k+1}$. The iterative scheme can be recast as an inexact Newton method applied to (8) possibly exhibiting a fast superlinear convergence. Each iteration amounts to approximate the dominant eigenpair of a matrix $J_T(\boldsymbol{v}_k)$. Krylov methods are the algorithms of choice for the computation of a few eigenvalues of largest

[1] For complexity comparisons the term 'convergence rate' here and hereafter denotes the reciprocal of the usual convergence rate and it is roughly the number of iterations required to attain a error tolerance of $1.0e-1$.

magnitude of large-scale matrices [3] and they have been proven to be efficient for achieving eigenvalue/eigenvector separation [15]. Since $J_T(\boldsymbol{v}_k)$ is diagonally similar to a symmetric semidefinite matrix fast eigensolvers relying upon the Lanczos process are specifically tailored to solve these problems for large-scale matrices and can achieve the accelerated convergence rate of $O(1/\sqrt{\Delta_k})$ being Δ_k the eigen-gap of $J_T(\boldsymbol{v}_k)$. Filtered power methods by Chebyshev polynomials [8,39] are also suited to provide a systematic acceleration over the basic power method. Numerical results show that the inner-outer schemes (9) complemented with Lanczos or filtered power iterations are successful attempts to accelerate the convergence of the SK algorithm in the case of clustered dominant eigenvalues of $J_T(\boldsymbol{x})$.

The paper is organized as follows. In Section 2 after briefly recalling the properties of the SK fixed point iteration (6) we exploit the eigenvalue connection by devising accelerated variants of (6) using Arnoldi-type methods. The description and implementation of these variants together with numerical results are discussed in Section 3. Finally, in section 4 conclusion and some remarks on future work are given.

2 Theoretical Setup

Let us denote by $\mathcal{P}, \mathcal{P}_0$ and $\mathcal{P}_\infty$ the subsets of $\bar{\mathbb{R}}^n$, $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$, defined by $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \geq \boldsymbol{0}\}$, $\mathcal{P}_0 = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} > \boldsymbol{0}\}$ and $\mathcal{P}_\infty = \{\boldsymbol{x} \in \bar{\mathbb{R}}^n : \boldsymbol{x} \geq \boldsymbol{0}\}$, respectively. For the sake of simplicity, we consider the matrix scaling problem (1) for a given $A \in \mathbb{R}^{n\times n}$ with all positive entries, that is, $A > 0$. Extensions to more general fully indecomposable nonnegative matrices can be obtained by using classical results in matrix theory possibly complemented with the perturbative analysis introduced in [26] (see also Section 6.2 in [22]). These extensions are briefly sketched in the following when required. Numerical evidences shown in Section 3 indicate that our approach also works in the more general setting.

Arithmetic operations are generalized as in [26] over the nonnegative extended real line $[0, +\infty] \subset \bar{\mathbb{R}}$ by setting $1/0 = \infty$, $1/\infty = 0$, $\infty + \infty = \infty$, $0 \cdot \infty = 0$, $a \cdot \infty = \infty$ if $a > 0$, where $\infty = +\infty$. Under these assumptions we can introduce the nonlinear operators defined as follows:

1. $U : \mathcal{P}_\infty \to \mathcal{P}_\infty$, $U(\boldsymbol{x}) = 1./\boldsymbol{x}$;
2. $S : \mathcal{P}_\infty \to \mathcal{P}_\infty$, $S(\boldsymbol{x}) = U(A\boldsymbol{x})$;
3. $T : \mathcal{P}_\infty \to \mathcal{P}_\infty$, $T(\boldsymbol{x}) = U(A^T U(A\boldsymbol{x}))$.

In this way it can be easily noticed that T is the same as the operator introduced in (6) and, therefore, the Sinkhorn-Knopp problem for the matrix A reduces to computing the fixed points of T, that is, the vectors $\boldsymbol{x} \in \mathcal{P}_\infty$ such that

$$\boldsymbol{x} = T(\boldsymbol{x}) = U(A^T U(A\boldsymbol{x})), \quad \boldsymbol{x} \in \mathcal{P}_\infty. \tag{10}$$

Summing up the results stated in [25, 26] we obtain the following theorem concerning the existence and the uniqueness of the desired fixed point.

Theorem 1 *Let $A \in \mathbb{R}^{n\times n}$ be a matrix with all positive entries. Then $\forall \boldsymbol{u} \in \mathcal{P} \setminus \{\boldsymbol{0}\}$ we have $\sup\{\lambda : T(\boldsymbol{u}) \geq \lambda \boldsymbol{u}\} \leq 1$. Moreover, T has a distinct eigenvalue equal to 1 with a unique (except for positive scalar multiples) corresponding eigenvector $\boldsymbol{x} \in \mathcal{P}_0$.*

The basic SK algorithm proceeds to approximate the eigenvector $\boldsymbol{x} \in \mathcal{P}_0$ by means of the fixed point iteration

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathcal{P}_0; \\ \boldsymbol{x}^{(k+1)} = T(\boldsymbol{x}^{(k)}), \quad k \geq 0. \end{cases} \tag{11}$$

Observe that $\boldsymbol{x}^{(0)} \in \mathcal{P}_0$ implies $\boldsymbol{x}^{(k)} \in \mathcal{P}_0$ $\forall k \geq 0$. The property is immediate under the assumption $A > 0$ but it still remains true if A is supposed to be nonnegative and fully indecomposable. Indeed, by Frobenius-König theorem [24] it follows that if $A \geq 0$ is fully indecomposable then there exist permutation matrices P and Q such that $B = PAQ$ is irreducible with positive diagonal. Clearly $\boldsymbol{x} \in \mathcal{P}_0$ implies $B\boldsymbol{x} \in \mathcal{P}_0$ and, hence, $T(\boldsymbol{x}) \in \mathcal{P}_0$.

The iteration (11) is shown to be globally convergent since T is a contraction for the Hilbert metric associated to the cone $\mathcal{P}$. [7, 23].

Theorem 2 *For any $\boldsymbol{x}^{(0)} \in \mathcal{P}_0$ there exists $\gamma = \gamma(\boldsymbol{x}^{(0)}) \in \mathbb{R}$, $\gamma > 0$, such that*

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \gamma \boldsymbol{x}.$$

The convergence is linear and the rate depends on the second singular value of the doubly stochastic matrix $\Sigma = \mathcal{D}(S(\boldsymbol{x}))A\mathcal{D}(\boldsymbol{x})$. We have the following [19].

Theorem 3 *Let $\boldsymbol{x} \in \mathcal{P}_0$ denote the limit of the sequence $\{\boldsymbol{x}^{(k)}\}_{k\in\mathbb{N}}$ generated according to (11). Then the matrix $\Sigma = \mathcal{D}(S(\boldsymbol{x}))A\mathcal{D}(\boldsymbol{x})$ is doubly stochastic and, moreover, if σ_2 is the second largest singular value of Σ it holds*

$$\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x} \|_2 \leq \sigma_2^2 \| \boldsymbol{x}^{(k)} - \boldsymbol{x} \|_2 + o\left(\| \boldsymbol{x}^{(k)} - \boldsymbol{x} \|_2\right), \quad k \geq 0.$$

The convergence can be very slow in the case of nearly decomposable matrices. The following definition is provided in [1, 28].

Definition 1 For a given $\epsilon > 0$, the matrix $A \in \mathbb{R}^{n\times n}$ is ϵ-nearly decomposable if there exist $E \in [0, 1]^{n\times n}$ and a permutation matrix P such that $PAP^T = \hat{A} + \epsilon E$ where $\hat{A}$ is block triangular with square diagonal blocks.

The relevance of nearly decomposable matrices for the study of dynamic systems in economics has been examined by Simon and Ando [34]. The role of near decomposability in queuing and computer system applications has been discussed in [10]. For a general overview of the properties of nearly decomposable graphs and networks with applications in data science and information retrieval one can see [30].

Example 1 For the matrix $A = \begin{bmatrix} 1 & \epsilon \\ 1 & 1 \end{bmatrix}$, $\epsilon > 0$, the SK iteration (11) is convergent but the number of iterations grows exponentially as ϵ becomes small. In Table 1 we show the number of iterations performed by Algorithm 1 in Section 3 applied to the matrix A with the error tolerance $\tau = 1.0e - 8$.

The local dynamics of (11) depend on the properties of the Jacobian matrix evaluated at the fixed point. By using the chain rule for the composite function we obtain that

$$J_T(\boldsymbol{z}) = J_{UA^TUA}(\boldsymbol{z}) = J_U(A^T U(A\boldsymbol{z})) \cdot J_{A^T}(U(A\boldsymbol{z})) \cdot J_U(A\boldsymbol{z}) \cdot J_A(\boldsymbol{z}).$$

Table 1: Number of SK iterations ItN for different values of $\epsilon = 10^{-k}$

k	1	2	3	4	5	6	7	8	9	10
ItN	16	46	132	391	1139	3312	9563	27360	77413	216017

where $J_A(\boldsymbol{z}) = A$ is the Jacobian matrix of the map on $\mathbb{R}^n$ induced by the matrix A, i.e., $\boldsymbol{z} \to A\boldsymbol{z}$. Since $J_U(\boldsymbol{z}) = -\mathcal{D}^{-2}(\boldsymbol{z}) = -\mathcal{D}^2(U(\boldsymbol{z}))$ we find that

$$J_T(\boldsymbol{z}) = \mathcal{D}^2(T(\boldsymbol{z})) \cdot A^T \cdot \mathcal{D}^2(S(\boldsymbol{z}))A. \tag{12}$$

The next result gives a lower bound for the spectral radius of $J_T(\boldsymbol{z})$ for $\boldsymbol{z} \in \mathcal{P}_0$.

Theorem 4 *For any given fixed $\boldsymbol{z} \in \mathcal{P}_0$ the spectral radius of $J_T(\boldsymbol{z})$ satisfies $\rho(J_T(\boldsymbol{z})) \geq 1$.*

Proof Let us denote $G = \mathcal{D}(T(\boldsymbol{z})) \cdot A^T \cdot \mathcal{D}(S(\boldsymbol{z}))$. It holds

$$\begin{aligned} J_T(\boldsymbol{z}) &= \mathcal{D}^2(T(\boldsymbol{z})) \cdot A^T \cdot \mathcal{D}^2(S(\boldsymbol{z}))A \\ &= \mathcal{D}(T(\boldsymbol{z})) \cdot G \cdot G^T \mathcal{D}^{-1}(T(\boldsymbol{z})), \end{aligned}$$

and, hence $J_T(\boldsymbol{z})$ and $G \cdot G^T$ are similar. Now observe that

$$G\boldsymbol{e} = \mathcal{D}(T(\boldsymbol{z})) \cdot A^T \cdot \mathcal{D}(S(\boldsymbol{z}))\boldsymbol{e} = \mathcal{D}(T(\boldsymbol{z})) \cdot A^T U A \boldsymbol{z} = \boldsymbol{e}.$$

It follows that $\rho(J_T(\boldsymbol{z})) = \sigma_1^2(G) = \| G \|_2^2 \geq 1$.

If $\boldsymbol{x} = T(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{P}_0$, then it is worth noting that

$$J_T(\boldsymbol{x}) = \mathcal{D}^2(T(\boldsymbol{x})) \cdot A^T \cdot \mathcal{D}^2(S(\boldsymbol{x}))A = \mathcal{D}^2(\boldsymbol{x}) \cdot A^T \cdot \mathcal{D}^2(S(\boldsymbol{x}))A,$$

and, hence,

$$J_T(\boldsymbol{x}) = \mathcal{D}(\boldsymbol{x}) \cdot \Sigma^T \Sigma \, \mathcal{D}^{-1}(\boldsymbol{x}),$$

where Σ is introduced in Theorem 3. This means that $J_T(\boldsymbol{x})$ and $F = \Sigma^T \Sigma$ are similar and therefore the eigenvalues of $J_T(\boldsymbol{x})$ are the squares of the singular values of Σ. Since $A > 0$ then it is irreducible and primitive and the same holds for Σ and a fortiori for F. These properties still hold if $A \geq 0$ is fully indecomposable. In this case it follows that Σ is fully indecomposable and therefore by Frobenius-König theorem there exist permutation matrices P and Q such that $B = P\Sigma Q$ is irreducible with positive diagonal. Then F is similar to BB^T which is a nonnegative irreducible and symmetric positive definite matrix. Hence, BB^T is primitive too. By the Perron-Frobenius theorem we conclude that the spectral radius of $J_T(\boldsymbol{x})$ satisfies $\rho(J_T(\boldsymbol{x})) = 1$ and $\lambda = 1$ is a simple eigenvalue of $J_T(\boldsymbol{x})$ with a positive corresponding eigenvector.

A characterization of such an eigenvector can be derived by the following result.

Theorem 5 *For each vector $\boldsymbol{z} \in \mathcal{P}_0$ it holds*

$$T(\boldsymbol{z}) = J_T(\boldsymbol{z}) \cdot \boldsymbol{z}.$$

Proof Let $\boldsymbol{z} \in \mathcal{P}_0$, then we have

$$\begin{aligned} J_T(\boldsymbol{z}) \cdot \boldsymbol{z} &= \mathcal{D}^2(T(\boldsymbol{z})) \cdot A^T \cdot \mathcal{D}^2(S(\boldsymbol{z}))A\boldsymbol{z} \\ &= \mathcal{D}^2(T(\boldsymbol{z})) \cdot A^T \cdot S(\boldsymbol{z}) \\ &= \mathcal{D}^{-2}(A^T S(\boldsymbol{z})) \cdot A^T \cdot S(\boldsymbol{z}) \\ &= \mathcal{D}^{-1}(A^T S(\boldsymbol{z}))\boldsymbol{e} \\ &= \mathcal{D}(U(A^T U(A\boldsymbol{z})))\boldsymbol{e} \\ &= T(z). \end{aligned}$$

This theorem implies that

$$\boldsymbol{x} \in \mathcal{P}_0,\ \boldsymbol{x} = T(\boldsymbol{x}) \quad \Longleftrightarrow \quad \boldsymbol{x} \in \mathcal{P}_0,\ \boldsymbol{x} = J_T(\boldsymbol{x})\boldsymbol{x}$$

and therefore the eigenvector of $J_T(\boldsymbol{x})$ corresponding with the eigenvalue 1 is exactly the desired solution of the SK problem. Furthermore, the SK iteration (11) can equivalently be written as

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathcal{P}_0; \\ \boldsymbol{x}^{(k+1)} = T(\boldsymbol{x}^{(k)}) = J_T(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)}, \quad k \geq 0. \end{cases} \tag{13}$$

According to Theorem 3 the convergence rate of (13) is $O(1/\Delta)$ where $\Delta = 1 - \lambda_2$ is the eigen-gap with $\lambda_2 = \sigma_2^2$ the second largest eigenvalue of $J_T(\boldsymbol{x})$ and $\mathcal{P}_0 \ni \boldsymbol{x} = \lim_{k\to+\infty} \boldsymbol{x}^{(k)}$. Notably, the power method, as well as some its inexact variants, applied to $J_T(\boldsymbol{x})$ inherits the same rate of convergence. Lanczos method and filtered power methods can achieve the accelerated rate $O(1/\sqrt{\Delta})$ for computing the top eigenpair of a symmetric positive semidefinite matrix [15, 39]. In the following we elaborate upon the relationship between (13) and the power method in order to devise similar accelerated modifications of the iterative scheme (13).

In principle one can accelerate the convergence of this scheme without improving the efficiency of the resulting method by replacing T with the operator $T_\ell = T \circ T \circ \cdots \circ T$, $T_1 = T$, generated from the composition (ℓ times) of T for a certain $\ell \geq 1$. The linearized form of the resulting iteration around the fixed point $\boldsymbol{x} = T(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{P}_0$, is

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathcal{P}_0; \\ \boldsymbol{x}^{(k+1)} = J_T(\boldsymbol{x})^\ell \boldsymbol{x}^{(k)}, \quad k \geq 0. \end{cases} \tag{14}$$

This is the power method applied to the matrix $J_T^\ell(\boldsymbol{x})$ for the approximation of an eigenvector associated with the dominant eigenvalue $\lambda = 1$. A normalized variant of (14) can be more suited for numerical computations

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathcal{P}_0; \\ \begin{cases} \boldsymbol{v}^{(k+1)} = J_T(\boldsymbol{x})^\ell \boldsymbol{x}^{(k)}, \\ \boldsymbol{x}^{(k+1)} = v^{(k+1)}/(\boldsymbol{e}^T \boldsymbol{v}^{(k+1)}) \end{cases}, \quad k \geq 0. \end{cases} \tag{15}$$

Since $\lambda = 1$ is the simple dominant eigenvalue of $J_T(\boldsymbol{x})$ with a positive corresponding eigenvector it is well known that (15) generates sequences such that

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}/(\boldsymbol{e}^T\boldsymbol{x}), \quad \limsup_{k\to\infty} \| \boldsymbol{x}^{(k)} - \boldsymbol{x}/(\boldsymbol{e}^T\boldsymbol{x}) \|_2^{1/k} \leq \lambda_2^\ell = \sigma_2^{2\ell},$$

where $0 \leq \lambda_2 = \sigma_2^2 < 1$ is the second largest eigenvalue of $J_T(\boldsymbol{x})$ and σ_2 denotes the second largest singular value of P defined as in Theorem 3. For practical purposes we introduce the following modified adaptation of (15) called SK_ℓ iteration:

$$\begin{cases} \boldsymbol{x}^{(0)} \in \mathcal{P}_0; \\ \begin{cases} \boldsymbol{v}^{(k+1)} = J_T(\boldsymbol{x}^{(k)})^\ell \boldsymbol{x}^{(k)}, \\ \boldsymbol{x}^{(k+1)} = v^{(k+1)}/(\boldsymbol{e}^T \boldsymbol{v}^{(k+1)}) \end{cases}, \quad k \geq 0. \end{cases} \tag{16}$$

For $\ell = 1$ SK_1 reduces to the scaled customary SK iteration. Under suitable assumptions we can show that SK_ℓ generates a sequence converging to the desired fixed point.

Theorem 6 *Let $\{\boldsymbol{x}^{(k)}\}_k$ be the sequence generated by SK_ℓ from a given initial guess $\boldsymbol{x}^{(0)} \in \mathcal{P}_0$. Let $\boldsymbol{x} \in \mathcal{P}_0$ be such that $\boldsymbol{x} = T(\boldsymbol{x})$ and $\boldsymbol{e}^T\boldsymbol{x} = 1$. Assume that:*

1. $\exists \eta > 0 \; : \; J_T(\boldsymbol{x}^{(k)})^\ell = J_T(\boldsymbol{x})^\ell + E_k, \; \| E_k \|_2 \leq \eta \sigma_2^{2\ell k}, \;\; k \geq 0;$
2. $\exists \gamma > 0 \; : \; \| \prod_{k=0}^m J_T(\boldsymbol{x}^{(k)})^\ell \|_2 \geq \gamma, \;\; m \geq 0.$

Then we have

$$\lim_{k \to \infty} \boldsymbol{x}^{(k)} = \boldsymbol{x},$$

and

$$\limsup_{k \to \infty} \| \boldsymbol{x}^{(k)} - \boldsymbol{x} \|_2^{1/k} \leq \sigma_2^{2\ell}.$$

Proof Since $\sum_{k=0}^\infty \| E_k \|_2 < \infty$ from Theorem 4.1 in [38] we obtain that the matrix sequence $P_m = \prod_{k=0}^m J_T(\boldsymbol{x}^{(k)})^\ell$ is such that

$$\lim_{m \to \infty} P_m = \boldsymbol{x}\boldsymbol{z}^T, \quad \boldsymbol{z} \in \mathcal{P}.$$

From Property 2 in view of the continuity of the norm it follows that $\boldsymbol{z} \neq \boldsymbol{0}$ and this implies the convergence of $\{\boldsymbol{x}^{(k)}\}_k$. About the rate of convergence we observe that

$$\begin{aligned} &\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x} \|_2 = \| \frac{P_k \boldsymbol{x}^{(0)}}{\boldsymbol{e}^T P_k \boldsymbol{x}^{(0)}} - \boldsymbol{x} \frac{\boldsymbol{z}^T \boldsymbol{x}^{(0)}}{\boldsymbol{z}^T \boldsymbol{x}^{(0)}} \|_2 \\ &\leq \| \frac{P_k \boldsymbol{x}^{(0)}}{\boldsymbol{z}^T \boldsymbol{x}^{(0)}} - \boldsymbol{x} \frac{\boldsymbol{z}^T \boldsymbol{x}^{(0)}}{\boldsymbol{z}^T \boldsymbol{x}^{(0)}} \|_2 + \| \frac{P_k \boldsymbol{x}^{(0)}}{\boldsymbol{e}^T P_k \boldsymbol{x}^{(0)}} - \frac{P_k \boldsymbol{x}^{(0)}}{\boldsymbol{z}^T \boldsymbol{x}^{(0)}} \|_2 \\ &\leq \frac{\| (P_k - \boldsymbol{x}\boldsymbol{z}^T)\boldsymbol{x}^{(0)}) \|_2}{|\boldsymbol{z}^T \boldsymbol{x}^{(0)}|} + \| P_k \boldsymbol{x}^{(0)} \|_2 \left| \frac{\boldsymbol{e}^T (\boldsymbol{x}\boldsymbol{z}^T - P_k) x^{(0)}}{\boldsymbol{e}^T P_k \boldsymbol{x}^{(0)} \boldsymbol{z}^T \boldsymbol{x}^{(0)}} \right| \end{aligned}$$

which says that $\boldsymbol{x}^{(k)}$ approaches $\boldsymbol{x}$ as fast as P_k tends to $\boldsymbol{x}\boldsymbol{z}^T$. Again using Theorem 4.1 in [38] under our assumptions there follows that

$$\limsup_{k \to \infty} \| P_k - \boldsymbol{x}\boldsymbol{z}^T \|_2^{1/k} \leq \sigma_2^{2\ell} = \lambda_2^\ell.$$

which concludes the proof.

This theorem says that the speed of convergence of SK_ℓ increases as ℓ increases. Also, notice that for any $\boldsymbol{z} \in \mathcal{P}_0$ the matrix $J_T(\boldsymbol{z})$ is primitive and irreducible and therefore by the Perron-Frobenius theorem its spectral radius is a dominant eigenvalue with a corresponding positive eigenvector. The proof is straightforward for $A > 0$. The generalization to the case where $A \geq 0$ is fully indecomposable proceeds by the same arguments stated after the proof of Theorem 4.

From Theorem 4 the dominant eigenvalue of $J_T(\boldsymbol{z})$ is greater than or equal to 1. It follows that for large ℓ the iterate $\boldsymbol{x}^{(k+1)}$ provides an approximation of the positive dominant eigenvector of $J_T(\boldsymbol{x}^{(k)})$. This fact suggests to consider SK_∞ as an effective method for approximating the limit vector $\boldsymbol{x}$. The method performs as an inner-outer procedure. In the inner phase given the current approximation $\boldsymbol{x}^{(k)}$ of $\boldsymbol{x}$ we apply the Power Method

$$\begin{cases} \boldsymbol{v}^{(0)} = \boldsymbol{x}^{(k)}; \\ \begin{cases} \boldsymbol{z}^{(k+1)} = J_T(\boldsymbol{x}^{(k)})\boldsymbol{v}^{(k)}, \\ \boldsymbol{v}^{(k+1)} = \boldsymbol{z}^{(k+1)}/(\boldsymbol{e}^T\boldsymbol{z}^{(k+1)}) \end{cases}, \quad k \geq 0 \end{cases} \tag{17}$$

until convergence to find the new approximation $\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^{(\hat{k}+1)}$. Numerically this latter vector solves

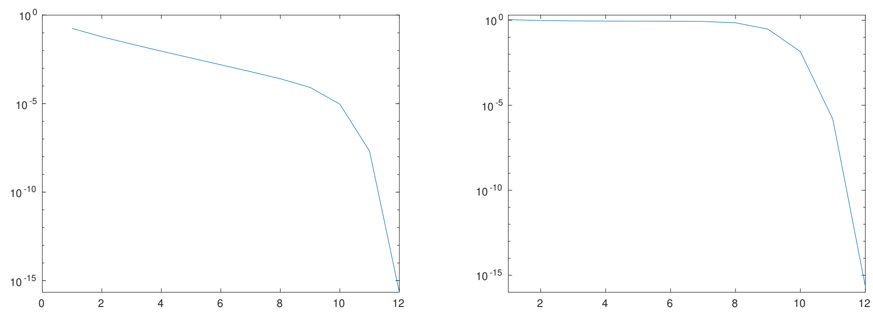
$$J_T(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k+1)} = \theta_k \boldsymbol{x}^{(k+1)},\ \theta_k = \rho(J_T(\boldsymbol{x}^{(k)})),\ \boldsymbol{e}^T\boldsymbol{x}^{(k+1)} = 1. \tag{18}$$

If convergence occurs, then $(\theta_k, \boldsymbol{x}^{(k+1)})$ approaches $(1, \boldsymbol{x})$ in the limit and convergence would be superlinear. Indeed, by setting $\theta_k = 1 + \delta_k$, $\delta_k \geq 0$, it follows that (18) can be recast as an inexact Newton method applied for the solution of the system of nonlinear equations $F(\boldsymbol{x}) = T(\boldsymbol{x}) - \boldsymbol{x} = \boldsymbol{0}$. In fact in view of Theorem 5 we obtain that

$$\begin{array}{l} F'(\boldsymbol{x}^{(k)})(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) = -F(\boldsymbol{x}^{(k)}) + \delta_k \boldsymbol{x}^{(k+1)} \iff \\ (J_T(\boldsymbol{x}^{(k)} - I_n)(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) = \boldsymbol{x}^{(k)} - T(\boldsymbol{x}^{(k)}) + \delta_k \boldsymbol{x}^{(k+1)} \iff \\ (J_T(\boldsymbol{x}^{(k)} - I_n)\boldsymbol{x}^{(k+1)} = \delta_k \boldsymbol{x}^{(k+1)} \iff \\ J_T(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k+1)} = (1 + \delta_k)\boldsymbol{x}^{(k+1)} = \theta_k \boldsymbol{x}^{(k+1)}. \end{array}$$

Example 2 As in Example 1 let $A = \begin{bmatrix} 1 & \epsilon \\ 1 & 1 \end{bmatrix}$ with $\epsilon = 1.0e-8$. In Figures 1a and 1b we illustrate the convergence history of iteration (18) applied to A with starting guess $\boldsymbol{x}^{(0)} = \boldsymbol{v}^{(0)} = [1/2, 1/2]^T$. The iterative scheme stops after 13 steps. The dominant eigenpair $(\lambda_k, \boldsymbol{v}^{(k+1)})$ is computed by using the function `eig` of MATLAB R2019a. In Figure 1b we show the distance between two consecutive normalized eigenvectors measured in the Hilbert metric $d_H(\boldsymbol{u}, \boldsymbol{v}) = \max_{i,j} \log\left(\frac{u_i v_j}{v_i u_j}\right)$, $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{P}_0$. It is seen that the convergence is regular and it looks ultimately superlinear.

In principle the matrix eigenvalue problem (18) can be solved by using any reliable method. For large sparse matrices in the case of clustered eigenvalues the convergence of the inner iteration can be greatly improved by considering filtered variants of the power method or Krylov-based algorithms for approximating a few largest eigenvalues of the matrix. In the next section the effectiveness and robustness of these methods are evaluated by numerical experiments.



(a) Figure 1a.   (b) Figure 1b

Fig. 1: Plot of $\psi(k) = \lambda_k - 1$ (Figure 1a) and of $\theta(k) = d_H(\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)})$ (Figure 1b) for the matrix A as given in Example 2 where $(\lambda_k, \boldsymbol{v}^{(k+1)})$ is the dominant eigenpair of $J_T(\boldsymbol{v}^{(k)})$.

3 Numerical Results

We have tested the algorithms presented above in a numerical environment using MATLAB R2019a on a PC with Intel Core i9-9900K processor with 64GB of RAM. The first method to be considered is the scaled SK iteration implemented by Algorithm 1.

Algorithm 1 Scaled SK iteration

Input: $A \in \mathbb{R}^{n\times n}$, $A \geq 0$ and a tolerance $\tau > 0$
Output: $\boldsymbol{x}$ such that $\boldsymbol{x} = T\boldsymbol{x}$, $\boldsymbol{x} \geq \boldsymbol{0}$, $\mathtt{sum}(\boldsymbol{x}) = 1$

Function SK(A, τ)
$\boldsymbol{x} = \mathtt{ones(n,1)/n}$;
$err = inf$;
while $err > \tau$ **do**
  $\boldsymbol{z} = T(\boldsymbol{x})$;
  $s = \mathtt{sum}(\mathbf{z})$;
  $z = z/s$;
  $err = \mathtt{norm}(\boldsymbol{z} - \boldsymbol{x})$;
  $\boldsymbol{x} = \boldsymbol{z}$;
EndFunction

Based on the results of the previous section we propose to exploit the properties of either power methods and Arnoldi-type iterations for computing the SK vector. The resulting schemes are specializations of Algorithm 2. Algorithm 2 makes use of an internal function $FDE(J_T(\boldsymbol{x}), \boldsymbol{x}, \tau)$ for "finding the dominant eigenpair" of $J_T(\boldsymbol{x})$ at a prescribed tolerance depending on the value of τ. If FDE implements the power method then Algorithm 2 reduces to the SK_∞ iterative method. However, when the largest eigenvalues of $J_T(\boldsymbol{x})$ are clustered the power method will perform poorly. In this case the performance of the eigensolver can be improved by exploiting different techniques and methodologies. In our experiments, we have considered three different approaches:

Algorithm 2 Arnoldi-type method

Input: $A \in \mathbb{R}^{n\times n}$, $A \geq 0$ and given tolerances $\tau > 0$
Output: $\boldsymbol{x}$ such that $\boldsymbol{x} = T\boldsymbol{x}$, $\boldsymbol{x} \geq \boldsymbol{0}$, $\texttt{sum}(\boldsymbol{x}) = 1$

Function Arnoldi_SK(A, τ)
$\boldsymbol{x} = \texttt{ones(n, 1)/n}$;
$err = inf$;
while $err > \tau$ **do**
  $[\lambda, \boldsymbol{z}] = FDE(J_T(\boldsymbol{x}), \boldsymbol{x}, \tau)$;
  $s = \texttt{sum}(\mathbf{z})$;
  $z = z/s$;
  $err = \texttt{norm}(\texttt{T}(\boldsymbol{z}) - \boldsymbol{z})$;
  $\boldsymbol{x} = \boldsymbol{z}$;
EndFunction

1. Filtered power methods using Chebyshev approximation to provide better separation of the eigenvalues of $J_T(\boldsymbol{x})$. A similar method was proposed in [39] for accelerating the HITS algorithm. The power method is applied to the matrix $\tilde{c}_m(J_T(\boldsymbol{x}))$ where $\tilde{c}_m(z) = c_m(2z-1)$ with $c_m(z)$ the $m-$th degree Chebyshev polynomial of the first kind. When $\lambda_1 > 1 > \lambda_2$ and λ_1, λ_2 are the top eigenvalues of $J_T(\boldsymbol{x})$ then $|\tilde{c}_m(\lambda_2)/\tilde{c}_m(\lambda_1)|$ is significantly smaller than $|\lambda_2/\lambda_1|$. The resulting adaptations of Algorithm 2 are denoted as Alg2_FCPM$_m$ where m is the degree of the Chebyshev polynomial.
2. The Lanczos iteration can implicitly use the properties of Chebyshev approximation by achieving an accelerated convergence rate. We compute an approximation of the dominant eigenpair of $J_T(\boldsymbol{x})$ by using the MATLAB function `eigs` which implements an implicitly restarted Arnoldi/Lanczos method. The input sequence of `eigs` is given as

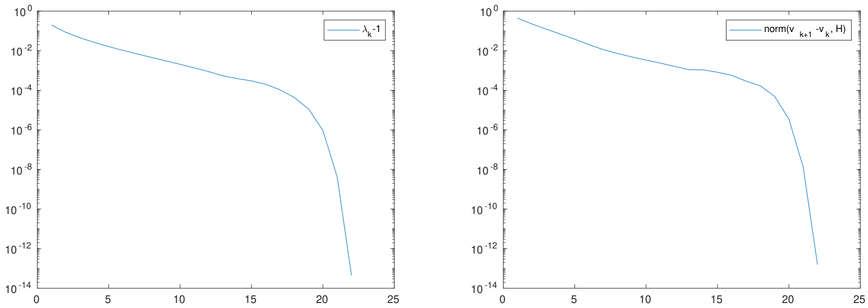
```
[V,D]=eigs(@(w)D2*D2*AFUNT(D1*D1*AFUN(w)),length(A),1,'largestabs','StartVector',x);
```

   where $AFUN(\boldsymbol{w})$ and $AFUNT(\boldsymbol{w})$ are functions that compute the product $A\boldsymbol{w}$ and $A^T\boldsymbol{w}$, respectively, $D1 = \mathcal{D}(S(\boldsymbol{x}))$ and $D2 = \mathcal{D}(T(\boldsymbol{x}))$ are diagonal matrices and A, $D1$ and $D2$ are stored in a sparse format. This modification of Algorithm 2 is referred as Alg2_EIGS
3. A further improvement of the Lanczos iteration are the block variants whose convergence depend on the separation between those eigenvalues that are "close" to λ_1 and those that are sufficiently smaller in magnitude. In our experiments we consider the function `ahbeigs` [2] which implements a block Arnoldi/Lanczos method for computing a few eigenvalues of sparse matrices. Block methods can suffer from the occurrence of complex eigenpairs. Therefore, based on the proof of Theorem 4 the method is applied to the symmetric matrix $G \cdot G^T$, $G = D_2 \cdot A^T \cdot D_1$ which is similar to $J_T(\boldsymbol{x})$. The input sequence of `ahbeigs` is given as

```
OPTS.sigma='LM';OPTS.k=m;OPTS.V0=R0;[V,D]=ahbeigs('afuncsym', n, speye(n),  OPTS)
```

   where n is the size of the matrix A, m is the number of desired eigenvalues, $R0 \in \mathbb{R}^{n\times m}$ is the set of starting vectors and `'afuncsym'` denotes a function that computes the product of $G \cdot G^T$ by a vector where A is stored in a sparse format. The resulting variation of Algorithm 2 is named Alg2_AHBEIGS.

It is worth pointing out that MATLAB complies with the IEEE 754 standard for floating point arithmetic, including in its treatment of infinite values, and



(a) Figure 2a. (b) Figure 2b

Fig. 2: Plot of $\psi(k) = \lambda_k - 1$ (Figure 2a) and of $\theta(k) = d_H(\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)})$ (Figure 2b) for the matrix $H_{128}(127)$ where $(\lambda_k, \boldsymbol{v}^{(k+1)})$ is the dominant eigenpair of $J_T(\boldsymbol{v}^{(k)})$.

therefore, differently from the convention assumed at the beginning of Section 2, we find that $0 \star (+Inf) = NaN$. In some exceptional cases this discrepancy can produce numerical difficulties and wrong results. Nevertheless, we have preferred to avoid the redefinition of the multiplication operation by presenting tests that are unaffected by such issue.

Our first batch of test matrices were introduced in [32] to compare the performances of different algorithms for matrix balancing. These are upper Hessenberg matrices $H_n(\gamma) \in \mathbb{R}^{n\times n}$ defined as $H_n(\gamma) = \widehat{H}_N + \gamma I_n$ with $\widehat{H}_N = (\hat{h}_{i,j})$ and $\hat{h}_{i,j} = 1$ if $j - i \geq -1$. For large values of γ and n the matrix becomes very close in a relative sense to a decompasable matrix. Our timing results for $n = 128$ and $\gamma = n - 1$ are given in Table 2. The algorithms considered here are our variants of Algorithm 2; the SK iteration implemented in Algorithm 1; EQ algorithm from [32] and BNEWT algorithm from [20]. In our tests the algorithm EQ from [32] performs slower than the SK iteration. This is in accordance with the comparisons shown in [20]. The adaptations of Algorithm 2 are stopped when $err = d_H(T(\boldsymbol{z}), \boldsymbol{z})$ is below a fixed tolerance of $\tau = 1.0e-12$. The stopping criterion in BNEWT is adjusted to obtain a comparable accuracy. It is seen that our proposed methods are faster than the original SK iteration but BNEWT outperforms the other choices. However, the point to stress here is that BNEWT is weakly connected to SK. As noticed in [20] the convergence of BNEWT is far from monotonic and the vector $\boldsymbol{x}$ returned as output is very large in norm ($\| \boldsymbol{x} \|_1 \simeq 1.0e + 18$). Conversely our algorithms retain many properties of SK. In Figure 2 we show the convergence graphs of Alg2_EIGS for $H_{128}(127)$. The method exhibits a monotonic convergence which is ultimately superlinear.

Table 2: Computing times (in seconds) of different algorithms applied to $H_{128}(127)$

Alg1	Alg2_FCPM$_3$	Alg2_FCPM$_6$	Alg2_EIGS	BNEWT
3.4	1.9	0.6	0.08	0.03

The performances of our proposed algorithms have been also evaluated and compared with SK on large sparse matrices either fully indecompasable or not. The test suite consists of the following matrices with entries 0 or 1 only:

i HB/can_1072 of size $n = 1072$ from the Harwell-Boeing collection;
ii SNAP/email-Eu-core of size $n = 1005$ from the SNAP (Stanford Network Analysis Platform) large network dataset collection;
iii SNAP/Oregon-1 of size $n = 11492$ from the SNAP collection;

The matrix HB/can_1072 is sparse and irreducible. The (scaled) SK iteration is convergent. Both Algorithm 1 and our variants of Algorithm 2 – Alg2_FCPM$_3$, Alg2_FCPM$_6$ and Alg2_EIGS– perform quite well with running times in the range between 0.02 and 0.06 seconds for a tolerance $\tau = 1.0e - 14$. More specifically Alg2_EIGS converges in 4 outer iterations. Table 3 gives the measured errors by showing quadratic convergence.

Table 3: Errors generated by Alg2_EIGS applied to HB/can_1072 for $\tau = 1.0e-14$

it	1	2	3	4
err	1.086e-02	1.678e-04	1.743e-08	2.664e-15

The remaining matrices (ii) and (iii) from the SNAP collections are not fully indecomposable. According to [19] in order to compute an approximate solution of the matrix balancing problem we may consider perturbations of the input matrix A of the form

$$\widetilde{A} = A + \gamma \boldsymbol{e}\boldsymbol{e}^T, \quad \boldsymbol{e} = [1, \dots, 1]^T, \quad \gamma > 0,$$

for decreasing values γ_i, $1 \le i \le K$, of γ. The approach resembles the customary strategy employed for solving the PageRanking problem. As γ approaches zero the associated eigenproblem becomes more and more challenging due to the occurrence of clustered eigenvalues around 1 of $J_T(\boldsymbol{x})$ where $\boldsymbol{x} \in \mathcal{P}_0$ is a fixed point of T. From Theorem 4.3 in [5] we know that Arnoldi-type methods can greatly benefit of the choice of the initial vector and therefore Alg2_EIGS is especially suited to be used in these continuation schemes.

In the next tables 4 and 5 we report the computing times of Algorithm 1 and Alg2_EIGS. When $\gamma = \gamma_1$ both algorithms start with $\boldsymbol{x} = \boldsymbol{e}/n$ whereas for $i > 1$ the starting vector is given by the solution computed at the previous step with $\gamma = \gamma_{i-1}$. In all experiments the tolerance was set at $\tau = 1.0e - 12$. We observe that Alg2_EIGS outperforms Algorithm 1 for sufficiently small values of γ when the perturbed matrix is close to the original web link graph.

Table 4: Computing times of Algorithm 1 and Alg2_EIGS applied to SNAP/email-Eu-core for different values of γ

γ	1.0e-2	1.0e-4	1.0e-6	1.0e-8	1.0e-10	1.0e-12	1.0e-14
Alg1	0.003	0.006	0.04	0.34	2.78	23.21	193.66
Alg2_EIGS	0.02	0.02	0.05	0.12	0.33	0.72	1.12

Table 5: Computing times of Algorithm 1 and Alg2_EIGS applied to SNAP/Oregon-1 for different values of γ

γ	1.0e-2	1.0e-4	1.0e-6	1.0e-8	1.0e-10	1.0e-12	1.0e-14
Alg1	0.005	0.01	0.05	0.32	2.21	13.9	61.56
Alg2_EIGS	0.04	0.06	0.21	0.63	2.05	6.04	11.67

In network analysis, an interesting class of difficult matrix scaling problems consists of adjacency matrices which exhibit community structures, that is, densely connected groups that are loosely associated with each other. In this case the number of eigenvalues clustered around the Perron root $\lambda_1 = 1$ can be related with the number of communities [27]. If we know some a priori upper bound on this number it can be recommended the use of a block Arnoldi-based eigensolver which using a set of starting vectors is able to compute multiple or clustered eigenvalues more efficiently than an unblocked routine. In our experiments we consider the function `ahbeigs` [2] which implements a block Arnoldi/Lanczos method for computing a few eigenvalues of sparse matrices. For numerical testing we consider the following matrices:

1. the adjacency matrix $A_{jazz} \in \mathbb{R}^{198\times 198}$ constructed from a collaboration network between Jazz musicians. Each node is a Jazz musician and an edge denotes that two musicians have played together in a band. The data was collected in 2003 [13].
2. the matrix $A_{mbeause} \in \mathbb{R}^{496\times 496}$ generated by taking the absolute value of the matrix HB/mbeause from the the Harwell-Boeing collection. The original matrix is derived from an economic model which reveals several communities. This structure is maintained in the modified matrix.

In Table 6 we compare the computing times of Algorithm 1 and Alg2_AHBEIGS applied to the matrices $\widetilde{A}\colon = A + \gamma \boldsymbol{e}\boldsymbol{e}^T$ for different values of γ and $A = A_{jazz}$, $A = A_{mbeause}$. In each experiment we set $\tau = 1.0e - 14$ and the (block) starting vector is $X = \texttt{ones}(n, m)$ where n is the size of A, $m = 1$ for Algorithm 1 and $m = 16, 32$ for Algorithm 2 applied to A_{jazz} and $A_{mbeause}$, respectively.

Table 6: Computing times of Algorithm 1 and Alg2_AHBEIGS for different values of γ

	A_{jazz}			$A_{mbeause}$		
γ	1.0e-10	1.0e-12	1.0e-14	1.0e-10	1.0e-12	1.0e-14
Alg1	0.19	0.77	3.51	9.06	44.91	118.18
Alg2_AHBEIGS	0.19	0.21	0.23	1.83	2.31	2.81

For these matrices the methods based on eigenvalue computations can be dramatically faster than the fixed point iteration. In particular, Alg2_AHBEIGS applied to A_{jazz} with $\tau = 1.0 - e - 14$ and $m = 16$ converges in 12,14,16 iterations for $\gamma = 1.0e - 10, 1.0e - 12, 1.0e - 14$, respectively.

4 Conclusions and Future Work

In this paper we have discussed some numerical techniques for accelerating the customary SK iteration based on certain equivalent formulations of the fixed point problem as a matrix eigenvalue problem. Variants of the power method relying upon the Arnoldi/Lanczos process have been proposed for the efficient solution of the matrix eigenvalue problem. There are several topics which remain to be addressed. Specifically:

1. A formal proof of the convergence for the SK_∞ method is still missing. As suggested by Figure 1b and Figure 2b in this respect it might be useful to investigate the contraction properties of the map $E\colon \mathcal{P}_0 \to \mathcal{P}_0$ defined by $E(\boldsymbol{v}) = \boldsymbol{w}$ where $\boldsymbol{w}$ is the normalized dominant eigenvector of $J_T(\boldsymbol{v})$.
2. The numerical behavior of (block) Arnoldi-based methods can be improved by exploiting several additional properties In particular following [14] we can take advantage of knowing the largest eigenvalue of the limit problem to speed up the intermediate steps. Also, when the matrix is symmetric then simplifications are introduced in the Jacobian matrix which can alleviate the numerical difficulties of the associated eigenvalue problem. Finally, the invariance by scaling of the solution of the matrix equilibration problem can be used to balance numerical computations.
3. Further theoretical and computational investigation of the modified algorithms for the generalized matrix balancing problem is an ongoing research project.

References

1. A. Ando and F.M. Fisher. Near-decomposability, partition and aggregation, and the relevance of stability discussions. *International Economic Review*, 4(1):53–67, 1963.
2. J. Baglama. Augmented block Householder Arnoldi method. *Linear Algebra Appl.*, 429(10):2315–2334, 2008.
3. Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the solution of algebraic eigenvalue problems*, volume 11 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. A practical guide.
4. Z. Bai, D. Lu, and B. Vandereycken. Robust Rayleigh quotient minimization and nonlinear eigenvalue problems. *SIAM J. Sci. Comput.*, 40(5):A3495–A3522, 2018.
5. M. Bellalij, Y. Saad, and H. Sadok. Further analysis of the Arnoldi process for eigenvalue problems. *SIAM J. Numer. Anal.*, 48(2):393–407, 2010.
6. E. Bozzo and M. Franceschet. A theory on power in networks. *Commun. ACM*, 59(11):75–83, October 2016.
7. R. A. Brualdi, S. V. Parter, and H. Schneider. The diagonal equivalence of a nonnegative matrix to a stochastic matrix. *J. Math. Anal. Appl.*, 16:31–50, 1966.
8. De Sa C., He B., Mitliagkas I., Ré C., and Xu P. Accelerated stochastic power iteration. *Proc Mach. Learn. Res.*, 84:58–67, 2018.
9. Y. Cai, L. H. Zhang, Z. Bai, and R. C. Li. On an eigenvector-dependent nonlinear eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 39(3):1360–1382, 2018.
10. P.-J. Courtois. *Decomposability*. Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1977. Queueing and computer system applications, ACM Monograph Series.
11. M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.
12. S. Diamond and S. Boyd. Stochastic matrix-free equilibration. *J. Optim. Theory Appl.*, 172(2):436–454, 2017.
13. P.M Gleiser and L. Danon. Community structure in jazz. *Advances in Complex Systems*, 6(4), 2003.

14. G. H. Golub and C. Greif. An Arnoldi-type algorithm for computing PageRank. *BIT*, 46(4):759–771, 2006.
15. G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.
16. M. Idel. A review of matrix scaling and Sinkhorn's normal form for matrices and positive maps. Technical report, arXiv:1609.06349, 2016.
17. E. Jarlebring, S. Kvaal, and W. Michiels. An inverse iteration method for eigenvalue problems with eigenvector nonlinearities. *SIAM J. Sci. Comput.*, 36(4):A1978–A2001, 2014.
18. B. Kalantari, L. Khachiyan, and A. Shokoufandeh. On the complexity of matrix balancing. *SIAM J. Matrix Anal. Appl.*, 18(2):450–463, 1997.
19. P. A. Knight. The Sinkhorn-Knopp algorithm: convergence and applications. *SIAM J. Matrix Anal. Appl.*, 30(1):261–275, 2008.
20. P. A. Knight and D. Ruiz. A fast algorithm for matrix balancing. *IMA J. Numer. Anal.*, 33(3):1029–1047, 2013.
21. B. Lamond and N. F. Stewart. Bregman's balancing method. *Transportation Res. Part B*, 15(4):239–248, 1981.
22. B. Lemmens and R. Nussbaum. *Nonlinear Perron-Frobenius theory*, volume 189 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2012.
23. B. Lemmens and R. Nussbaum. *Nonlinear Perron-Frobenius theory*, volume 189 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2012.
24. M. Marcus and H. Minc. *A survey of matrix theory and matrix inequalities*. Dover Publications, Inc., New York, 1992. Reprint of the 1969 edition.
25. M. V. Menon. Some spectral properties of an operator associated with a pair of nonnegative matrices. *Trans. Amer. Math. Soc.*, 132:369–375, 1968.
26. M. V. Menon and Hans Schneider. The spectrum of a nonlinear operator associated with a matrix. *Linear Algebra and Appl.*, 2:321–334, 1969.
27. C. D. Meyer and C. D. Wessell. Stochastic data clustering. *SIAM J. Matrix Anal. Appl.*, 33(4):1214–1236, 2012.
28. H. Minc. Nearly decomposable matrices. *Linear Algebra and Appl.*, 5:181–187, 1972.
29. M. Morishima. Generalizations of the Frobenius-Wielandt theorems for non-negative square matrices. *J. London Math. Soc.*, 36:211–220, 1961.
30. Αθανάσιος Ν Νικολακόπουλος. *Ranking under near decomposability*. PhD thesis, Department of Computer Engineering and Informatics, University of Patras, 2016.
31. A. Parikh. Forecasts of input-output matrices using the r.a.s. method. *The Review of Economics and Statistics*, 61(3):477–481, 1979.
32. B. N. Parlett and T. L. Landis. Methods for scaling to doubly stochastic form. *Linear Algebra Appl.*, 48:53–79, 1982.
33. L. Rüschendorf. Convergence of the iterative proportional fitting procedure. *Ann. Statist.*, 23(4):1160–1174, 1995.
34. H.A. Simon and A. Ando. Aggregation of variables in dynamic systems. *Econometrica*, 29(2):111–138, 1961.
35. R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879, 1964.
36. R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *Amer. Math. Monthly*, 74:402–405, 1967.
37. R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.*, 21:343–348, 1967.
38. G. W. Stewart. On the powers of a matrix with perturbations. *Numer. Math.*, 96(2):363–376, 2003.
39. Y. Zhou. Practical acceleration for computing the HITS ExpertRank vectors. *J. Comput. Appl. Math.*, 236(17):4398–4409, 2012.