

---

This is an electronic reprint of the original article.  
This reprint may differ from the original in pagination and typographic detail.

Dopico, Froilán M.; Quintana, María C.; van Dooren, Paul  
**Diagonal Scalings for the Eigenstructure of Arbitrary Pencils**

*Published in:*  
SIAM Journal on Matrix Analysis and Applications

*DOI:*  
[10.1137/20M1364011](https://doi.org/10.1137/20M1364011)

Published: 01/01/2022

*Document Version*  
Publisher's PDF, also known as Version of record

*Please cite the original version:*  
Dopico, F. M., Quintana, M. C., & van Dooren, P. (2022). Diagonal Scalings for the Eigenstructure of Arbitrary Pencils. *SIAM Journal on Matrix Analysis and Applications*, 43(3), 1213-1237.  
<https://doi.org/10.1137/20M1364011>

---

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

## DIAGONAL SCALINGS FOR THE EIGENSTRUCTURE OF ARBITRARY PENCILS\*

FROILÁN M. DOPICO<sup>†</sup>, MARÍA C. QUINTANA<sup>‡</sup>, AND PAUL VAN DOOREN<sup>§</sup>

**Abstract.** In this paper we show how to construct diagonal scalings for arbitrary matrix pencils  $\lambda B - A$ , in which both  $A$  and  $B$  are complex matrices (square or nonsquare). The goal of such diagonal scalings is to “balance” in some sense the row and column norms of the pencil. We see that the problem of scaling a matrix pencil is equivalent to the problem of scaling the row and column sums of a particular nonnegative matrix. However, it is known that there exist square and nonsquare nonnegative matrices that cannot be scaled arbitrarily. To address this issue, we consider an approximate embedded problem, in which the corresponding nonnegative matrix is square and can always be scaled. The new scaling methods are then based on the Sinkhorn–Knopp algorithm for scaling a square nonnegative matrix with total support to be doubly stochastic or on a variant of it. In addition, using results of U. G. Rothblum and H. Schneider [*Linear Algebra Appl.*, 114–115 (1989), pp. 737–764], we give simple sufficient conditions on the zero pattern for the existence of diagonal scalings of square nonnegative matrices to have any prescribed common vector for the row and column sums. We illustrate numerically that the new scaling techniques for pencils improve the accuracy of the computation of their eigenvalues.

**Key words.** pencils, accuracy of computed eigenvalues, diagonal scaling, Sinkhorn–Knopp algorithm

**MSC codes.** 15A18, 15A22, 65F15, 65F35

**DOI.** 10.1137/20M1364011

**1. Introduction.** The problem of scaling an entrywise nonnegative  $m \times n$  matrix  $A$  with diagonal transformations and prespecified vectors  $r$  and  $c$  for the row and column sums, respectively, consists of finding a matrix of the form  $S = D_\ell A D_r$ , where  $D_\ell \in \mathbb{R}^{m \times m}$  and  $D_r \in \mathbb{R}^{n \times n}$  are diagonal matrices having positive diagonal elements, and such that

$$(1.1) \quad S \mathbf{1}_n = r \quad \text{and} \quad \mathbf{1}_m^T S = c^T,$$

where  $\mathbf{1}_i := [1, \dots, 1]^T \in \mathbb{R}^i$  for  $i = n, m$  [3, 18]. When  $r = \mathbf{1}_m$  and  $c = \mathbf{1}_n$  the scaled matrix  $S$  is necessarily square and is said to be doubly stochastic; i.e., its row and column sums are all equal to 1.

---

\*Received by the editors September 2, 2020; accepted for publication (in revised form) May 5, 2022; published electronically July 18, 2022.

<https://doi.org/10.1137/20M1364011>

**Funding:** This publication is part of the “Proyecto de I+D+i PID2019-106362GB-I00 financiado por MCIN/AEI/10.13039/501100011033.” The work of the authors was supported by “Ministerio de Economía, Industria y Competitividad (MINECO)” of Spain and “Fondo Europeo de Desarrollo Regional (FEDER)” of the EU through grants MTM2015-65798-P and MTM2017-90682-REDT and by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M23) in the V PRICIT (Regional Programme of Research and Technological Innovation). The work of the second author was supported by the “contrato predoctoral” BES-2016-076744 of MINECO and by an Academy of Finland grant (Suomen Akatemian päätös 331240). This work was partially developed while the third author held a “Chair of Excellence UC3M - Banco de Santander” at Universidad Carlos III de Madrid in the academic year 2019–2020.

<sup>†</sup>Departamento de Matemáticas, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Spain (dopico@math.uc3m.es).

<sup>‡</sup>Department of Mathematics and Systems Analysis, Aalto University, Otakaari 1, Espoo, Finland (maria.quintanaponce@aalto.fi).

<sup>§</sup>Department of Mathematical Engineering, Université catholique de Louvain, Avenue Georges Lemaître 4, B-1348 Louvain-la-Neuve, Belgium (paul.vandooren@uclouvain.be).

The related problem of scaling the rows and columns of a complex square matrix  $A$  (not necessarily nonnegative) using real and positive diagonal similarity transformations in order to compute more accurate eigenvalues is a well-established technique to improve the sensitivity of the eigenvalue problem of the matrix  $A$  [16]. This is known as *balancing* the matrix  $A$ . In exact arithmetic, it amounts to minimizing the Frobenius norm of the scaled matrix  $D^{-1}AD$ , where  $D$  ranges over all nonsingular real diagonal matrices, which is equivalent to minimizing the departure from normality of  $D^{-1}AD$  [12]. Since the eigenvalues of normal matrices have condition numbers equal to 1, such scaling very often improves the sensitivity of eigenvalues. The method for computing the optimal scaling is a very simple cyclic procedure where at each step only a single diagonal element of  $D$  is updated. This method is implemented in MATLAB [21] as a default option of the eigenvalue computation problem, which indicates that its effectiveness is well accepted. For improving the accuracy of the eigenvalues computed in floating point arithmetic, it is essential that the diagonal elements of  $D$  are integer powers of 2 because in this way the scaling does not produce any rounding errors and the eigenvalues are preserved *exactly* under such a scaling transformation. Otherwise, the rounding errors inherent to constructing  $D^{-1}AD$  would spoil any potential improvement in the accuracy of the computed eigenvalues. As explained in [16], the restriction to diagonal matrices  $D$  whose entries are integer powers of 2 allows for a relaxed stopping criterion of the cyclic procedure for computing  $D$  and implies that the related minimization problem is only approximately solved.

The idea of performing positive diagonal scalings in order to improve the accuracy of computed eigenvalues was also extended to the generalized eigenvalue problem of a regular pencil  $\lambda B - A$ . In this case, the nonsingular diagonal matrices multiplying the pencil on the left and on the right are different. In [23], Ward describes a scaling technique which aims at making the pencil entries have magnitudes as close to unity as possible. In [12], Lemonnier and Van Dooren propose a diagonal scaling that in exact arithmetic minimizes the Frobenius norm of the pencils over all positive diagonal scalings with fixed determinant. This improves very often the conditioning of the eigenvalues since the solution of such a minimization problem over general nonsingular transformations is a so-called standardized normal pencil, which is a pencil whose eigenvalues all have a condition number in the chordal metric that is smaller than or equal to  $\sqrt{2}$ . The method of Ward is the one that LAPACK [1] proposes as a built-in option for scaling a regular pencil, but it was pointed out in [12] that the method of Lemonnier–Van Dooren outperforms that of Ward in terms of the accuracy of the computed eigenvalues, especially when the pencil has entries of strongly varying magnitudes. The experiments in section 6 will further confirm the superiority of the method in [12] for a wide variety of pencils of different sizes and types. As in the case of balancing matrices, it is essential that the entries of the diagonal scaling matrices are integer powers of 2 in order to improve the accuracy of the computed eigenvalues in floating point arithmetic. Currently, MATLAB does not offer any built-in option for scaling pencils. We will see in section 3 that the method in [12] is equivalent to scaling a particular nonnegative matrix to a multiple of a doubly stochastic matrix, which motivates us to revise briefly the literature on this and other related problems.

There is a vast literature on diagonal scaling of nonnegative matrices for getting a matrix with prescribed row and column sums. The origin of these problems goes back at least until the beginning of the 20th century [10, 25] and originates in the area of optimal transport [17], though it has applications in many other areas [7]. See [7, section 3.1] and [17, Remark 4.5] for historical remarks on these problems. Relevant classical references from the point of view of matrix analysis include [3, 11, 18, 19],

among many others. Despite this vast literature, several issues are still open for improvement, such as a good understanding of the convergence of related algorithms for sparse matrices and simple conditions on the zero pattern of the matrix for existence and unicity of a solution for special cases, especially in the case of rectangular matrices [7]. The most relevant papers on diagonal scalings that are related to this paper are, in chronological order, those of Sinkhorn and Knopp [19], Krupp [11], Rothblum and Schneider [18], and Knight [8], which is why we quote theorems from those papers.

In this paper we show that there exists a link between the problem of scaling a regular square pencil and that of scaling a square nonnegative matrix to become doubly stochastic. This implies that the scaling is essentially unique and bounded if and only if the corresponding nonnegative matrix satisfies certain conditions, namely, *total support* and *full indecomposability*. Moreover, in that situation, the scaling can be found through the well-known Sinkhorn–Knopp algorithm [8, 19]. We then show how to extend this to singular or nonsquare pencils, which, to the best of our knowledge, has not been considered yet in the literature. For that, we introduce a regularization term into the original problem which ensures existence of a solution of an approximate problem with bounded diagonal scalings  $D_\ell$  and  $D_r$ . In addition, the regularization term can be considered in both square or nonsquare cases.

These ideas are connected to the results in [18] about scaling arbitrary nonnegative matrices (square or rectangular) with prespecified row and column sums, which can be obtained using a Sinkhorn–Knopp–like algorithm, but many other optimization methods have been proposed in the literature [7, 17]. We build on these ideas to further improve the technique of Lemonnier and Van Dooren by introducing the regularization term as an additional cost. This regularization ensures the existence and boundedness of our scaling, but it also ensures essential unicity of the scaling.

The paper is organized as follows: in section 2, we give some basic notions about scaling pencils, scaling nonnegative matrices, and the Sinkhorn–Knopp–like algorithm. In sections 3 and 4, we study the diagonal scaling problem for square and nonsquare pencils, respectively. In section 3, we will also recall the necessary and sufficient conditions for a square nonnegative matrix to become doubly stochastic under diagonal scalings, and we give simple sufficient conditions based on the zero pattern of the matrix for the existence of diagonal scalings having any prespecified common vector for the row and column sums. These results will be useful in section 5. In that section, we develop a new scaling technique for generalized eigenvalue problems and show that it can be applied to any pencil, regular or singular, square or rectangular. For that, we introduce a regularization term into the original problem which guarantees existence, unicity, and boundedness of the scaling. In addition, in subsection 5.1, we consider a modified version of the new scaling technique that is often better for scaling nonsquare pencils. In section 6 we then illustrate the improved accuracy of the computed eigenvalues using several numerical examples. More numerical examples can be found in the accompanying supplementary file `supp_materials.pdf` [local/web 354KB], linked from the main article webpage. In particular, there we describe a real-world example where the proposed scaling techniques allow us to get a dramatic improvement in the conditioning of the resulting generalized eigenvalue problem. In the last section 7 we give some concluding remarks.

## 2. Preliminaries: Scaling arbitrary pencils and nonnegative matrices.

The standard techniques for computing eigenvalues of complex pencils of matrices guarantee that the backward errors corresponding to the computed spectrum are essentially bounded by the norm of the coefficients of the pencil, multiplied by the

machine precision of the computer used. But one can improve this bound by reducing the norms of the coefficients without affecting the spectrum. This is where balancing using diagonal scaling comes in. We emphasize that the diagonal entries of such scalings must be integer powers of 2 in order to avoid rounding errors that would destroy any potential improvement in accuracy that such scalings might achieve.

Two types of scalings can be applied to a complex pencil  $\lambda B - A$ .

The first one is a change of variable  $\hat{\lambda} := d_\lambda \lambda$  to make sure that the scaled matrices  $A$  and  $B/d_\lambda$  have approximately the same norm. This can be done without introducing rounding errors by taking  $d_\lambda$  equal to an integer power of 2. The staircase [22] and the  $QZ$  [15] algorithms work independently on both matrices, and this scaling can be restored afterwards, again without introducing any additional errors. One could therefore argue that this scaling is irrelevant for these algorithms, but we will see that it affects the second scaling procedure we will discuss. Therefore we will assume in what follows that both matrices  $A$  and  $B$  are of comparable norms and that no such variable scaling needs to be applied.

The second scaling is based on multiplication on the left and on the right by positive diagonal matrices  $D_\ell$  and  $D_r$ , respectively, that are chosen to “balance” in some sense the row and column norms of the complex matrices  $\tilde{A} := D_\ell A D_r$  and  $\tilde{B} := D_\ell B D_r$ . We will see that balancing the row and column norms of  $\tilde{A}$  and  $\tilde{B}$  is equivalent to performing two-sided diagonal scalings to a particular real entrywise nonnegative matrix  $M$ . Therefore, we recall in what follows some results on this problem.

The first result we revise appears in [18, Theorem 2(a)–(b)] and is the next one.

**THEOREM 2.1.** *Given a real nonnegative matrix  $M \in \mathbb{R}^{m \times n}$  and vectors  $r \in \mathbb{R}^{m \times 1}$  and  $c \in \mathbb{R}^{n \times 1}$  with strictly positive entries satisfying  $\mathbf{1}_m^T r = c^T \mathbf{1}_n$ , there exist positive diagonal matrices  $D_{M,\ell}$  and  $D_{M,r}$  such that*

$$(2.1) \quad D_{M,\ell} M D_{M,r} \mathbf{1}_n = r \quad \text{and} \quad \mathbf{1}_m^T D_{M,\ell} M D_{M,r} = c^T$$

*if and only if there exists a nonnegative matrix  $S$  with the same zero pattern as  $M$  such that  $S \mathbf{1}_n = r$  and  $\mathbf{1}_m^T S = c^T$ .*

This is a nontrivial existence result that in a less general form appeared in [13]. To tackle the problem of finding the scaled matrix, one can perform a *Sinkhorn-Knopp-like algorithm* by alternatively normalizing the row and column sums of  $M$  as follows.

**Algorithm 1** (*Sinkhorn-Knopp-like algorithm for nonnegative  $M \in \mathbb{R}^{m \times n}$* ).

Initialize:  $D_{M,\ell} = I_m$  and  $D_{M,r} = I_n$

- (1) Multiply each row  $i$  of  $M$  and of  $D_{M,\ell}$  by  $\frac{r_i}{\sum_j m_{ij}}$  to obtain an updated matrix  $M$  with row sums  $r$  and an updated matrix  $D_{M,\ell}$ .
- (2) Multiply each column  $j$  of the updated  $M$  and of  $D_{M,r}$  by  $\frac{c_j}{\sum_i m_{ij}}$  to obtain an updated matrix  $M$  with column sums  $c$  and an updated matrix  $D_{M,r}$ .
- (3) If the row sums of the matrix  $M$  obtained in step (2) are far from  $r$ , repeat steps (1) and (2) with such  $M$  until an adequate stopping criterion is satisfied.

We give a MATLAB code of Algorithm 1 in Appendix A. This algorithm has appeared as early as in [25] and [10], and according to [7, section 3.1], it has been rediscovered several times in the literature and has received different names as, for instance, the Kruithof’s projection method (see [11]) or the RAS method. In this paper, we call it the Sinkhorn-Knopp-like algorithm because if  $r = c = \mathbf{1}_n$  and  $M$  is square, then it collapses to the famous Sinkhorn-Knopp algorithm for scaling a nonnegative matrix to a doubly stochastic matrix [19]. If the Sinkhorn-Knopp-like algorithm converges, i.e.,  $M$  converges and the diagonal matrices of the iteration

converge to positive bounded diagonal matrices, the limit will be the scaled matrix  $D_{M,\ell}MD_{M,r}$  in Theorem 2.1.

The cost of one iteration of Algorithm 1, i.e., the combination of one step (1) and one step (2), is  $4mn + m + n$  flops, where the term  $m + n$  comes from updating  $D_{M,\ell}$  and  $D_{M,r}$ . A marginally more efficient algorithm that updates  $M$  implicitly can be found in [17, page 64] (see also [8, page 262] for the case  $r = c = \mathbf{1}_n$ ). The cost of one iteration of this implicit algorithm is  $4mn$  flops. However, this implicit algorithm requires an overhead cost of  $2mn$  flops for constructing the final scaled matrix, which may be larger than  $(m + n)$  times the number of iterations for large  $m$  and  $n$ . All these costs are much smaller if  $M$  is a sparse matrix.

Another important result is that there exists at most one solution for the diagonal scaling problem in (2.1) for any prescribed vectors  $r$  and  $c$ . This is stated in the following Theorem 2.2, which is a partial result of what is proven in [18, Theorem 4].

**THEOREM 2.2.** *Let  $M \in \mathbb{R}^{m \times n}$  be a nonnegative matrix and let  $r \in \mathbb{R}^{m \times 1}$  and  $c \in \mathbb{R}^{n \times 1}$  be strictly positive vectors satisfying  $\mathbf{1}_m^T r = c^T \mathbf{1}_n$ . Then there exists at most one two-sided scaled matrix  $S = D_{M,\ell}MD_{M,r}$  with row sums  $S\mathbf{1}_n = r$  and column sums  $\mathbf{1}_m^T S = c^T$ , where  $D_{M,\ell}$  and  $D_{M,r}$  are positive diagonal matrices.*

A less general version of Theorem 2.2 appeared in [13], and the general case is implicit in [14]. We emphasize that, although  $S$  is unique when it exists, the matrices  $D_{M,\ell}$  and  $D_{M,r}$  are not necessarily unique. We refer the reader to [18, Theorem 4] for a description of all matrices  $D_{M,\ell}$  and  $D_{M,r}$  that satisfy  $S = D_{M,\ell}MD_{M,r}$ .

A surprising and useful result is that the Sinkhorn-Knopp-like algorithm converges if and only if the scaling problem (2.1) has a solution. This was proved in general in [11] and for square nonnegative matrices and  $r = c = \mathbf{1}_n$  in [19], i.e., for the doubly stochastic case (see also [7, Theorem 4.1]). Next, we state this important result.

**THEOREM 2.3.** *Under the assumptions in Theorem 2.2, there exist diagonal matrices  $D_{M,\ell}$  and  $D_{M,r}$  with positive main diagonals such that (2.1) is satisfied if and only if the Sinkhorn-Knopp-like algorithm converges.*

Thus, the Sinkhorn-Knopp-like algorithm gives a reliable numerical procedure to check for scalability. Unfortunately, the Sinkhorn-Knopp-like algorithm can be very slow, in particular for sparse matrices, and other faster algorithms have been developed in the literature (see [7, section 7], [17, section 4.3] and the references therein). However, we emphasize that, for the main purpose of this paper, i.e., improving the accuracy of computed eigenvalues of pencils, we have always found that the Sinkhorn-Knopp-like algorithm is fast enough and that the cost of its application is much smaller than the cost of computing the eigenvalues. The reason is that, in this case, the diagonal entries of the scalings  $D_{M,\ell}$  and  $D_{M,r}$  to be applied to the pencil must be integer powers of 2 which allows us to use a very relaxed stopping criterion in the Sinkhorn-Knopp-like algorithm. We will discuss this issue in depth in section 6.

One can find necessary and sufficient nonalgorithmic conditions for the scaled matrix to exist in [18, Theorem 2], [3, Theorem 2.1], and [7, Theorem 4.1]. However, these conditions depend on nontrivial properties that must be satisfied by the vectors  $r$  and  $c$ , as those we state in Lemma 3.7. In general, necessary and sufficient conditions depending only on the zero pattern of  $M$  are not known. A remarkable exception to this comment is the doubly stochastic scaling problem  $r = c = \mathbf{1}_n$  for square matrices, where such a condition is provided by the total support of the matrix (see section 3). In the next section, we will present new simple sufficient conditions depending only on the zero pattern for diagonal scalings to exist with a prescribed common vector for the row and column sums in the case of balancing square pencils and matrices.

**3. Scaling square pencils and related problems.** Let us first look at the case of square pencils. In [12, page 259], positive diagonal matrices  $D_\ell$  and  $D_r$  are chosen to equilibrate the row and column norms of a complex  $n \times n$  regular pencil  $\lambda B - A$  by imposing

$$(3.1) \quad \|\text{col}_j(\tilde{A})\|_2^2 + \|\text{col}_j(\tilde{B})\|_2^2 = \|\text{row}_i(\tilde{A})\|_2^2 + \|\text{row}_i(\tilde{B})\|_2^2 = \gamma^2 \text{ for } i, j = 1, \dots, n$$

for some constant  $\gamma$  resulting from the balancing, where  $\tilde{A} := D_\ell A D_r$  and  $\tilde{B} := D_\ell B D_r$  and  $\|\cdot\|_2$  denotes the standard Euclidean norm of a vector [5]. A pencil satisfying these conditions was called *balanced*, and an algorithm was presented in [12] to compute a scaling to balance a regular pencil  $\lambda B - A$ . It was shown that this amounts to solving the norm minimization problem

$$(3.2) \quad \inf_{\det D_\ell, \det D_r=1} \|D_\ell(\lambda B - A)D_r\|_F^2,$$

using the so-called Frobenius norm of a pencil:  $\|\lambda B - A\|_F^2 := \|B\|_F^2 + \|A\|_F^2$ , where  $\|A\|_F$  and  $\|B\|_F$  are the matrix Frobenius norms of  $A$  and  $B$  [5]. Moreover, the following result was proven in [12].

**THEOREM 3.1.** *The minimization problem*

$$(3.3) \quad \inf_{\det T_\ell, \det T_r=1} \|T_\ell(\lambda B - A)T_r\|_F^2,$$

where  $T_\ell$  and  $T_r$  are arbitrary nonsingular matrices, has a so-called standardized normal pencil  $\lambda \hat{B} - \hat{A}$  as a solution, satisfying

$$U_\ell(\lambda \hat{B} - \hat{A})U_r = \lambda \Lambda_B - \Lambda_A, \quad U_\ell^* U_\ell = U_r^* U_r = I_n, \quad |\Lambda_B|^2 + |\Lambda_A|^2 = \gamma^2 I_n,$$

where  $\Lambda_B$  and  $\Lambda_A$  are diagonal. If the eigenvalues of the regular pencil  $\lambda B - A$  are distinct, then  $T_\ell$  and  $T_r$  have a bounded solution, and the infimum is a minimum; otherwise they may be unbounded.

As shown in [12], the standardized normal pencils have eigenvalues with a condition number bounded by  $\sqrt{2}$ . Thus, performing the same minimization over the diagonal scalings is likely to improve the sensitivity of the eigenvalue computation. Moreover, if the transformation matrices are bounded, the eigenstructure of the regular pencil is preserved. But the positive diagonal scalings that achieve the balancing in [12] are not unique, and they may not exist or may be unbounded. In order to analyze this further we relate this problem to that of scaling a real nonnegative square matrix by two-sided scalings to a doubly stochastic matrix. Algorithm 1 with  $r = c = \mathbf{1}_n$  solves this problem. Its analysis can be found in [19, 8]. The link between both problems is the following: let us define the nonnegative matrices

$$(3.4) \quad M := |A|^{\circ 2} + |B|^{\circ 2}, \quad \text{and} \quad \tilde{M} := |\tilde{A}|^{\circ 2} + |\tilde{B}|^{\circ 2},$$

where  $|X|$  indicates the elementwise absolute value of the matrix  $X$ , where  $X^{\circ 2}$  indicates the elementwise square of the matrix  $X$ , and where  $D_\ell$  and  $D_r$  satisfy the balancing equations (3.1). Then the scaled matrix  $\tilde{M} = D_\ell^2 M D_r^2$  satisfies

$$\tilde{M} \mathbf{1}_n = D_\ell^2 (|A|^{\circ 2} + |B|^{\circ 2}) D_r^2 \mathbf{1}_n = \gamma^2 \mathbf{1}_n, \quad \mathbf{1}_n^T \tilde{M} = \mathbf{1}_n^T D_\ell^2 (|A|^{\circ 2} + |B|^{\circ 2}) D_r^2 = \gamma^2 \mathbf{1}_n^T,$$

which implies that  $\tilde{M}/\gamma^2$  is doubly stochastic and that the two-sided scaling for the nonnegative matrix  $M$  satisfies

$$\tilde{M}/\gamma^2 = D_{M,\ell} M D_{M,r}, \quad \text{where} \quad D_{M,\ell} := D_\ell^2/\gamma, \quad D_{M,r} := D_r^2/\gamma.$$

The only difference is that, for balancing, we impose a scalar constraint  $\det D_\ell \cdot \det D_r = 1$ , which is why the resulting row and column norms are equal to  $\gamma^2$  rather than 1. In fact, the algorithm proposed in [12] was alternately normalizing the rows and columns of  $M$  to 1 (rather than  $\gamma$ ), and that is precisely the algorithm of Sinkhorn–Knopp. This connection was not established in [12]. It follows from this that the unicity or boundedness of the scalings are equivalent for the two problems.

We recall in Theorem 3.4 the results given for two-sided scaling in [19] for square nonnegative matrices  $M \in \mathbb{R}^{n \times n}$  in order for the corresponding matrix to become doubly stochastic. Before stating Theorem 3.4, we introduce the notions of total support and full indecomposability that will be used.

**DEFINITION 3.2.** *The sequence  $m_{1,\sigma(1)}, m_{2,\sigma(2)}, \dots, m_{n,\sigma(n)}$ , where  $\sigma$  is a permutation of  $\{1, 2, \dots, n\}$ , is called a diagonal of  $M \in \mathbb{R}^{n \times n}$ . A nonnegative matrix  $M \in \mathbb{R}^{n \times n}$  has total support if every positive element of  $M$  lies on a positive diagonal.*

**DEFINITION 3.3.** *A nonnegative matrix  $M \in \mathbb{R}^{n \times n}$  is fully indecomposable if there do not exist permutation matrices  $P_\ell$  and  $P_r$  such that  $P_\ell M P_r$  can be partitioned as*

$$P_\ell M P_r = \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix},$$

where  $M_{11}$  and  $M_{22}$  are square matrices.

It was proved in [2] that a fully indecomposable matrix has total support.

**THEOREM 3.4** (Sinkhorn–Knopp). *If  $M \in \mathbb{R}^{n \times n}$  is a nonnegative matrix, then a necessary and sufficient condition needed so that there exists a doubly stochastic matrix  $S$  of the form  $S = D_{M,\ell} M D_{M,r}$ , where  $D_{M,\ell}$  and  $D_{M,r}$  are positive diagonal matrices, is that  $M$  has total support. If  $S$  exists, then it is unique.  $D_{M,\ell}$  and  $D_{M,r}$  are also unique up to a nonnegative scalar multiple if and only if  $M$  is fully indecomposable.*

The doubly stochastic matrix  $S$  can be obtained by applying Algorithm 1 with  $r = c = \mathbf{1}_n$ , which is the Sinkhorn–Knopp algorithm. As a consequence of Theorems 2.3 and 3.4, a necessary and sufficient condition needed so that the Sinkhorn–Knopp algorithm applied to  $M$  will converge to a doubly stochastic limit of the form  $D_{M,\ell} M D_{M,r}$  is that  $M$  has total support [8, 19].

We recall in Theorem 3.5 the particular case of a symmetric and fully indecomposable matrix  $M$ , which will be important in the new scaling method of section 5.

**THEOREM 3.5** ([8, Lemma 4.1]). *If  $M \in \mathbb{R}^{n \times n}$  is a symmetric nonnegative and fully indecomposable matrix, then there exists a unique diagonal matrix  $D$  with a positive main diagonal such that  $DMD$  is doubly stochastic.*

*Remark 3.6.* When  $M$  is fully indecomposable, the solution set for the diagonal scalings is  $\mathcal{S} := \{(D_{M,\ell}/c, cD_{M,r}) : c > 0\}$  for a given solution  $(D_{M,\ell}, D_{M,r})$ . To guarantee unicity for a solution in  $\mathcal{S}$ , one can consider a unique “normalized” scaling pair  $(D_{M,\ell}, D_{M,r})$ . For instance, by imposing that the solution satisfies  $\det D_{M,\ell} = \det D_{M,r}$  or  $\max_{i=1,\dots,n} \{d_i^\ell\} = \max_{i=1,\dots,n} \{d_i^r\}$ , where  $d_i^\ell$  and  $d_i^r$  are the diagonal entries of  $D_{M,\ell}$  and  $D_{M,r}$ , respectively. Then the pair  $(D_{M,\ell}, D_{M,r})$  is unique in  $\mathcal{S}$ . When  $M$  is symmetric, then these normalizations imply that  $D_{M,\ell} = D_{M,r}$ . In summary, one can always perform a normalization in order to obtain unicity for the diagonal scalings.

For the general scaling problem in Theorem 2.2, with arbitrary prespecified row and column sums, the sufficient conditions on  $M$  for the scaling to exist as simply as those in Theorem 3.4, which are based only on the zero pattern of  $M$ , are not known



to the best of our knowledge. This motivated us to develop the results in the next subsection.

**3.1. Diagonal scalings of square nonnegative matrices with prescribed common vector for the row and column sums.** We now derive simple sufficient conditions on the zero pattern for the existence of a diagonal scaling of a square matrix  $M$  by considering not only  $\mathbf{1}_n$  but any prescribed common vector  $v$  for the row and column sums. The conditions are a simple corollary of Lemma 3.7, which is a partial result of [18, Theorem 2]. In what follows, the support of a matrix  $A \in \mathbb{R}^{m \times n}$ , denoted by  $\text{supp}(A)$ , is defined as the set  $\{(i, j) \mid a_{ij} \neq 0, i = 1, \dots, m, \text{ and } j = 1, \dots, n\}$ .

**LEMMA 3.7.** *Let  $M \in \mathbb{R}^{m \times n}$  be a nonnegative matrix, and let  $r \in \mathbb{R}^{m \times 1}$  and  $c \in \mathbb{R}^{n \times 1}$  be strictly positive vectors such that  $\mathbf{1}_m^T r = c^T \mathbf{1}_n$ . Then there exists a scaled matrix  $S = D_{M,\ell} M D_{M,r}$  with row sums  $S \mathbf{1}_n = r$  and column sums  $\mathbf{1}_m^T S = c^T$ , where  $D_{M,\ell}$  and  $D_{M,r}$  are diagonal matrices with positive main diagonals if and only if there exists no pair of vectors  $(u, v) \in \mathbb{R}^m \times \mathbb{R}^n$  for which*

- (a)  $u_i + v_j \leq 0$  for each pair  $(i, j) \in \text{supp}(M)$ ,
- (b)  $r^T u = c^T v = 0$ , and
- (c)  $u_{i_0} + v_{j_0} < 0$  for some pair  $(i_0, j_0) \in \text{supp}(M)$ .

**THEOREM 3.8.** *Let  $M \in \mathbb{R}^{n \times n}$  be a nonnegative matrix with  $(i, i) \in \text{supp}(M)$  for all  $i = 1, \dots, n$  and  $\text{supp}(M) = \text{supp}(M^T)$ . Let  $v \in \mathbb{R}^{n \times 1}$  be a strictly positive vector. Then there exists a scaled matrix  $S = D_{M,\ell} M D_{M,r}$  with  $S \mathbf{1}_n = v$  and  $\mathbf{1}_n^T S = v^T$ , where  $D_{M,\ell}$  and  $D_{M,r}$  are positive diagonal matrices. Moreover,  $S$  is unique and is the limit of the Sinkhorn-Knopp-like algorithm. If, in addition,  $M$  is fully indecomposable, then  $D_{M,\ell}$  and  $D_{M,r}$  are unique up to a nonnegative scalar multiple, and if  $M = M^T$ , then there exists a unique positive diagonal matrix  $D$  such that  $S = DMD$ .*

*Proof.* Consider a matrix  $M$  as in the statement. By contradiction, let us assume that there exists no scaled matrix  $S$  with  $S \mathbf{1}_n = v$  and  $\mathbf{1}_n^T S = v^T$ . Then, by Lemma 3.7, there exists a pair of vectors  $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$  for which

- (a)  $x_i + y_j \leq 0$  for each pair  $(i, j) \in \text{supp}(M)$ ,
- (b)  $v^T x = v^T y = 0$ , and
- (c)  $x_{i_0} + y_{j_0} < 0$  for some pair  $(i_0, j_0) \in \text{supp}(M)$ .

Condition (b) implies that

$$(3.5) \quad v_1(x_1 + y_1) + \dots + v_n(x_n + y_n) = 0.$$

In addition, since  $(i, i) \in \text{supp}(M)$  for all  $i = 1, \dots, n$ , condition (a) implies that  $x_i + y_i \leq 0$  for all  $i = 1, \dots, n$ . It then follows from (3.5) that  $x_i + y_i = 0$  for all  $i = 1, \dots, n$  since  $v_i > 0$ . Moreover, by (c), there exists  $(i_0, j_0) \in \text{supp}(M)$  such that  $x_{i_0} + y_{j_0} < 0$ . Taking into account that  $x_i + y_i = 0$  for all  $i = 1, \dots, n$  we have that

$$(3.6) \quad (x_{i_0} + y_{i_0}) + (x_{j_0} + y_{j_0}) = 0.$$

From (3.6) and  $x_{i_0} + y_{j_0} < 0$ , we obtain that  $x_{j_0} + y_{i_0} > 0$ . Therefore, by (a),  $(j_0, i_0) \notin \text{supp}(M)$ , which is a contradiction since  $(i_0, j_0) \in \text{supp}(M) = \text{supp}(M^T)$ .

The uniqueness of  $S$  is a consequence of Theorem 2.2, and it is the limit of the Sinkhorn-Knopp-like algorithm by Theorem 2.3. If  $M$  is fully indecomposable, its bipartite graph is connected [4, Theorem 1.3.7], and thus, it is chainable [6, Theorem 1.2] (see [6] or [18] for the definition of “chainable”). Then, by [18, Theorem 4],  $D_{M,\ell}$  and  $D_{M,r}$  are also unique up to a nonnegative scalar multiple. Finally, if, in this situation,  $M = M^T$ , then transposing both sides of  $D_{M,\ell} M D_{M,r} \mathbf{1}_n = v$  and

$\mathbf{1}_n^T D_{M,\ell} M D_{M,r} = v^T$  implies  $\mathbf{1}_n^T D_{M,r} M D_{M,\ell} = v^T$  and  $D_{M,r} M D_{M,\ell} \mathbf{1}_n = v$ , which, combined with the uniqueness of  $D_{M,\ell}$  and  $D_{M,r}$  up to a scalar multiple, implies that  $D_{M,r} = \alpha D_{M,\ell}$  for some  $\alpha > 0$ , and  $D = \sqrt{\alpha} D_{M,\ell}$  is the unique nonnegative diagonal matrix satisfying  $S = DMD$ .  $\square$

If  $M$  satisfies the conditions in Theorem 3.8, the scaled matrix  $S$  can be computed by using the Sinkhorn-Knopp-like algorithm in Appendix A with  $r = c = v$ .

In section 5, we will present new cost functions for our minimization problem (3.2) to make sure that it always has a unique and bounded solution. This new approach will be based on the results presented in this section combined with regularization techniques. In addition, this new approach will be applied to arbitrary pencils (square or nonsquare). First, we study in section 4 the unregularized nonsquare case.

**4. Scaling nonsquare pencils and related problems.** In the square case, we scaled the pencil so that its row norms and column norms were equal as in (3.1). This is not possible for  $m \times n$  rectangular pencils since the numbers of rows and columns are different. Instead, one can try to balance  $\lambda B - A$  by achieving the following equalities:

$$(4.1) \quad \begin{aligned} \|\text{col}_j(\tilde{A})\|_2^2 + \|\text{col}_j(\tilde{B})\|_2^2 &= \gamma_\ell^2, \text{ for } j = 1, \dots, n \text{ and} \\ \|\text{row}_i(\tilde{A})\|_2^2 + \|\text{row}_i(\tilde{B})\|_2^2 &= \gamma_r^2, \text{ for } i = 1, \dots, m, \end{aligned}$$

where  $\tilde{A} := D_\ell A D_r$  and  $\tilde{B} := D_\ell B D_r$  and  $\|\lambda \tilde{B} - \tilde{A}\|_F^2 = n\gamma_\ell^2 = m\gamma_r^2$ . For the nonsquare case, we also define the nonnegative matrices

$$(4.2) \quad M := |A|^{\circ 2} + |B|^{\circ 2} \quad \text{and} \quad \tilde{M} := |\tilde{A}|^{\circ 2} + |\tilde{B}|^{\circ 2}.$$

The scaling problem discussed in this section is a special case of the general scaling problem in Theorem 2.2, where we choose  $r = \gamma_r^2 \mathbf{1}_m$  and  $c = \gamma_\ell^2 \mathbf{1}_n$ .

We now show that there is an optimization problem whose first order optimality conditions correspond to the equalities in (4.1).

**THEOREM 4.1.** *The minimization problem over the set of positive diagonal matrices  $D_\ell = \text{diag}(d_{\ell_1}, \dots, d_{\ell_m})$  and  $D_r = \text{diag}(d_{r_1}, \dots, d_{r_n})$*

$$\inf_{\det D_\ell^2 = c_\ell, \det D_r^2 = c_r} (\|D_\ell A D_r\|_F^2 + \|D_\ell B D_r\|_F^2)$$

*has the balancing equations (4.1) as first order optimality conditions.*

*Proof.* If one makes the change of variables for the elements of  $D_\ell$  and  $D_r$  as  $d_{\ell_i}^2 = \exp(u_i)$ ,  $d_{r_j}^2 = \exp(v_j)$  and notes that  $m_{ij} := |a_{ij}|^2 + |b_{ij}|^2$ , then the above minimization is equivalent to a convex minimization problem with linear constraints:

$$(4.3) \quad \inf \sum_{i=1}^m \sum_{j=1}^n m_{ij} \exp(u_i + v_j) \quad \text{subject to} \quad \sum_{i=1}^m u_i = \ln c_\ell, \quad \sum_{j=1}^n v_j = \ln c_r.$$

The corresponding unconstrained problem with Lagrange multipliers  $\Gamma_\ell$  and  $\Gamma_r$  is

$$\inf \sum_{i=1}^m \sum_{j=1}^n m_{ij} \exp(u_i + v_j) + \Gamma_\ell \left( \ln c_\ell - \sum_{i=1}^m u_i \right) + \Gamma_r \left( \ln c_r - \sum_{j=1}^n v_j \right).$$

The first order conditions of optimality are the equality constraints of (4.3) and the equations

$$(4.4) \quad \sum_{j=1}^n d_{\ell_i}^2 m_{ij} d_{r_j}^2 = \Gamma_\ell, \quad \sum_{i=1}^m d_{\ell_i}^2 m_{ij} d_{r_j}^2 = \Gamma_r,$$

which express exactly that the row norms of  $\widetilde{M} := D_\ell^2 M D_r^2$  are equal to each other and that its column norms are equal to each other. Since the Lagrange multipliers  $\Gamma_\ell$  and  $\Gamma_r$  are clearly nonnegative, we can write them as  $\gamma_\ell^2 := \Gamma_\ell$  and  $\gamma_r^2 := \Gamma_r$ , which completes the proof.  $\square$

Unfortunately, the optimization problem in Theorem 4.1 does not always have a solution. If there exists solution, it can be obtained by a sequence of alternating scalings  $D_\ell^2$  and  $D_r^2$  that make the rows of  $D_\ell^2 (M D_r^2)$  have equal sum  $\gamma_r^2$ , and then the columns of  $(D_\ell^2 M) D_r^2$  have equal sum  $\gamma_\ell^2$ , while maintaining the constraints  $\det D_\ell^2 = c_\ell$ ,  $\det D_r^2 = c_r$  in the accumulated diagonal transformations, which determine the values of  $\gamma_r^2$  and  $\gamma_\ell^2$ . The cyclic alternation of row and column scalings, then amounts to coordinate descent applied to the minimization. This algorithm thus continues to decrease the cost function as long as the equalities (4.4) are not met. This is very similar to the Sinkhorn-Knopp-like Algorithm 1 applied to  $M$  with  $r = \gamma_r^2 \mathbf{1}_m$  and  $c = \gamma_\ell^2 \mathbf{1}_n$ . Since the exact values of  $\gamma_r^2$  and  $\gamma_\ell^2$  are of no interest, in practice one can simply apply Algorithm 1 to  $M$  with  $r = n \mathbf{1}_m$  and  $c = m \mathbf{1}_n$ , which converges if and only if the corresponding scaling problem has solution, according to Theorem 2.3.

**5. The regularized scaling method for pencils.** The facts that for a non-square pencil the doubly stochastic scaling cannot be applied anymore, that even for square pencils the corresponding matrix  $M$  may not have total support, and that the optimization problem in Theorem 4.1 does not always have a solution can be bypassed by introducing a regularization term which will ensure an essentially unique bounded solution for  $D_\ell$  and  $D_r$ . The cost is that we will obtain a solution of an approximate problem. But, with the new approach, we will always find such a solution.

Given two complex matrices  $A, B$  of size  $m \times n$ , we consider the following constrained minimization problem over the set of positive diagonal matrices  $D_\ell = \text{diag}(d_{\ell_1}, \dots, d_{\ell_m})$  and  $D_r = \text{diag}(d_{r_1}, \dots, d_{r_n})$ :

$$(5.1) \quad \inf_{\det D_\ell^2 \det D_r^2 = c} 2(\|D_\ell A D_r\|_F^2 + \|D_\ell B D_r\|_F^2) + \alpha^2 \left( \frac{1}{m^2} \|D_\ell\|_F^4 + \frac{1}{n^2} \|D_r\|_F^4 \right)$$

for some real number  $c > 0$  and a regularization parameter  $\alpha$ . If we denote again the matrix  $M := |A|^{\circ 2} + |B|^{\circ 2}$ , then we can rewrite this as

$$(5.2) \quad \inf_{\det D_\ell^2 \det D_r^2 = c} \mathbf{1}_{m+n}^T \begin{bmatrix} \frac{\alpha^2}{m^2} D_\ell^2 \mathbf{1}_m \mathbf{1}_m^T D_\ell^2 & D_\ell^2 M D_r^2 \\ D_r^2 M^T D_\ell^2 & \frac{\alpha^2}{n^2} D_r^2 \mathbf{1}_n \mathbf{1}_n^T D_r^2 \end{bmatrix} \mathbf{1}_{m+n},$$

which suggests that there may be a link to doubly stochastic scaling. Indeed, let us consider the two-sided scaling problem  $\widetilde{M}_\alpha := D_{\ell,r} M_\alpha D_{\ell,r}$ , where

$$D_{\ell,r} := \begin{bmatrix} D_\ell & 0 \\ 0 & D_r \end{bmatrix},$$

subject to  $\det D_\ell^2 \det D_r^2 = \det D_{\ell,r}^2 = c$ , and

$$(5.3) \quad M_\alpha^{\circ 2} = \begin{bmatrix} \frac{\alpha^2}{m^2} \mathbf{1}_m \mathbf{1}_m^T & M \\ M^T & \frac{\alpha^2}{n^2} \mathbf{1}_n \mathbf{1}_n^T \end{bmatrix}.$$

Notice that both diagonal blocks in  $M_\alpha$  have Frobenius norm  $\alpha$ . We prove in Theorem 5.2 that the optimization problem (5.1) can be solved by the Sinkhorn-Knopp algorithm in a unique way. We will need the following auxiliary Lemma 5.1 in our proof.

LEMMA 5.1. Let  $M_\alpha^{\circ 2}$  be the nonnegative matrix in (5.3) with  $\alpha \neq 0$ . Then  $M_\alpha^{\circ 2}$  has total support. Moreover, if  $M \neq 0$ , then  $M_\alpha^{\circ 2}$  is fully indecomposable.

*Proof.* See Appendix B. □

THEOREM 5.2. Let  $A$  and  $B$  be  $m \times n$  complex matrices and  $\alpha, c > 0$  be real numbers. Let us consider the constrained minimization problem (5.1) over the set  $\{(D_\ell, D_r) : D_\ell := \text{diag}(\delta_{\ell_1}, \dots, \delta_{\ell_m}), D_r := \text{diag}(\delta_{r_1}, \dots, \delta_{r_n}), \delta_{\ell_i} > 0, \delta_{r_j} > 0\}$ . Then the following statements hold:

- (a) The optimization problem (5.1) is equivalent to the optimization problem (5.2).
- (b) The optimization problem (5.1) is equivalent to the optimization problem

$$\inf_{\det D_\ell^2 \det D_r^2 = c} \left\| \begin{bmatrix} D_\ell & 0 \\ 0 & D_r \end{bmatrix} M_\alpha \begin{bmatrix} D_\ell & 0 \\ 0 & D_r \end{bmatrix} \right\|_F^2,$$

where  $M_\alpha^{\circ 2}$  is given in (5.3).

- (c) There exists a unique solution  $(\tilde{D}_\ell, \tilde{D}_r)$  of (5.1). Moreover,  $(\tilde{D}_\ell, \tilde{D}_r)$  is bounded and makes the matrix

$$\begin{bmatrix} \tilde{D}_\ell^2 & 0 \\ 0 & \tilde{D}_r^2 \end{bmatrix} M_\alpha^{\circ 2} \begin{bmatrix} \tilde{D}_\ell^2 & 0 \\ 0 & \tilde{D}_r^2 \end{bmatrix}$$

a scalar multiple of a doubly stochastic matrix. Therefore,  $(\tilde{D}_\ell, \tilde{D}_r)$  can be computed, up to a scalar multiple, by applying the algorithm in Appendix A to  $M_\alpha^{\circ 2}$  with  $r = c = \mathbf{1}_{m+n}$ .

*Proof.* We have already seen parts (a) and (b) in this section because the optimization problem in (b) is just (5.2). Then we only need to prove (c). We make the change of variables  $\delta_{\ell_i}^2 = \exp(u_i)$  and  $\delta_{r_j}^2 = \exp(v_j)$  for the elements of  $D_\ell$  and  $D_r$ , respectively. Then the optimization problem (5.1) is equivalent to the optimization problem:

(5.4)

$$\inf 2 \sum_{i=1}^m \sum_{j=1}^n m_{ij} \exp(u_i + v_j) + \alpha^2 \left( \frac{1}{m^2} \left( \sum_{i=1}^m \exp(u_i) \right)^2 + \frac{1}{n^2} \left( \sum_{j=1}^n \exp(v_j) \right)^2 \right)$$

subject to  $\sum_{i=1}^m u_i + \sum_{j=1}^n v_j = \ln c$ .

The corresponding unconstrained problem with Lagrange multiplier  $\Gamma$  is

(5.5)

$$\inf 2 \sum_{i=1}^m \sum_{j=1}^n m_{ij} \exp(u_i + v_j) + \alpha^2 \left( \frac{1}{m^2} \left( \sum_{i=1}^m \exp(u_i) \right)^2 + \frac{1}{n^2} \left( \sum_{j=1}^n \exp(v_j) \right)^2 \right) + \Gamma \left( \ln c - \sum_{i=1}^m u_i - \sum_{j=1}^n v_j \right).$$

The first order conditions of optimality are the equality constraint of (5.4) and

$$\frac{\alpha^2}{m^2} d_{\ell_i}^2 \sum_{i=1}^m d_{\ell_i}^2 + \sum_{j=1}^n d_{\ell_i}^2 m_{ij} d_{r_j}^2 = \frac{\Gamma}{2} \quad \text{and} \quad \frac{\alpha^2}{n^2} d_{r_j}^2 \sum_{j=1}^n d_{r_j}^2 + \sum_{i=1}^m d_{\ell_i}^2 m_{ij} d_{r_j}^2 = \frac{\Gamma}{2}$$

for  $i = 1, \dots, m$  and  $j = 1, \dots, n$ , respectively, which express that the row sum and the column sum of

$$\begin{bmatrix} D_\ell^2 & 0 \\ 0 & D_r^2 \end{bmatrix} M_\alpha^{\circ 2} \begin{bmatrix} D_\ell^2 & 0 \\ 0 & D_r^2 \end{bmatrix}$$

are equal to  $\frac{\Gamma}{2}$ . By Lemma 5.1, we know that  $M_\alpha^{\circ 2}$  is fully indecomposable. Then, by Theorem 3.4, there exists a unique and bounded scaling  $(E_\ell, E_r)$  that makes

$$\begin{bmatrix} E_\ell^2 & 0 \\ 0 & E_r^2 \end{bmatrix} M_\alpha^{\circ 2} \begin{bmatrix} E_\ell^2 & 0 \\ 0 & E_r^2 \end{bmatrix}$$

doubly stochastic. Assume that  $\det E_\ell^2 \det E_r^2 = k$ . We define  $\tilde{D}_\ell := (\frac{c}{k})^{\frac{1}{2(m+n)}} E_\ell$  and  $\tilde{D}_r := (\frac{c}{k})^{\frac{1}{2(m+n)}} E_r$ . Then  $\det \tilde{D}_\ell^2 \det \tilde{D}_r^2 = c$ , and  $(\tilde{D}_\ell, \tilde{D}_r)$  is the solution of (5.1). We can again redefine  $\gamma^2 := \Gamma/2$  since this quantity is nonnegative.  $\square$

For completeness, we include the following result, which is a direct corollary of the proof of Theorem 5.2.

**THEOREM 5.3.** *Let  $A$  and  $B$  be  $m \times n$  complex matrices and  $\alpha, c > 0$  be real numbers. Then the constrained minimization problem*

$$\inf_{\det D_\ell^2 \det D_r^2 = c} 2(\|D_\ell A D_r\|_F^2 + \|D_\ell B D_r\|_F^2) + \alpha^2 \left( \frac{1}{m^2} \|D_\ell\|_F^4 + \frac{1}{n^2} \|D_r\|_F^4 \right)$$

over the set  $\{(D_\ell, D_r) : D_\ell := \text{diag}(\delta_{\ell_1}, \dots, \delta_{\ell_m}), D_r := \text{diag}(\delta_{r_1}, \dots, \delta_{r_n}), \delta_{\ell_i} > 0, \delta_{r_j} > 0\}$  has a unique and bounded solution. Moreover, it satisfies the equations

$$\begin{aligned} \|col_j(\tilde{A})\|_2^2 + \|col_j(\tilde{B})\|_2^2 + \frac{\alpha^2}{n^2} \delta_{r_j}^2 \|D_r\|_F^2 &= \gamma^2, \text{ for } j = 1, \dots, n, \text{ and} \\ \|row_i(\tilde{A})\|_2^2 + \|row_i(\tilde{B})\|_2^2 + \frac{\alpha^2}{m^2} \delta_{\ell_i}^2 \|D_\ell\|_F^2 &= \gamma^2, \text{ for } i = 1, \dots, m, \end{aligned}$$

for some nonzero scalar  $\gamma$ , where  $\tilde{A} := D_\ell A D_r$  and  $\tilde{B} := D_\ell B D_r$ .

*Remark 5.4.* We know that the row sums and the column sums of the matrix in Theorem 5.2(c) are equal to each other. The quantity of such row and column sums is the scalar  $\gamma^2$  appearing in Theorem 5.3.

In Example 5.6, we will illustrate the effect of choosing different values for the regularization parameter  $\alpha$  in (5.3) in order to make the row and column sums of  $D_\ell^2 M D_r^2$  as equal as possible for a square matrix  $M$  (corresponding to a pencil  $\lambda B - A$ ) which does not have total support and, thus, cannot be scaled to a multiple of a doubly stochastic matrix. For measuring the quality of the obtained approximate scaling in this and other examples considered in this paper, we introduce the following definition.

**DEFINITION 5.5.** *Let  $M \in \mathbb{R}^{m \times n}$  be a real nonnegative matrix, let  $r(M) \in \mathbb{R}^{m \times 1}$  and  $c(M) \in \mathbb{R}^{n \times 1}$  be, respectively, the vectors of row sums and column sums of  $M$ , denote by  $r_i(M)$  and  $c_i(M)$  their  $i$ th entries, and assume  $r_i(M) > 0$  and  $c_j(M) > 0$  for all  $i, j$ . Then, the quality-factor of the homogeneous scaling of  $M$  is defined as*

$$(5.6) \quad q_S(M) := \max \left\{ \frac{\max_i r_i(M)}{\min_i r_i(M)}, \frac{\max_j c_j(M)}{\min_j c_j(M)} \right\}.$$

The row sums of  $M$  are all equal and the column sums of  $M$  are all equal if and only if  $q_S(M) = 1$ . The closer to one  $q_S(M)$  is, the better balanced the matrix  $M$  is.

*Example 5.6.* We consider the regular pencil

$$(5.7) \quad \lambda B_1 - A_1 := \begin{bmatrix} 1 & \lambda & 0 \\ \lambda & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{with} \quad M_1 = |A_1|^{\circ 2} + |B_1|^{\circ 2} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

$M_1$  has no total support and, thus, the Sinkhorn–Knopp algorithm does not converge. In our experiment, the algorithm in Appendix A applied to  $M_1$  with  $r = c = \mathbf{1}_3$  and  $\mathbf{tol} = 10^{-3}$  does not converge after 1000 steps. In contrast, the same algorithm applied to the matrix  $M_\alpha^{\circ 2}$  in (5.3) with  $r = c = \mathbf{1}_6$  and  $\mathbf{tol} = 10^{-3}$  for  $\alpha = 1, 0.5, 0.1$  converges and produces scaled matrices  $\widetilde{M}_\alpha = D_\ell^2 M D_r^2$  which are approximately doubly stochastic up to a scalar multiple. The results are shown in Table 1, where the last column shows the 2-norm condition numbers of  $D_\ell^2 \approx D_r^2$  and steps denote the number of steps until convergence, with each step comprising one right and one left diagonal scaling.

A smaller  $\alpha$  yields a better equilibration for the row and column sums as measured by  $q_S$  (to be compared with  $q_S(M_1) = 2$ ) but at the cost of a worse conditioning of the scaling matrices  $D_\ell^2, D_r^2$  and a slower convergence. The latter is to be expected since for  $\alpha = 0$  the scaling to a multiple of a double stochastic matrix does not exist for  $M_1$ .

Finally, we show the results obtained when the algorithm in Appendix A is applied directly to  $M_1$  with  $r = c = \mathbf{1}_3$ , i.e., without any regularization, but with the very relaxed stopping criterion  $\mathbf{tol} = 1$ , which will be used in all the numerical experiments for improving the accuracy of eigenvalues in section 6 and in the accompanying supplementary file `supp_materials.pdf` [local/web 354KB]. In this case the algorithm converges in only 3 steps, and the results are shown in Table 2, where  $\alpha = 0$  indicates that the problem has not been regularized (though the matrix  $M_\alpha^{\circ 2}$  is not used at all). We will use this convention in other numerical examples and tests. The motivation for computing this rough  $\mathbf{tol} = 1$  approximate solution will be explained in section 6 and is related to the fact that for the purpose of improving the accuracy of the eigenvalues of  $\lambda B - A$  computed in floating point arithmetic it is essential that the entries of the diagonal scaling matrices  $D_\ell$  and  $D_r$  be integer powers of 2. Thus, it makes no sense to compute very precise scaling matrices  $D_\ell$  and  $D_r$  since their entries will be later rounded to their nearest integer powers of 2, and thus, a relaxed stopping criterion can be used. We remark here three natural facts that will be further discussed in section 6 and that are confirmed by the numerical experiments in section

TABLE 1

Results of the regularization applied to the matrix  $M_1$  in (5.7) with  $\mathbf{tol} = 10^{-3}$ . The quality factors  $q_S$  should be compared with  $q_S(M_1) = 2$ .

$\alpha$	Steps	$q_S$	$\text{diag}(D_\ell^2) \approx \text{diag}(D_r^2)$	$\kappa(D_\ell^2) \approx \kappa(D_r^2)$
1	11	1.38	0.485, 1.29, 0.864	2.66
0.5	24	1.19	0.395, 2.05, 0.952	5.19
0.1	124	1.04	0.187, 5.15, 0.970	27.5

TABLE 2

Results of the unregularized Sinkhorn–Knopp algorithm applied to the matrix  $M_1$  in (5.7) with  $\mathbf{tol} = 1$ . The quality factor  $q_S$  should be compared with  $q_S(M_1) = 2$ .

$\alpha$	Steps	$q_S$	$\text{diag}(D_\ell^2)$	$\text{diag}(D_r^2)$	$\kappa(D_\ell^2)$	$\kappa(D_r^2)$
0	3	1.33	0.350, 2.45, 0.765	0.408, 2.45, 1.31	7	6

6 and in the supplementary file `supp_materials.pdf` [local/web 354KB]: `tol=1` very often has a regularization effect, speeds up considerably the convergence, and yields a reasonably “well-balanced” matrix. Observe that a large `tol` implies that the row sums and column sums of the matrix obtained after applying the Sinkhorn–Knopp algorithm are not necessarily too close to be equal to each other. Thus, it is natural that the algorithm may run successfully and produce a reasonable output for matrices that cannot be exactly scaled to a multiple of a doubly stochastic matrix. This is the situation illustrated in Table 2.

*Remark 5.7.* The choice of the regularization parameter  $\alpha$  has to be guided by the equilibrium one wants to achieve between the “quality” of the balancing, the boundedness/conditioning of the diagonal scaling matrices, and the speed of convergence. This depends on the applied problem to be solved. For improving the accuracy of computed eigenvalues, we do not need to consider a very small value of  $\alpha$  since it is enough to get a reasonably “well-balanced” matrix  $M$  because the entries of the diagonal scaling matrices have to be later rounded to their nearest integer powers of 2. Moreover, the use of the relaxed stopping criterion `tol=1` makes the use of the regularization often unnecessary, even in cases where there is no exact solution of the scaling problem. Thus, for the eigenvalue problem, we recommend to start always by using the unregularized method and if it does not converge in a small number of steps (say  $n/10$  for large  $n$ ) to change to the regularized method with a value of  $\alpha \lesssim 0.5 \max_{ij} \sqrt{M_{ij}}$ . In contrast, in other types of problems where it is important to always get a very “well-balanced matrix” and a relaxed stopping criterion is not adequate or necessary, a recommendable option might be to always use the regularization with a small value of  $\alpha$ , especially when  $M$  is sparse, since it guarantees the existence of a solution. This will increase the complexity of the Sinkhorn–Knopp algorithm each step by a factor 4 since the matrix sizes are doubled. In difficult cases, this might be very slow, and the regularized problem and the Sinkhorn–Knopp algorithm should be combined with faster algorithms (see [7, 17]).

*Remark 5.8.* One could also have considered for the regularization the cost function

$$\inf_{\det D_\ell^2 \det D_r^2 = c} 2(\|D_\ell A D_r\|_F^2 + \|D_\ell B D_r\|_F^2) + \alpha^2 (\|D_\ell^2\|_F^2 + \|D_r^2\|_F^2),$$

which would correspond to the matrix

$$M_\alpha^{\circ 2} := \begin{bmatrix} \alpha^2 I_m & M \\ M^T & \alpha^2 I_n \end{bmatrix}.$$

This matrix has total support for  $\alpha > 0$ . However, it is not necessarily fully indecomposable (assume, for instance, that  $M$  has a zero row or column), and therefore, we cannot guarantee the essential uniqueness of the scaling matrices  $D_\ell$  and  $D_r$ .

**5.1. The regularized method with prescribed nonhomogeneous common vector for the row and column sums.** In the nonsquare case, we know from the discussions of section 4 that making the column and row sums of  $\widetilde{M} = D_\ell^2 M D_r^2$  become equal cannot be achieved exactly, where  $M$  is the matrix in (4.2). In this case, we can use the regularized method in Theorem 5.2(c) in order to obtain a scaling that balances  $\widetilde{M}$  approximately. We have used this approach on many problems and have obtained pretty satisfactory results. However, since by using this method we always obtain a scalar multiple of a doubly stochastic matrix as a solution for  $M_\alpha^{\circ 2}$ , this method considers in some sense the rows and columns of  $M$  in the same way,

which is not natural in the rectangular case. Thus, one possible strategy for improving this approach is not to request that  $M_\alpha^{\circ 2}$  is scaled to be a scalar multiple of a doubly stochastic matrix but to impose a modified scaling with prescribed common vector

$$(5.8) \quad v := \begin{bmatrix} n\mathbf{1}_m \\ m\mathbf{1}_n \end{bmatrix}$$

for the row and column sums. The new regularized method is then described by

$$(5.9) \quad \begin{bmatrix} D_\ell^2 & 0 \\ 0 & D_r^2 \end{bmatrix} M_\alpha^{\circ 2} \begin{bmatrix} D_\ell^2 & 0 \\ 0 & D_r^2 \end{bmatrix} \begin{bmatrix} \mathbf{1}_m \\ \mathbf{1}_n \end{bmatrix} = v$$

and

$$(5.10) \quad \begin{bmatrix} \mathbf{1}_m^T & \mathbf{1}_n^T \end{bmatrix} \begin{bmatrix} D_\ell^2 & 0 \\ 0 & D_r^2 \end{bmatrix} M_\alpha^{\circ 2} \begin{bmatrix} D_\ell^2 & 0 \\ 0 & D_r^2 \end{bmatrix} = v^T.$$

Notice that the matrix  $M_\alpha^{\circ 2}$  satisfies the hypotheses in Theorem 3.8 if  $\alpha \neq 0$ , i.e.,  $\text{supp}(M_\alpha^{\circ 2}) = \text{supp}((M_\alpha^{\circ 2})^T)$  and  $(i, i) \in \text{supp}(M_\alpha^{\circ 2})$  for all  $i = 1, \dots, n + m$ . Then, for  $\alpha \neq 0$ , there always exists a solution for this modified scaling problem. Moreover, since  $M_\alpha^{\circ 2}$  is fully indecomposable when  $M \neq 0$ , by Lemma 5.1, and is symmetric, there exists a unique and bounded diagonal scaling matrix  $\text{diag}(D_\ell^2, D_r^2)$  solving the problem (5.9)–(5.10), again by Theorem 3.8. It can be computed by the Sinkhorn-Knopp-like algorithm in Appendix A with  $r = c = v$ , as it converges to the unique solution by Theorem 3.8. In our experience, this approach very often improves, for rectangular matrices  $M$ , the approach in Theorem 5.2(c) (corresponding to apply to  $M_\alpha^{\circ 2}$  the algorithm in Appendix A with  $r = c = \mathbf{1}_{m+n}$ ) in terms of the number of steps until convergence and the quality of the scaling of the obtained matrix.

When  $\alpha = 0$ , the scaling problem (5.9)–(5.10) reduces to the problem in section 4. Then, for very small  $\alpha$ , the regularized scaling with prescribed row and column sums  $v$  tends to the scaling problem in section 4, which does not always have a solution.

In the rectangular case, simple necessary and sufficient conditions on the zero pattern of  $M$  needed for the scaling technique in section 4 (i.e., the algorithm in Appendix A applied to  $M$  with  $r = n\mathbf{1}_m$  and  $c = m\mathbf{1}_n$ ) to converge are not known (see [7, 17]). In contrast, the regularized method with the matrix  $M_\alpha^{\circ 2}$  and prescribed common vector  $v$  in (5.8) always has a solution for rectangular pencils, and we have checked in many examples that it produces satisfactory results, even when the unregularized problem has a solution. Therefore, this new regularized method is always an available option for scaling a rectangular  $M$ .

We now consider an example of a rectangular pencil whose matrix  $M$  cannot be scaled to have equal row sums and equal column sums, but we use the regularized method with prescribed common vector (5.8) to obtain an approximate scaling.

*Example 5.9.* We consider the nonsquare pencil

$$(5.11) \quad \lambda B - A = \begin{bmatrix} \lambda & 1 & 1 \\ 0 & 0 & \lambda \end{bmatrix}, \quad \text{with} \quad M = |A|^{\circ 2} + |B|^{\circ 2} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

It is easy to check that  $M$  cannot be scaled with prescribed vectors  $r := [3, 3]^T$  for the row sums and  $c := [2, 2, 2]^T$  for the column sums. Therefore, the algorithm in section 4, i.e., the algorithm in Appendix A with this  $r$  and  $c$ , does not converge for this matrix. We have run it with  $\text{tol} = 10^{-3}$  and with  $\text{tol} = 1$ , and it has not converged in any of these cases (which shows that  $\text{tol} = 1$  does not always yield convergence). More



TABLE 3

Results of the regularized method applied to the matrix  $M$  in (5.11) with  $v := [3, 3, 2, 2, 2]^T$  and  $\text{tol} = 10^{-3}$  in all cases. The quality factors  $q_S$  should be compared with  $q_S(M) = 3$ .

$\alpha$	Steps	$q_S$	$\kappa(D_\ell^2)$	$\kappa(D_r^2)$
0.5	14	1.6441	10.39	8.0413
$10^{-1}$	20	1.5073	198.27	148.92
$10^{-2}$	29	1.5001	19422	14566
$10^{-4}$	45	1.5	1.9416e+08	1.4562e+08
$10^{-10}$	93	1.5	1.9416e+20	1.4562e+20

precisely, we have run this algorithm until  $10^4$  steps, and it gets stuck, alternating periodically between the following two matrices:

$$M_{\infty,1} = \begin{bmatrix} 1.5 & 1.5 & 4.9407e - 324 \\ 0 & 0 & 3 \end{bmatrix} \quad \text{and} \quad M_{\infty,2} = \begin{bmatrix} 2 & 2 & 4.9407e - 324 \\ 0 & 0 & 2 \end{bmatrix}.$$

Observe that the quality-factors for the homogeneous scalings of the three matrices above are  $q_S(M) = 3$  and  $q_S(M_{\infty,1}) = q_S(M_{\infty,2}) = 2$ , which means that, although the un-regularized method does not converge, it has progressed toward a better scaling. Then, we use the regularized approach with different values of  $\alpha$  and prescribed common vector  $v := [3, 3, 2, 2, 2]^T$  for the row and column sums of  $M_\alpha^{o2}$ , i.e., the algorithm in Appendix A applied to  $M_\alpha^{o2}$  with  $r = c = v$  and  $\text{tol} = 10^{-3}$ . The results are shown in Table 3, where we observe that the regularization yields, even for rather large values of  $\alpha$ , a significant improvement in the scaling with a moderate number of steps and well-conditioned  $D_\ell$  and  $D_r$ . In our experiment,  $q_S$  reaches quickly a limit value of 1.5 as  $\alpha \rightarrow 0$  with the following corresponding limiting scaled matrix for  $\alpha = 10^{-10}$ :

$$M_{\alpha \rightarrow 0} = \begin{bmatrix} 1.5 & 1.5 & 1.0301e - 20 \\ 0 & 0 & 2 \end{bmatrix}.$$

We end this section by looking at the effect of the two sided scaling on the sensitivity of the underlying eigenvalue problem. In the case of regular pencils, we argued [12] (see also the discussion in section 3) that the minimization problem  $\inf_{\det T_\ell \det T_r = 1} \|T_\ell(\lambda B - A)T_r\|_F^2$ , over the arbitrary nonsingular matrix pairs  $(T_\ell, T_r)$ , yielded nearly optimal sensitivity for the generalized eigenvalues of the pencil. But since the eigenvalue problem for a singular pencil is known to be ill-conditioned, this may not make sense anymore. Nevertheless, if we constrain the transformations to be bounded, then the Kronecker structure cannot change anymore, and it then makes sense to talk about the sensitivity of the eigenvalues again. In the numerical examples in the accompanying supplementary file `supp_materials.pdf` [local/web 354KB], we show that the scaling also improves the sensitivity of the eigenvalues of the regular part of a singular pencil.

**6. Numerical examples.** In this section and in the accompanying supplementary file `supp_materials.pdf` [local/web 354KB], we verify in many numerical tests that the scaling procedures described in sections 3, 4, and 5 indeed improve the accuracy of computed eigenvalues of arbitrary pencils with a much smaller cost than computing the eigenvalues by the  $QZ$  or staircase algorithms [15, 22]. The tests were performed in MATLAB R2019a. In subsection 6.1, we focus on the computational cost of the scaling procedures, which is much smaller than the cost of computing the eigenvalues as a consequence of the use of the relaxed stopping criterion  $\text{tol} = 1$  in the algorithm in Appendix A. In subsection 6.2, we compare the accuracy of the computed eigenvalues

of regular pencils without scaling and after the scaling described in section 3. Moreover, we also compare the results with those corresponding to the scaling method of Ward [23], which is the only method implemented in LAPACK for scaling regular pencils.<sup>1</sup> This comparison was already performed in [12] but only for  $10 \times 10$  regular pencils. Our experiments confirm that the method in section 3, i.e., that in [12], outperforms Ward's method, which has a very poor behavior for certain pencils. In the supplementary file `supp_materials.pdf` [local/web 354KB], we perform similar tests on square singular pencils applying either the unregularized scaling in section 3 or, if necessary, the regularized one in section 5 and extract similar conclusions. In the supplementary file `supp_materials.pdf` [local/web 354KB], we also perform tests on rectangular pencils applying either the unregularized scaling in section 4 or, if necessary, the regularized one in subsection 5.1, which improves significantly the accuracy of the computed eigenvalues. Finally, we include in the supplementary file `supp_materials.pdf` [local/web 354KB] a test on a real-world regular pencil appearing in a polynomial eigenvalue problem.

**6.1. The stopping criterion  $\text{tol} = 1$ , computational cost and regularization.** Given a complex  $m \times n$  pencil  $\lambda B - A$ , all the scaling procedures described in this paper start by constructing the nonnegative matrix  $M := |A|^{\circ 2} + |B|^{\circ 2}$ . Then, the unregularized methods in sections 3 and 4 apply the algorithm in Appendix A to  $M$  with  $r = n\mathbf{1}_m$  and  $c = m\mathbf{1}_n$ , which in the square case means  $r = c = n\mathbf{1}_n$ . On the other hand, the regularized methods in section 5 apply the algorithm in Appendix A to the nonnegative matrix  $M_\alpha^{\circ 2}$  in (5.3) with  $r = c = (2n)\mathbf{1}_{2n}$ , when  $m = n$ , or  $r = c = v$  in the rectangular case, where  $v$  is the vector in (5.8). In both the unregularized and regularized methods, one obtains a scaled matrix  $\tilde{M} = D_\ell^2 M D_r^2$ , together with the diagonal matrices  $D_\ell^2, D_r^2$ . Then, the scaling process of the pencil finishes in exact arithmetic by computing  $D_\ell, D_r, \tilde{A} = D_\ell A D_r$ , and  $\tilde{B} = D_\ell B D_r$ , with the aim of computing the eigenvalues of  $\lambda \tilde{B} - \tilde{A}$  via some numerical algorithm. However, in real practice this must be applied in a computer, and then, there are rounding errors in the computation of  $\tilde{A} = D_\ell A D_r$  and  $\tilde{B} = D_\ell B D_r$ . This implies that the pencils  $\lambda B - A$  and  $\lambda \tilde{B} - \tilde{A}$  are not exactly strictly equivalent to each other and, in the case  $D_\ell$  and  $D_r$  are ill conditioned, as often happens in practice, their eigenvalues may be very different to each other, and the process would not be useful for improving the accuracy of computed eigenvalues. In the spirit of the classical reference [16] (see also [12, 23]), we can circumvent this difficulty if, once  $D_\ell$  and  $D_r$  have been computed, we replace their diagonal entries by their nearest integer powers of 2 to get new  $D_\ell$  and  $D_r$ . With these new approximate diagonal scalings,  $\tilde{A} = D_\ell A D_r$  and  $\tilde{B} = D_\ell B D_r$  are computed exactly in floating point arithmetic, and  $\lambda B - A$  and  $\lambda \tilde{B} - \tilde{A}$  have exactly the same eigenvalues. Of course, in this way, we do not obtain the same scaled pencil as in exact arithmetic, but it is expected that the obtained one is good enough for improving the accuracy of the computed eigenvalues.

The discussion above indicates that, for eigenvalue computations, it is not needed to apply the algorithm in Appendix A to either  $M$  or  $M_\alpha^{\circ 2}$  with a stringent stopping criterion because we will replace anyway the entries of  $D_\ell$  and  $D_r$  by their nearest integer powers of 2. The stopping criterion of the algorithm in Appendix A applied to  $M$  used for *the updating scaling*  $D_{\ell,up}$  and  $D_{r,up}$  in the iterative procedure is

$$\max \left\{ 1 - \frac{1}{\kappa(D_{\ell,up}^2)}, 1 - \frac{1}{\kappa(D_{r,up}^2)} \right\} < \frac{\text{tol}}{2}$$

<sup>1</sup>Neither MATLAB nor LAPACK [1] includes built-in functions or routines for computing eigenvalues of singular pencils.

in terms of the spectral condition numbers of  $D_{\ell,up}^2$  and  $D_{r,up}^2$ . This is equivalent to

$$\max \{ \kappa(D_{\ell,up}^2), \kappa(D_{r,up}^2) \} < 1 + \frac{\text{tol}}{2 - \text{tol}}.$$

Thus,  $\text{tol}=1$  implies that the algorithm stops when both  $D_{\ell,up}$  and  $D_{r,up}$  have a condition number smaller than  $\sqrt{2}$ . Since we are approximating the final scaling matrices to integer powers of 2, this is a safe stopping criterion for practical purposes.

We will use  $\text{tol}=1$  in all the experiments in subsection 6.2 and in the supplementary file `supp_materials.pdf` [local/web 354KB]. In this subsection, we will present some numerical tests that illustrate the impact of  $\text{tol}=1$  on the reduction of the number of steps that the algorithm in Appendix A needs for convergence and on the regularization of the problem. In all the tables for the experiments, “steps” denotes the number of steps until convergence, where one step includes one right and one left diagonal scaling. Moreover,  $q_S(M_{orig})$  denotes the quality-factor defined in (5.6) for the original matrix  $M = |A|^{\circ 2} + |B|^{\circ 2}$  and  $q_S(M_{scal})$  the one of the scaled matrix<sup>2</sup>  $\widetilde{M} = D_{\ell}^2 M D_r^2$ . The ideal goal of all our scalings procedures is to make the row sums of  $\widetilde{M}$  as equal as possible and its column sums as equal as possible as well, i.e., to get  $q_S(M_{scal}) \approx 1$ . We know that this is not always possible in exact arithmetic and that, even when it is possible in exact arithmetic, the use of entries that are integer powers of 2 in  $D_{\ell}$  and  $D_r$  prevents reaching such a goal. Thus, the practical goal is getting  $q_S(M_{scal})$  much closer to 1 than  $q_S(M_{orig})$ .

In our first test, we chose pencils of dimension  $n \times n$  with  $n = 400, 800, 1200, 1600, 2000$  and with elements that were generated using MATLAB’s `randn` function elevated to power 20, yielding matrices  $M$  with row and column sums strongly unbalanced. For each size  $n$ , we ran the algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  on ten random pencils and averaged the different tested magnitudes, both with  $\text{tol}=1$  and  $\text{tol}=10^{-3}$  and, in both cases, approximating  $D_{\ell}$  and  $D_r$  by their nearest integer powers of 2. The results are shown in Table 4. In this test, regularization is not needed because the random generation used for  $A$  and  $B$  implies that the entries of  $M$  are almost always different from zero, and thus,  $M$  has total support. Observe, that  $\text{tol}=1$  yields a much faster convergence than and similar values of  $q_S(M_{scal})$  compared with  $\text{tol}=10^{-3}$ , which is very slow on these highly unbalanced matrices. Moreover, the number of required steps does not grow with the dimension of the pencils. Observe that in this experiment  $q_S(M_{orig})$  remains essentially constant for all dimensions, which seems to be connected to the fact that the number of steps also remains essentially constant. Since each step of the scaling procedure costs  $O(n^2)$  flops, while the cost of computing the eigenvalues of an  $n \times n$  pencil with the  $QZ$ -algorithm is  $30n^3$  flops [5, section 7.7], we conclude that for the matrices in this test the computational cost of the scaling procedure with  $\text{tol}=1$  is much smaller than the cost of computing the eigenvalues.

Our second test is organized in the same way as that in Table 4, but the generated matrices  $A$  and  $B$  are sparse, with only around 1 % of their entries different from zero. They are generated as described in the caption of Table 5. The sparsity of the corresponding  $M$  matrices implies that they may not often have total support. In fact, the algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  gets stuck after 75000 steps and does not progress toward convergence for any of the matrices  $M$  generated in this test with  $\text{tol}=10^{-6}$ . This hints that a regularization would be needed in *exact arithmetic* for these

<sup>2</sup>We emphasize that, in all the experiments in section 6 and in the supplementary file `supp_materials.pdf` [local/web 354KB], the matrix  $\widetilde{M}$  is computed as  $\widetilde{M} = |D_{\ell} A D_r|^{\circ 2} + |D_{\ell} B D_r|^{\circ 2}$ , where the diagonal matrices  $D_{\ell}$  and  $D_r$  are the ones whose diagonal entries are integer powers of 2.

TABLE 4

Numerical test illustrating that the use of  $\text{tol}=1$  decreases very much the number of steps without affecting or the quality of the scal of the achieved scaled matrix  $\tilde{M}$  the condition numbers of  $D_\ell$  and  $D_r$ . The algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  has been applied to the matrices  $M$  of exactly the same set of  $n \times n$  pencils generated in MATLAB as  $\mathbf{A}=\text{randn}(n,n)$  and  $\mathbf{B}=\text{randn}(n,n)$ , one time with  $\text{tol}=1$  and another time with  $\text{tol}=10^{-3}$ . No regularization is used, which is indicated with  $\alpha = 0$ .

$\text{tol}=1$ and $\alpha = 0$					
$n$	$q_S(M_{orig})$	$q_S(M_{scal})$	$\kappa(D_\ell)$	$\kappa(D_r)$	Steps
400	1.94e+10	1.24e+01	2.76e+03	4.30e+04	9.8
800	4.90e+09	1.37e+01	2.46e+03	2.54e+04	10
1200	1.12e+10	1.35e+01	2.97e+03	1.35e+04	10.9
1600	2.79e+09	1.37e+01	2.56e+03	1.23e+04	10.7
2000	4.07e+09	1.42e+01	2.00e+03	1.37e+04	10.8
$\text{tol}=10^{-3}$ and $\alpha = 0$					
$n$	$q_S(M_{orig})$	$q_S(M_{scal})$	$\kappa(D_\ell)$	$\kappa(D_r)$	Steps
400	1.94e+10	1.18e+01	1.11e+04	1.64e+04	1367.5
800	4.90e+09	1.20e+01	5.53e+03	8.70e+03	1616
1200	1.12e+10	1.26e+01	7.17e+03	7.27e+03	1470.7
1600	2.79e+09	1.27e+01	6.14e+03	4.30e+03	1323.3
2000	4.07e+09	1.27e+01	5.32e+03	5.94e+03	1382.4

TABLE 5

Numerical test illustrating that the use of  $\text{tol}=1$  often has a regularizing effect. The algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  has been applied to the matrices  $M$  of  $n \times n$  pencils generated in MATLAB as  $\mathbf{A}=\text{eye}(n) + \text{sprandn}(n,n,0.01)$  and  $\mathbf{B}=\text{eye}(n) + \text{sprandn}(n,n,0.01)$ , with  $\text{tol}=1$ , and the results are shown. In contrast, the same algorithm applied to the same set of pencils with  $\text{tol}=10^{-6}$  does not converge for any of the generated matrices. The same happens if the power 20 is replaced by 10.

$\text{tol}=1$ and $\alpha = 0$					
$n$	$q_S(M_{orig})$	$q_S(M_{scal})$	$\kappa(D_\ell)$	$\kappa(D_r)$	Steps
400	2.58e+22	1.33e+01	9.19e+16	1.27e+17	40.5
800	3.29e+24	1.40e+01	2.72e+14	3.74e+15	33.9
1200	9.25e+25	1.47e+01	2.89e+11	2.97e+13	27.8
1600	2.31e+26	1.51e+01	2.34e+11	1.11e+12	28.4
2000	3.76e+22	1.51e+01	2.34e+10	1.61e+11	26

pencils. However, the unregularized algorithm has always converged rather quickly with  $\text{tol}=1$ , yielding, moreover, very satisfactory scalings as measured by  $q_S(M_{scal})$ . The results are shown in Table 5. This test is just one example of a natural phenomenon that we have observed, namely, that the use of  $\text{tol}=1$  has very often a regularization effect that makes it unnecessary to use, for computing accurate eigenvalues of pencils, the regularization techniques in section 5. We announced this phenomenon in Example 5.6, but we have observed it in many other cases where the matrix  $M$  does not have total support. The explanation of this phenomenon is that with  $\text{tol}=1$  the algorithm stops in the first step for which the column sums differ by a factor no larger than 2 and the next row sums also differ by a factor no larger than 2, and this may happen in many cases for which the scaling to a doubly stochastic matrix does not exist. Observe that the convergence in Table 5 is slower than in Table 4. As we discuss below, this is due to the fact that the values of  $q_S(M_{orig})$  are larger but also to the larger sparsity.

Our third test is organized as the previous ones. The test pencils are in this case random permutations of square block diagonal pencils with rectangular diagonal blocks. They are generated as described in the caption of Table 6. None of the corresponding

TABLE 6

Numerical test illustrating pencils where regularization with  $\text{tol}=1$  is used. The algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  applied to the matrices  $M$  of  $n \times n$  pencils generated in MATLAB as random permutations of  $A = \text{blkdiag}(\text{randn}(n_1, n_2) \cdot \wedge(20), \text{randn}(n_2, n_1) \cdot \wedge(20))$  and  $B = \text{blkdiag}(\text{randn}(n_1, n_2) \cdot \wedge(20), \text{randn}(n_2, n_1) \cdot \wedge(20))$  with  $n_1 = n/5$  and  $n_2 = n - n_1$  does not converge in 2000 steps with  $\text{tol}=1$ . The same happens if the exponent 20 is replaced by 10 or 5. In contrast, the algorithm in Appendix A with  $r = c = (2n)\mathbf{1}_{2n}$  applied to the matrices  $M_\alpha^{\circ 2}$  in (5.3) with  $\text{tol}=1$  and  $\alpha = 0.5, 10^{-4}$  converges, and the results are shown below. We have checked that the use of smaller values of  $\alpha$  does not improve the quality of the achieved scaling but worsens the condition numbers of  $D_\ell$  and  $D_r$  and increases the number of steps until convergence.

$\text{tol}=1$ and $\alpha = 0.5$					
$n$	$q_S(M_{orig})$	$q_S(M_{scal})$	$\kappa(D_\ell)$	$\kappa(D_r)$	Steps
200	1.59e+15	1.75e+09	3.74e+12	3.30e+12	32.8
400	2.14e+14	4.93e+07	1.94e+13	6.77e+13	34.5
600	9.30e+13	6.64e+06	1.37e+14	5.63e+13	33.6
800	1.20e+13	3.45e+06	1.20e+14	1.13e+14	33.0
1000	4.72e+12	4.57e+06	1.48e+14	1.41e+14	34.0

$\text{tol}=1$ and $\alpha = 10^{-4}$					
$n$	$q_S(M_{orig})$	$q_S(M_{scal})$	$\kappa(D_\ell)$	$\kappa(D_r)$	Steps
200	1.59e+15	1.83e+09	2.25e+16	1.46e+16	39.5
400	2.14e+14	5.42e+07	9.01e+16	5.22e+17	41.0
600	9.30e+13	4.97e+06	6.20e+17	2.59e+17	40.0
800	1.20e+13	3.72e+06	5.04e+17	4.76e+17	39.1
1000	4.72e+12	5.20e+06	7.21e+17	5.76e+17	40.1

$M$  matrices has total support. The key difference with respect to the tests in Tables 4 and 5 is that in this case the algorithm in Appendix A with  $\text{tol}=1$  and  $r = c = n\mathbf{1}_n$  applied to the matrices  $M$  never converges in 2000 steps; i.e.,  $\text{tol}=1$  does not have a quick regularizing effect for these pencils. Thus, we use the regularized algorithm. The results are shown in Table 6. Note that, although the values for  $q_S(M_{scal})$  are much better than those of  $q_S(M_{orig})$ , they are far from 1. Moreover, the values of  $q_S(M_{scal})$  do not improve by decreasing  $\alpha$ . Despite these facts, we will see in some experiments done in the supplementary file [supp\\_materials.pdf](#) [local/web 354KB] on similar pencils that the regularized scaling has significant positive effects on the accuracy of the computed eigenvalues.

We finish this subsection with two additional tests. The first one is described in Table 7 and is the same as the one in Table 4 but with starting matrices  $M$  that are less strongly unbalanced as measured by  $q_S(M_{orig})$ . This leads to a much faster convergence than in Table 4, as it is naturally expected. The comparison of Tables 4 and 7 shows that the number of steps until convergence grows with the unbalancing of the  $M$  matrices but, also, that the number of steps is independent of the dimension of the matrices. The last test is described in Table 8 and is the same as the one in Table 5 but with sparse starting matrices  $M$  that are less strongly unbalanced, which lead again to a much faster convergence, independent, more or less, of the dimension of the matrices. The comparison of Table 4 (for  $\text{tol}=1$ ), for dense pencils, and Table 8, for sparse pencils, is interesting because both show similar values of  $q_S(M_{orig})$ , but the convergence is slower in the sparse case. This illustrates that, for  $\text{tol}=1$ , the well-known effect that sparsity slows down the convergence of the Sinkhorn–Knopp algorithm also holds [8].

As a summary of the results in this subsection, we emphasize that, even for pencils leading to extremely unbalanced matrices  $M$ , the computational cost of the scaling procedures proposed in this paper with the stopping criterion  $\text{tol}=1$  is much smaller than the cost of computing the eigenvalues.

TABLE 7

Numerical test equal to that in Table 4 for  $\text{tol}=1$  except for the fact that the  $n \times n$  pencils are generated in MATLAB as  $\mathbf{A}=\text{randn}(n,n) \cdot 10$  and  $\mathbf{B}=\text{randn}(n,n) \cdot 10$ . The use of the exponent 10 instead of 20 in the generation of the test matrices implies that the original matrices  $M$  are better equilibrated than those in Table 4, as indicated by the values of  $q_S(M_{orig})$ , which, in turns, implies a faster convergence in approximately half of the steps.

$n$	$q_S(M_{orig})$	$\text{tol}=1$ and $\alpha=0$			
		$q_S(M_{scal})$	$\kappa(D_\ell)$	$\kappa(D_r)$	Steps
400	4.04e+04	1.11e+01	3.84e+01	8.64e+01	5.1
800	1.75e+04	1.07e+01	2.72e+01	6.40e+01	5.0
1200	1.78e+04	1.12e+01	2.24e+01	5.12e+01	5.1
1600	1.37e+04	1.13e+01	2.72e+01	4.80e+01	4.8
2000	1.42e+04	1.15e+01	2.40e+01	5.12e+01	5.0

TABLE 8

Numerical test equal to that in Table 5 except for the fact that the  $n \times n$  pencils are generated in MATLAB as  $\mathbf{A}=\text{eye}(n) + \text{sprandn}(n,n,0.01) \cdot 10$  and  $\mathbf{B}=\text{eye}(n) + \text{sprandn}(n,n,0.01) \cdot 10$ . The use of the exponent 10 instead of 20 in the generation of the test matrices implies that the original matrices  $M$  are better equilibrated than those in Table 5, as indicated by the values of  $q_S(M_{orig})$ , which, in turns, implies a faster convergence in approximately half of the steps.

$n$	$q_S(M_{orig})$	$\text{tol}=1$ and $\alpha=0$			
		$q_S(M_{scal})$	$\kappa(D_\ell)$	$\kappa(D_r)$	Steps
400	8.87e+10	1.31e+01	1.96e+08	1.85e+08	20.6
800	1.04e+12	1.38e+01	8.07e+06	1.93e+07	17.1
1200	3.96e+12	1.41e+01	3.87e+05	3.04e+06	14.3
1600	3.39e+12	1.42e+01	3.28e+05	5.24e+05	14.5
2000	1.12e+11	1.54e+01	8.19e+04	2.29e+05	13.1

**6.2. Examples on the accuracy of computed eigenvalues of regular pencils.** In this section, we discuss numerical tests for three families of regular pencils. In each of these families, we generated random diagonalizable  $n \times n$  regular pencils  $\lambda B - A$  for which their “exact” eigenvalues  $\lambda_i$  were known. Then, we applied the  $QZ$ -algorithm [15] in MATLAB to such pencils, to the scaled pencils  $D_\ell(\lambda B - A)D_r$  obtained by applying the algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  and  $\text{tol}=1$  to  $M = |A|^2 + |B|^2$ , and to the pencils balanced by Ward’s method [23]. In all cases, we constrained the diagonal elements of the diagonal scaling matrices to be integer powers of 2. Since MATLAB does not have a built-in function implementing Ward’s method, we used the one in [24]. For each generated pencil, we compared the “exact” eigenvalues  $\lambda_i$  of the pencil with the eigenvalues  $\tilde{\lambda}_i$  computed via the three options above. For the comparison of the eigenvalues, we used their chordal distances [20]

$$c_i := \chi(\lambda_i, \tilde{\lambda}_i) := \frac{|\lambda_i - \tilde{\lambda}_i|}{\sqrt{1 + |\lambda_i|^2} \sqrt{1 + |\tilde{\lambda}_i|^2}}.$$

We compared the quantities  $c := \|[c_1, \dots, c_n]\|_2$  for the original pencil  $(\lambda B - A)$  ( $c_{orig}$ ), for the balanced pencil  $D_\ell(\lambda B - A)D_r$  constructed by applying the algorithm in Appendix A to  $M$  ( $c_{bal}$ ), and for the pencil balanced by Ward’s method ( $c_{ward}$ ). The regularization techniques of section 5 were not used in this section since the algorithm in Appendix A applied to  $M$  with  $\text{tol}=1$  always converged in a very small number of steps, as can be seen in the tables. In fact, we have not found any *regular* pencil where the algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  and  $\text{tol}=1$  applied to  $M$  does not converge in a small number of steps, even considering very sparse pencils and particular nonrandom pencils whose associated matrices  $M$  do not have total support.

In the first family of tests of this subsection, we generated  $500 \times 500$  random diagonalizable pencils of the form  $T_\ell(\lambda\Lambda_B - \Lambda_A)T_r$ , where  $(\lambda\Lambda_B - \Lambda_A)$  is in standard normal form [12], i.e.,  $\Lambda_A$  and  $\Lambda_B$  are diagonal, and  $|\Lambda_A|^2 + |\Lambda_B|^2 = I_n$ . The condition number of the random square nonsingular matrices  $T_\ell$  and  $T_r$  was controlled by taking the  $k$ th power of normally distributed random numbers  $r_{ij}$  as their elements. A larger power  $k$  typically yields a larger condition number. The obtained results are shown in Table 9, where each row corresponds to a value of  $k$  taken in increasing order from  $k = 1 : 5 : 41$  in MATLAB notation. This experiment shows that the scaling in section 3 based on the algorithm in Appendix A does improve the accuracy of the computed eigenvalues with respect to the original pencil and to the pencil scaled by Ward's method, especially when the pencil corresponds to badly conditioned transformations  $T_\ell$  and  $T_r$ . Moreover, the algorithm in Appendix A converged in a very small number of steps and produced a very well scaled matrix  $\tilde{M}$ .

It is well known that Ward's method can severely deteriorate the accuracy of the computed eigenvalues of some pencils [9, Chapter 2, Section 4.2], [12]. In the second family of tests of this subsection, we generated a family of  $500 \times 500$  pencils where Ward's method led to computed eigenvalues with large errors, but the method in section 3 performed very well in accuracy and convergence rate. We have not been able to generate pencils with the opposite behavior. The pencils were generated as follows: (1) a random  $500 \times 500$  matrix  $T$  was constructed with the MATLAB command `randn`; (2) small entries were created in  $T$  with  $T(1, 2 : 500) = 10^{-k}T(1, 2 : 500)$  and  $T(4 : 500, 3) = 10^{-k}T(4 : 500, 3)$ ; (3)  $A = TD$ , with  $D$  a random diagonal matrix of integer positive numbers, and  $B = T$ . The eigenvalues of  $\lambda B - A$  are the diagonal entries of  $D$ . The results are in Table 10, where each row corresponds to a value of  $k$ .

In the experiments presented so far in this subsection, the scaling method in section 3 always improved significantly the accuracy of the computed eigenvalues. However, there are pencils where the improvement is much larger. This is illustrated in the last

TABLE 9

*Eigenvalue accuracy of the QZ-algorithm for regular  $500 \times 500$  pencils: for the original pencil, for the pencil balanced by applying the algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  and  $\text{tol} = 1$  to  $M = |A|^{\circ 2} + |B|^{\circ 2}$ , and for the pencil balanced by Ward's method. The improvement in the scaling of  $M$  produced by the algorithm in Appendix A is also shown in terms of  $q_S(M_{orig})$  and  $q_S(M_{scal})$  (see (5.6)), as well as the number of its steps until convergence.*

$\kappa(T_\ell)$	$\kappa(T_r)$	$c_{orig}$	$c_{bal}$	$c_{ward}$	$c_{bal}/c_{orig}$	$c_{bal}/c_{ward}$
2.45e+03	1.03e+03	7.42e-13	7.42e-13	7.19e-13	1.00e+00	1.03e+00
4.11e+03	4.20e+03	5.29e-13	4.25e-13	4.61e-13	8.03e-01	9.22e-01
2.01e+05	5.26e+04	1.59e-11	4.89e-12	5.33e-12	3.08e-01	9.17e-01
4.25e+07	3.87e+06	9.94e-10	1.92e-11	2.28e-10	1.93e-02	8.39e-02
4.55e+08	2.83e+07	2.09e-08	1.07e-10	2.07e-09	5.13e-03	5.20e-02
7.47e+10	2.62e+10	1.19e-05	5.67e-08	8.97e-06	4.76e-03	6.31e-03
9.18e+11	7.91e+11	2.57e-03	1.96e-05	1.16e-03	7.63e-03	1.69e-02
5.31e+14	1.29e+14	4.80e-01	3.44e-06	7.40e-03	7.18e-06	4.65e-04
9.66e+16	5.23e+14	1.33e-01	2.20e-03	2.09e-01	1.65e-02	1.05e-02

$q_S(M_{orig})$	$q_S(M_{scal})$	Steps
1.62e+00	1.62e+00	1
4.25e+03	5.73e+00	4
1.11e+06	9.03e+00	5
9.32e+09	1.16e+01	9
6.12e+11	1.01e+01	12
7.54e+16	9.97e+00	14
5.57e+18	1.18e+01	17
5.15e+24	1.07e+01	23
3.53e+26	1.35e+01	21

TABLE 10

*Eigenvalue accuracy of the QZ-algorithm for regular  $500 \times 500$  pencils for which Ward's method deteriorates the precision of computed eigenvalues: for the original pencil, for the pencil balanced by applying the algorithm in Appendix A with  $r = c = n\mathbf{1}_n$  and  $\text{tol} = 1$  to  $M$ , and for the pencil balanced by Ward's method.*

k	$c_{orig}$	$c_{bal}$	$c_{ward}$	$c_{bal}/c_{orig}$	$c_{bal}/c_{ward}$	$c_{ward}/c_{orig}$
1	2.61e-13	3.40e-15	8.87e-15	1.31e-02	3.84e-01	3.40e-02
3	1.48e-13	7.59e-15	1.91e-14	5.14e-02	3.98e-01	1.29e-01
5	4.13e-13	8.72e-15	4.56e-09	2.11e-02	1.91e-06	1.10e+04
7	7.16e-14	2.27e-15	3.47e-02	3.17e-02	6.54e-14	4.84e+11
9	3.90e-13	3.01e-15	1.05e+00	7.72e-03	2.87e-15	2.69e+12
11	1.34e-13	7.99e-15	1.08e+00	5.96e-02	7.38e-15	8.08e+12

$q_S(M_{orig})$	$q_S(M_{scal})$	steps
5.11e+04	4.02e+00	2
1.16e+05	4.33e+00	3
1.43e+05	4.33e+00	3
6.40e+03	4.50e+00	3
1.47e+05	4.73e+00	3
1.37e+05	4.74e+00	3

TABLE 11

*Eigenvalue accuracy of the QZ-algorithm for regular  $1000 \times 1000$  pencils for which the method based on the algorithm in Appendix A applied to both  $M$  and Ward's method very well works.*

$c_{orig}$	$c_{bal}$	$c_{ward}$	$c_{bal}/c_{orig}$	$c_{bal}/c_{ward}$	$q_S(M_{orig})$	$q_S(M_{scal})$	Steps
5.27e-14	5.33e-14	5.17e-14	1.01e+00	1.03e+00	1.12e+02	4.13e+00	2
4.47e-06	5.23e-14	5.77e-14	1.17e-08	9.05e-01	1.65e+20	4.21e+00	3
1.33e+01	6.49e-14	6.41e-14	4.86e-15	1.01e+00	1.53e+38	4.13e+00	3

test of this subsection. The pencils were constructed as those in the experiment of Table 9, i.e.,  $T_\ell(\lambda\Lambda_B - \Lambda_A)T_r$ , but with different  $T_\ell$  and  $T_r$ . In this case,  $T_\ell = D_1Q_\ell$  and  $T_r = Q_rD_2$ , with  $Q_\ell$  and  $Q_r$  random orthogonal matrices and  $D_1$  and  $D_2$  random diagonal matrices with condition numbers  $10^k$  and geometrically distributed singular values, constructed with the command `gallery('randsvd', ...)` of MATLAB. The results are shown in Table 11 for  $1000 \times 1000$  pencils and  $k = 1, 10, 19$  (each value for each row of the table). Ward's method also yields very accurate eigenvalues.

As a consequence of the results in this subsection, we emphasize again that the scaling method in section 3, i.e., that in [12], often contributes to improve the accuracy of computed eigenvalues of regular pencils significantly and outperforms the method of Ward [23], which is the only one available so far in LAPACK [1].

**7. Concluding remarks.** In this paper, we developed new scaling techniques that apply to both regular and singular pencils. The techniques are based on applying the Sinkhorn-Knopp-like algorithm to certain nonnegative matrices easily constructed from the matrix coefficients of the pencil, and that depends on whether the scaling problem needs to be regularized or not. The regularization guarantees to always get a unique and bounded solution. Extensive numerical experiments confirm that the proposed techniques very often significantly improve the accuracy of computed eigenvalues of arbitrary pencils and outperform earlier methods for scaling regular pencils. Finally, the algorithms computing these scalings have a computational cost that is much smaller than the cost of the subsequent generalized eigenvalue problem as a consequence of using in the Sinkhorn-Knopp-like algorithm a proper stopping criterion compatible with computing diagonal scalings whose diagonal entries are integer powers of 2.



### Appendix A.

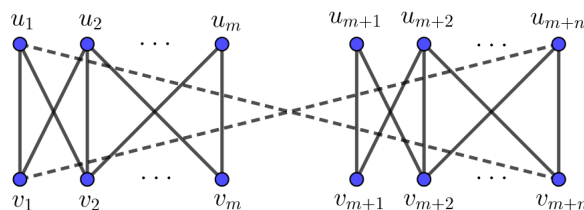
Sinkhorn-Knopp-like algorithm MATLAB code with prescribed row sums and column sums.

```
function [Md,dleft,dright] = rowcolsums(M,r,c,maxiter,tol)
%
%       Md:=diag(dleft)*M*diag(dright)
%
% has column sums equal to a positive row vector c and
% row sums equal to a positive column vector r where sum(c)=sum(r)
%
% M and Md are nonnegative matrices, maxiter is the maximum number
% of allowed iterations and tol determines the stopping criterion
%
[m,n]=size(M);
% Scale the matrix to have total sum(sum(M))=sum(c)=sum(r);
sumcr=sum(c);sumM=sum(sum(M));Md=M*sumcr/sumM;
dleft=ones(m,1)*sqrt(sumcr/sumM);dright=ones(1,n)*sqrt(sumcr/sumM);
% Scale left and right to make row and column sums equal to r and c
for i=1:maxiter;
dr=sum(Md,1)./c;Md=Md./dr;er=min(dr)/max(dr);dright=dright./dr;
dl=sum(Md,2)./r;Md=dl.\Md;el=min(dl)/max(dl);dleft=dleft./dl;
if max([1-er, 1-el]) < tol/2, break; end
end
% Finally scale the two scalings to have equal maxima
scaled=sqrt(max(dright)/max(dleft));
dleft=dleft*scaled;dright=dright'/scaled;
end
```

**Appendix B. Proof of Lemma 5.1.**  $M_\alpha^{\circ 2}$  has total support for all  $\alpha \neq 0$  since every nonzero element is an element of a positive diagonal. To see that  $M_\alpha^{\circ 2}$  is fully indecomposable, we apply [4, Theorem 1.3.7]. This theorem states that a square matrix with total support is fully indecomposable if and only if its bipartite graph is connected. Then we consider the bipartite graph of  $M_\alpha^{\circ 2}$ , denoted by  $BG(M_\alpha^{\circ 2})$ . We assume without loss of generality that  $m_{1n}$  is a nonzero element of  $M := [m_{ij}]$ . Then we consider the matrix

$$N := \left[ \begin{array}{cc|cc} \frac{\alpha^2}{m^2} 1_m 1_m^T & 0 & 0 & m_{1n} \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \frac{\alpha^2}{n^2} 1_n 1_n^T & \\ m_{1n} & 0 & & \end{array} \right].$$

Notice that  $BG(N)$  is a subgraph of  $BG(M_\alpha^{\circ 2})$ . Moreover, if  $\{u_1, u_2, \dots, u_{m+n}\}$  and  $\{v_1, v_2, \dots, v_{m+n}\}$  are the sets of vertices associated with the rows and columns of  $N$ , respectively, then  $BG(N)$  is of the form



where the left and right groups of solid edges are each bicliques (and hence connected) and where the two dashed edges correspond to the element  $m_{1n}$ . This proves that  $BG(N)$  is connected since the dashed edges make a connection between two connected components. Therefore,  $BG(M_\alpha^{\circ 2})$  is connected, and  $M_\alpha^{\circ 2}$  is fully indecomposable.

**Acknowledgments.** The authors sincerely thank two anonymous referees for pointing out several significant suggestions and a number of relevant references that have contributed to improving this manuscript.

## REFERENCES

- [1] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide*, 3rd ed., Software Environ. Tools 9, SIAM, Philadelphia, 1999.
- [2] R. A. BRUALDI, *Matrices of 0's and 1's with total support*, J. Combin. Theory Ser. A, 28 (1980), pp. 249–256.
- [3] R. A. BRUALDI, *Convex sets of non-negative matrices*, Canad. J. Math., 20 (1968), pp. 144–157.
- [4] R. A. BRUALDI, *Combinatorial Matrix Classes*, Cambridge University Press, New York, 2006.
- [5] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 4th ed., Johns Hopkins Stud. Math. Sci., Johns Hopkins University Press, Baltimore, MD, 3 2013.
- [6] D. J. HARTFIEL AND C. J. MAXSON, *The chainable matrix, a special combinatorial matrix*, Discrete Math., 12 (1975), pp. 245–256.
- [7] M. IDEL, *A Review of Matrix Scaling and Sinkhorn's Normal Form for Matrices and Positive Maps*, preprint, arXiv:1609.06349, 2016.
- [8] P. A. KNIGHT, *The Sinkhorn–Knopp algorithm: Convergence and applications*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 261–275.
- [9] D. KRESSNER, *Numerical Methods and Software for General and Structured Eigenvalue Problems*, Ph.D. Thesis, Technische Universität Berlin, 2004.
- [10] J. KRUIHOF, *Telefoonverkeersrekening*, De Ingenieur, 52 (1937), pp. E15–E25.
- [11] R. S. KRUPP, *Properties of Kruihof's projection method*, Bell Syst. Tech. J., 58 (1979), pp. 517–538.
- [12] D. LEMONNIER AND P. VAN DOOREN, *Balancing regular matrix pencils*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 253–263.
- [13] M. V. MENON, *Matrix links, an extremization problem, and the reduction of a non-negative matrix to one with prescribed row and column sums*, Canad. J. Math., 20 (1968), pp. 225–232.
- [14] M. V. MENON AND H. SCHNEIDER, *The spectrum of a nonlinear operator associated with a matrix*, Linear Algebra Appl., 2 (1969), pp. 321–334.
- [15] C. B. MOLER AND G. W. STEWART, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1971), pp. 241–256.
- [16] B. PARLETT AND C. REINSCH, *Balancing a matrix for the calculation of eigenvalues and eigenvectors*, Numer. Math., 13 (1969), pp. 293–304.
- [17] G. PEYRÉ AND M. CUTURI, *Computational optimal transport: with applications to data science*, Found. Trends Mach. Learn., 11 (2019), pp. 355–607.
- [18] U. G. ROTHBLUM AND H. SCHNEIDER, *Scalings of matrices which have prespecified row sums and column sums via optimization*, Linear Algebra Appl., 114/115 (1989), pp. 737–764.
- [19] R. SINKHORN AND P. KNOPP, *Concerning nonnegative matrices and doubly stochastic matrices*, Pacific J. Math., 21 (1967), pp. 343–348.
- [20] G. W. STEWART AND J.-G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
- [21] MathWorks, *MATLAB, Reference Guide: High-Performance Numeric Computation and Visualization Software*, The MathWorks, Natick, MA, 1992.
- [22] P. VAN DOOREN, *The computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 27 (1979), pp. 103–141.
- [23] R. C. WARD, *Balancing the generalized eigenvalue problem*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 141–152.
- [24] M. WEDER, *BALANCE2 Balancing Generalized Eigenvalue Problem*, MATLAB Central File Exchange, 2015 <https://www.mathworks.com/matlabcentral/fileexchange/49719-balance2-balancing-generalized-eigenvalue-problem> (retrieved May 5, 2021).
- [25] G. U. YULE, *On the methods of measuring association between two attributes*, J. R. Stat. Soc., 75 (1912), pp. 579–652.